# Optimal Transport: The Dynamical Monge-Kantorovich Model as a Bridge between Theory and Applications

**Dissertation**
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
M. Sc. Diego Alejandro Baptista Theuerkauf
aus Mérida, Venezuela

Tübingen
2024

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

| | |
|---|---|
| Tag der mündlichen Qualifikation: | 24.07.2024 |
| Dekan: | Prof. Dr. Thilo Stehle |
| 1. Berichterstatter/-in: | Dr. Caterina De Bacco |
| 2. Berichterstatter/-in: | Prof. Dr. Britta Dorn |

# Abstract

Traffic problems are an essential part of our everyday lives, from daily commuter traffic to the development of urban infrastructures and global freight transportation. A key challenge is to optimize the allocation of resources to minimize costs or maximize efficiency. Optimal Transportation (OT) theory provides a mathematical framework to formally describe such problems and their solution, taking into account complex objectives and constraints. Over the years, various methods have been developed to solve OT problems. However, traditional methods reach their limits when solution spaces are continuous and the consideration of traffic density is important. The Dynamic Monge-Kantorovich (DMK) algorithm developed in recent years offers a solution to these challenges. It models traffic problems using dynamic equations and enables the solution of transportation problems in continuous two-dimensional solution spaces. It also allows traffic constraints due to traffic density to be taken into account, such as prioritizing busy routes over many smaller, similar routes. With these properties, DMK offers a practical method for solving real-world problems.

Nevertheless, the DMK algorithm reaches its limits in certain situations. Although many practical scenarios are defined in continuous space and therefore require a continuous solution, a discrete solution, for example in the form of a graph, is often required for implementation in the real world. In addition, the algorithm often requires a longer execution time, which can affect its practical applicability in time-critical scenarios. Furthermore, the DMK model has not yet been sufficiently tested in its practical application, especially in the field of machine learning, where OT problems play a major role.

In this thesis, we extend the aforementioned limitations of the DMK and show the practical application of the model in classical machine learning contexts. We first present different methods to transform the continuous solutions of the original two-dimensional space in which the DMK model is defined into practically realizable discrete solutions. In the process, we transform DMK solutions, which are available as density distributions, into graphs and hypergraphs that represent discrete solutions. Furthermore, we show that the DMK algorithm can be stopped before its termination without strongly affecting the solution quality. In time-critical practical applications, the DMK algorithm can thus find effective solutions in a short time by determining an optimal stopping time.

The second part of this thesis focuses on the practical applicability of the DMK model. Based on the theoretical findings described above, we develop algorithms for the practical use of a variant of the DMK model, the Graph-DMK algorithm, in various machine learning tasks. Specifically, we show how DMK can be used for the clustering of networks, the extraction of networks from images, and the classification of images. Our results show the potential of the DMK algorithm to successfully solve a variety of machine learning tasks.

Thus, this thesis extends the previous limits of the DMK model and demonstrates its practical application.

# Zusammenfassung

Verkehrsprobleme sind ein wesentlicher Bestandteil unseres Alltags, angefangen beim täglichen Berufsverkehr über die Entwicklung städtischer Infrastrukturen bis hin zum globalen Gütertransport. Eine zentrale Herausforderung besteht dabei darin, die Ressourcenallokation zu optimieren, um Kosten zu minimieren oder Effizienz zu maximieren. Die Theorie des optimalen Transports (OT) bietet einen mathematischen Rahmen zur formalen Beschreibung solcher Probleme und ihrer Lösung unter Berücksichtigung komplexer Ziele und Einschränkungen. Im Laufe der Jahre wurden verschiedene Methoden entwickelt, um OT-Probleme zu lösen. Traditionelle Methoden stoßen jedoch an Grenzen, wenn Lösungsräume kontinuierlich sind und die Berücksichtigung der Verkehrsdichte von Bedeutung ist. Der in den letzten Jahren entwickelte Dynamic Monge-Kantorovich (DMK) Algorithmus bietet eine Lösung für diese Herausforderungen. Er modelliert Verkehrsprobleme mithilfe dynamischer Gleichungen und ermöglicht die Lösung von Transportproblemen in kontinuierlichen zweidimensionalen Lösungsräumen. Außerdem erlaubt er die Berücksichtigung von Verkehrseinschränkungen aufgrund der Verkehrsdichte, wie etwa die Priorisierung stark frequentierter Routen gegenüber vielen kleineren, ähnlich verlaufenden Wegen. Mit diesen Eigenschaften bietet DMK eine praktische Methode zur Lösung realweltlicher Probleme.

Dennoch stößt der DMK-Algorithmus in bestimmten Situationen an seine Grenzen. Viele praktische Szenarien sind zwar im kontinuierlichen Raum definiert und erfordern daher eine kontinuierliche Lösung, aber für die Umsetzung in der realen Welt ist oft eine diskrete Lösung erforderlich, zum Beispiel in Form eines Graphen. Außerdem benötigt der Algorithmus häufig eine längere Ausführungszeit, was seine praktische Anwendbarkeit in zeitkritischen Szenarien beeinträchtigen kann. Darüber hinaus wurde das DMK-Modell bisher noch nicht in seiner praktischen Anwendung, insbesondere im Bereich des maschinellen Lernens, wo OT-Probleme eine große Rolle spielen, ausreichend getestet.

In dieser Arbeit erweitern wir die genannten Grenzen des DMK und zeigen die praktische Anwendung des Modells in klassischen Machine-Learning-Kontexten. Wir stellen zunächst verschiedene Methoden vor, um die kontinuierlichen Lösungen des ursprünglichen zweidimensionalen Raums, in dem das DMK-Modell definiert ist, in praktisch umsetzbare diskrete Lösungen zu transformieren. Dabei wandeln wir DMK-Lösungen, die als Dichteverteilungen vorliegen, in Graphen und Hypergraphen um, die diskrete Lösungen darstellen. Außerdem zeigen wir, dass der DMK-Algorithmus vor seiner Terminierung angehalten werden kann, ohne die Lösungsqualität stark zu beeinträchtigen. In zeitkritischen praktischen Anwendungen kann der DMK-Algorithmus somit durch die Festlegung einer optimalen Anhaltezeit effektive Lösungen in kurzer Zeit finden.

Der zweite Teil dieser Arbeit fokussiert sich auf die praktische Anwendbarkeit des DMK-Modells. Basierend auf den oben beschriebenen theoretischen Erkenntnissen entwickeln wir Algorithmen zur praktischen Nutzung einer Variante des DMK-Modells, dem Graph-DMK-Algorithmus, in verschiedenen Aufgaben des maschinellen Lernens. Konkret zeigen wir, wie DMK für das Clustern von Netzwerken, die Extraktion von Netzwerken aus Bildern und die Klassifizierung von Bildern eingesetzt werden kann. Unsere Ergebnisse zeigen das Potenzial des DMK-Algorithmus, eine Vielzahl von maschinellen Lernaufgaben erfolgreich zu lösen.

Damit erweitert diese Arbeit die bisherigen Grenzen des DMK-Modells und demonstriert seine praktische Anwendung.

# Acknowledgements

I want to express my gratitude to everyone who has helped me along the way. First of all, I am extremely grateful to Caterina De Bacco for being an incredible supervisor. I value her support, trust, insightful discussions, exchange of ideas, and examples of how to maintain a healthy work-life balance, as well as her advice, both academic and personal, and her genuine interest in my achievements in and outside of my studies. I am also incredibly grateful for this life-changing opportunity.

During my PhD, I have had the privilege of working with great collaborators: Daniela Leite, Alessandro Lonardi, Abdullahi Ibrahim, Enrico Facca, and Mario Putti. Their knowledge of different topics made me grow and feel even more passionate about my research. Their company made my work much more exciting and efficient.

I would like to thank the administrative teams at the Max Planck Institute for Intelligent Systems and the IMPRS-IS graduate school for creating an academic environment that is both enriching and enjoyable. I would like to give a special thanks to Leila Masri for her support, guidance, and engaging conversations about Venezuela.

I am incredibly grateful to Georg Martius and Matthias Hein for being part of my Thesis Advisory Committee. Their curiosity and advice helped me to keep up the pace. I would also like to thank Fabian Theis and the Helmholtz Information and Data Science Academy for hosting me and funding my visit last summer at the Helmholtz Munich Center. My stay gave me new perspectives on research and helped me find a new passion. I am grateful to Jean Claude-Passy for inviting me to join the Software Workshop group after my visit and to explore this interest further.

I am also grateful to the Physics for Inference and Optimization (current and former) members Andrea Della Vecchia, Hadiseh Safdari, Emanuele Pigani, Johannes Schulz, Sameh Othman, Anna Badalyan, and Tainá dos Santos for the interesting discussions over the years.

I would like to thank my family for their support, some of whom are very far away, but have shown themselves to be much closer through their support and interest. My parents and siblings, my grandmother, uncles, aunts and cousins. ¡Gracias a todos por estar siempre junto a mi!

I would like to thank my friends for their constant support, their valuable advice, and the great enjoyment they have given me over the past years: Martina Contisciani, Daniela Leite, Kibidi Neocosmos, Nicolo Ruggeri, Alessandro Lonardi, Pablo Sanchez, Laura Iacovisi, Lorenzo Ferretti, and Nicolò Zottino. Their presence has made the PhD journey much smoother. I am also grateful to many friends who, although not mentioned here, have contributed to this achievement with moments of support.

Finally, I would like to thank Miriam Rateike for being an endless source of creativity and passion for science, and for bringing a lot of growth, fun, and love to my PhD path. To you: Thank you! ¡Gracias! Danke! ▼

This thesis is based on the kaobook template by Federico Marotta, which in turn is inspired by Ken Arroyo Ohori's doctoral thesis. I am grateful for their contributions to making this template available.

# Contents

**Abstract** i

**Zusammenfassung** iii

**Acknowledgements** v

**Contents** vii

**1 Introduction** 1
- 1.1 Motivation 1
- 1.2 Outline and Contributions 3

**Preliminaries** 5

**2 Optimal Transport Theory and the Dynamical Monge-Kantorovich Model** 7
- 2.1 From Monge and Kantorovich to Optimal Transport Densities 7
  - 2.1.1 Measure Theory Concepts 7
  - 2.1.2 Gaspard Monge's Transportation Challenge 8
  - 2.1.3 Kantorovich's Relaxation: From Deterministic to Probabilistic 9
  - 2.1.4 The $L^1$-OT Problem: $c(x, y) = |x - y|$ 10
- 2.2 The Dynamical Monge-Kantorovich Model 11
  - 2.2.1 Branched and Congested OT 13

**Published Work** 15

**3 From Continuous Spaces to Discrete Structures** 17
- 3.1 Preamble 17
  - 3.1.1 Graphs 17
  - 3.1.2 The DMK Model on Graphs 18
  - 3.1.3 Biological Motivation 19
- 3.2 Publications 20
  - 3.2.1 Network Extraction by Routing Optimization 20
  - 3.2.2 Convergence Properties of Optimal Transport-based Temporal Networks 21
  - 3.2.3 Convergence Properties of Optimal Transport-based Temporal Hypernetworks 22

**4 Applications on Discrete Spaces** 25
- 4.1 Preamble 25
  - 4.1.1 The Wasserstein distance 25
- 4.2 Publications 26
  - 4.2.1 Community Detection in Networks by Dynamical Optimal Transport Formulation 26
  - 4.2.2 Principled Network Extraction from Images 27
  - 4.2.3 Immiscible Color Flows in Optimal Transport Networks for Image Classification 28

**Discussion** 31

**5 Discussion** 33

**Bibliography** 35

Alphabetical Index 39

Appendix 41

Network Extraction by Routing Optimization 43

Convergence Properties of Optimal Transport-Based Temporal Networks 57

Convergence Properties of Optimal Transport-Based Temporal Hypergraphs 73

Community Detection in Networks by Dynamical Optimal Transport Formulation 89

Principled Network Extraction from Images 101

Immiscible Color Flows in Optimal Transport Networks for Image Classification 115

# Introduction 1

1.1 Motivation . . . . . . . . . . 1
1.2 Outline and Contributions 3

## 1.1 Motivation

Transportation problems play a fundamental role in our daily lives, whether it is us commuting from home to work, companies shipping goods between locations, or municipalities managing underground wastewater flow. When confronted with these challenges, whether in designing or utilizing these transportation systems, we often have different objectives in mind: minimizing commuting time, reducing the environmental impact of shipping, or mitigating risks of transportation failures. When optimization fails, it can lead to detrimental outcomes for our lives, such as arriving late to work, emitting large amounts of CO2, or experiencing wastewater backups in our drains.

Such transport problems can be formally described as seeking to move mass from designated sources to destinations, commonly referred to as sinks, while minimizing transportation costs. Consider the infamous case of Michigan State University (MSU) [1], where the administration aimed to facilitate student mobility between campus buildings by constructing paved pedestrian paths to minimize commuting time. We will use this as a running example throughout this chapter. When pedestrian paths are not optimally planned, it is common for people to bypass official paths and instinctively opt for the most efficient route, often walking over unpaved areas such as grass. These informal routes, known as "desire paths", can pose safety hazards [1] and are thus undesirable for the university's administration. MSU thus commissioned landscape architects to create footpaths efficiently connecting campus buildings, such that they would not be bypassed.

Optimal Transport (OT) is a mathematical framework that allows us to formally describe such problems. OT problems are often highly complex, characterized by numerous sources and sinks, high-dimensional and large solution spaces, as well as various capacity or spatial constraints. As a result, computational approaches to these problems demand sophisticated algorithms capable of managing such complexity. Over the past few decades, research has introduced several computational methods to solve these types of problems efficiently, such as Transportation Simplex [2], Auction algorithm [3], the Shortlist method [4], the Earth Mover's Distance (EMD)-$L_1$ [5], and the Sinkhorn-Knopp algorithm [6].

The architects at MSU did not opt for a computational approach. They left open spaces covered with grass without any footpaths [1]. Over time, they observed students naturally forming informal "desire paths" over the grass area. Later, these pathways were paved and formalized to comply with safety regulations. To this day, these paths continue to honor the naturally chosen efficient routes of the students [1], likely to still result in high compliance with staying on the paved paths. Could the architects have also employed one of the previously enlisted algorithms instead?

While previous algorithms have indeed been successful in solving OT problems across a wide range of applications, such as traffic simulation [7], urban planning [8], and land-use dynamics [9, 10], they encounter some major challenges. First, these algorithms solve OT problems that are inherently defined on discrete spaces. In our example, this would be applicable if architects were utilizing an existing path network and were interested in discovering which paths the students most frequently took. However, in many practical scenarios, the solution space is continuous. In our example, the architects were indeed looking to construct paths anywhere in the open grass area, i.e., in the entire continuous space. Second, in many scenarios, we need to account for the density of resources moved through a transportation system. This is because congestion on overpopulated transportation paths can lead to increased transportation costs [11]. For example, if a large number of students simultaneously attempt to use a narrow footpath to walk between campus buildings, this can lead to traffic congestion resulting in delays for students. The architects may thus need to consider the density of student traffic when planning pavements, i.e., constructing wider footpaths where pedestrian flow is larger, and narrower ones where fewer students walk. The existing methods would thus have been unsuitable for the problem of the architects.

The Dynamic Monge-Kantorovich (DMK) model [12–14] has been developed to address these challenges.[*] It models transportation problems using dynamical equations. First, unlike previous approaches that focus on inherently discrete problems, DMK offers an efficient computational approach that enables solving problems defined over a continuous two-dimensional space. The DMK performs a fine-grained discretization of the continuous solution space and finds numerical solutions using standard finite-element techniques. This effectively yields high-resolution density maps for continuous problems. Second, it allows for imposing traffic constraints such as prioritizing high-traffic paths over many small parallel paths, and the solution includes information on traffic density, which can help prevent congestion. This makes DMK a more practical framework for solving real-world transportation problems. We use our example to illustrate this. The architects allow individuals to move freely around at each centimeter of the grass in the open space, which we can consider as a fine-grained discretization of the solution space. Over time, the students formed the “desire paths”, visible from an aerial perspective as varying shades of green and brown across the open area. This pattern resembles a density map of pedestrian movement. Consequently, architects could use the density information to construct wider paths where more people walked, effectively preventing congestion.

However, DMK also has a few limitations. While in many practical scenarios, the solution space is indeed continuous, planners often seek for a discrete solution, for example, a graph. As mentioned above, the DMK yields a high-resolution density map. While this solution provides a good approximation of the optimal solution, discretizing it to a graph presents a challenge due to the desiderata often associated with graphs, such as ensuring there are no redundant paths. In our example, the landscape architects at MSU also looked for a discrete solution, which

[*] Note that the DMK was introduced in 2018, which marked the start of my research on this topic.

can be seen as a graph of paved paths. Identifying which paths to pave from the density map of worn-out grass involved delineating boundaries and designating some areas of grass as paths, while others, potentially redundant and less frequented paths, were designated as areas not to be walked over. Thus, the density map of "desire paths" was discretized into a graph of formalized footpaths.

Another challenge of the DMK is that it often requires a substantial number of steps (updates) to achieve convergence. This poses challenges for its practical implementation in time-sensitive scenarios like emergency transportation problems of people [15] or drugs [16]. In the MSU example, this means the following. Assume the landscape architects propose to observe grass areas for one year to track "desire paths" before paving. They argue that well-defined paths remain unchanged only after nearly all students habitually utilize them, a process that typically takes about a year. We can think of this as running the DMK algorithm until convergence for 365 time steps. Concerned about safety, the university, however, seeks to minimize the number of days with permitted foot traffic in grass areas and cannot afford to leave them without formalized footpaths for an entire year.

Finally, the DMK has been introduced as a purely mathematical framework, and it remains unclear whether it is suitable for informing real-world applications. Previous OT approaches have proven successful in machine learning tasks extending beyond the conventional problems of transporting goods or people. These approaches have been utilized in shape recognition [4, 17], object detection [18], cross-domain image alignment [19], cell classification [4, 20, 21], and measuring linguistic similarities [4]. The open question of DMK is thus, whether it can be similarly applied to inform various other machine learning tasks beyond traditional optimal transport problems.

In the thesis, we address the aforementioned limitations and open questions of the DMK. We now offer a detailed outline of the thesis and its contributions.

## 1.2 Outline and Contributions

Chapter 2 provides an introduction to OT theory and presents a theoretical foundation for our contributions. It begins with a review of the mathematical discipline and then formally introduces the DMK model.

Chapter 3 examines scenarios where the solution space is continuous but requires a discrete solution, and approaches the problem from a more theoretical perspective. Our first contribution lies in proposing different methods to discretize a density map into graphs and hypergraphs. Graphs consist of edges, each connecting precisely two nodes (vertices), whereas hypergraphs permit hyperedges, which can link multiple vertices. In our MSU example, we can envision a hyperedge as a *plaza* connecting multiple individual paths. The primary distinction from a regular crossing of two paths (through a node) lies in the larger space the plaza (hyperedge) offers, allowing higher numbers of students to move freely across it. This can effectively prevent congestion of frequently used paths and turns.

Secondly, we investigate strategies aimed at reducing the costs and time requirements associated with running the DMK until it reaches convergence. We study early stopping strategies and find that stopping the algorithm after only a few time steps, long before it reaches convergence, leads to only a slight decrease in the optimality of the discretized solution compared to letting the algorithm continue until convergence. Early stopping thus presents an efficient strategy for practical applications constrained by cost and time considerations. Applying our findings to the MSU example means that paths established after, for instance, 50 days—though not yet used by all students, with some still taking their own routes—will show no significant change compared to waiting for the remaining year. Paving the paths after 50 days could thus substantially cut safety costs while preserving path optimality, making them highly likely to be accepted by all students.

Chapter 4 shifts the focus to scenarios where the solution space is discrete. Here, we use a special variant of the DMK algorithm tailored for such settings, known as the graph DMK. A discrete solution space is often prevalent in problems extending beyond classical transportation. For instance, social networks are typically represented by graphs [22]. Consequently using machine learning techniques for analyzing social networks, will require solving problems on graphs.

We demonstrate the versatility of the graph DMK algorithm in three case studies, showcasing how it can effectively contribute to real-world machine learning tasks across different domains. Thereby, Graph DMK offers fast convergence and the ability to accommodate traffic density constraints. Specifically, we use graph DMK for solving clustering problems in social networks under varying traffic rates affecting the spread of social information; for segmenting images by extracting networks from satellite images of rivers or retinal images of blood vessels; and for classifying images by analyzing color flow patterns between them to determine their similarities. Our results demonstrate that the DMK algorithm can effectively inform a great variety of machine learning tasks, underscoring the impact and significance of our research.

In conclusion, this thesis enhances the practicality of DMK for real-world optimal transport problems by identifying practical and efficient solutions to transportation problems. Moreover, our contributions pave the way for future applications of DMK to advance machine learning domains beyond transportation.

# Preliminaries

# Optimal Transport Theory and the Dynamical Monge-Kantorovich Model 2

2.1 From Monge and Kantorovich to Optimal Transport Densities . . . 7
2.1.1 Measure Theory Concepts 7
2.1.2 Gaspard Monge's Transportation Challenge . . . 8
2.1.3 Kantorovich's Relaxation: From Deterministic to Probabilistic . . . . . . . . 9
2.1.4 The $L^1$-OT Problem: $c(x,y) = |x-y|$ . . . . . . 10
2.2 The Dynamical Monge-Kantorovich Model . . . . 11
2.2.1 Branched and Congested OT . . . . . . . . . . . . . . 13

In this chapter, we introduce the basic concepts of Optimal Transport theory (OT) and present the theoretical foundation for our contributions. We begin with a historical overview, highlighting the seminal contributions of Gaspard Monge and Leonid Kantorovich, who formulated the mathematical optimization problem of transporting materials between locations (Section 2.1). We then introduce the *Dynamical Monge-Kantorovich Model* [12] (Section 2.2), which allows us to impose traffic constraints on the solution, such as controlling the capacity (congestion) of transport paths, which affects the spatial extent occupied. We explore this phenomenon in detail in the section on Branched and Congested OT (Section 2.2.1). In this thesis, we study the solutions of these types of OT problems, their properties, and their implications.

## 2.1 From Monge and Kantorovich to Optimal Transport Densities

OT theory provides a robust framework for optimizing resource allocation while minimizing transportation costs. Its roots can be traced back to Gaspard Monge's pioneering exploration of material transport, a foundation later refined and reshaped by Leonid Kantorovich's transformative insights. In the following paragraphs, we examine these influential contributions and highlight the connections between them.

Before going into more detail, however, we present some measure theory concepts that form the basis of the theoretical framework introduced in this chapter.

### 2.1.1 Measure Theory Concepts

The essential concepts for understanding OT theory are taken from the book "Measure Theory and Probability Theory" [23]:

- A *measure*, denoted as $\mu$, is a non-negative set function defined on a sigma-algebra $\Sigma$ of subsets of a set $\Omega$. In our context, we primarily work with measures defined on subsets of $\mathbb{R}^d$. The term *marginal* of a measure $\mu$ refers to the distribution of a measure along a specific dimension or coordinate.
- A *measurable space* is a pair $(\Omega, \Sigma)$ consisting of a set $\Omega$ and a sigma-algebra $\Sigma$ of subsets of $\Omega$.
- A pivotal measure in this context is the *Lebesgue measure*, denoted as $\lambda$, or $\lambda^d$ when explicitly referring to the $d$-dimensional Lebesgue measure. The Lebesgue measure is defined on the Borel $\sigma$-algebra $\mathscr{B}^d$ of subsets of $\mathbb{R}^d$, encompassing sets that can be precisely measured using real numbers. This $\sigma$-algebra

includes all open sets and is generated by open intervals, providing a foundation for rigorous measurement in real analysis. It is commonly used as the standard measure on $\mathbb{R}^d$.

- A function $T : \Omega \to \Omega$ is considered a *measurable map* if the pre-image of any measurable set is measurable, i.e., for all sets $B$ in $\Sigma$, $T^{-1}(B)$ is also in $\Sigma$.
- The *density* function, often denoted as $f$, represents a measure in terms of an integral. Given a measure $\mu$ and the Lebesgue measure $\lambda^d$, the density function $f$ satisfies the Radon-Nikodym derivative condition, which ensures its existence and connection to $\mu$:

  $$\mu(A) = \int_A f d\lambda^d, \quad A \in \Sigma.$$

  Here, $f$ characterizes the distribution and intensity of the measure $\mu$ with respect to the Lebesgue measure $\lambda^d$. The function $f$ is required to be measurable, and it facilitates the integration of various functions involving the measure $\mu$. For the measure $\mu$ and the Lebesgue measure $\lambda^d$ to interact as described in the equation above, $\mu$ must be $\sigma$-finite, ensuring that these mathematical operations are well-defined. Throughout our exploration, we may interchangeably refer to measures $\mu$ and their density functions $f$ using the same notation. In this context, we say that $\mu$ is *absolutely continuous with respect to the Lebesgue measure*.
- To simplify notation, we use $\int f\, dx$ to represent the integral of a function $f$ with respect to the Lebesgue measure, simplifying the notation for clarity.
- A property holds *almost everywhere* (a.e.) if the set of points where the property does not hold has measure zero. In other words, the exceptions are negligible in terms of measure.

### 2.1.2 Gaspard Monge's Transportation Challenge

The foundations of OT theory were laid by Gaspard Monge in 1781 [24], culminating in his seminal formulation. Monge's problem, rooted in practical considerations, addressed the optimal redistribution of material. Specifically, it addressed the strategic challenge of moving matter from one location ("déblais," *excavation materials* in French) to an equivalent volume of terrain elsewhere ("remblais," *fill materials*). The goal was to find the most efficient mass transportation strategy, where the cost of transportation is a function of both mass and distance. We formalize these concepts below.

Let $\Omega \subset \mathbb{R}^2$ be an open, bounded, convex, and connected 2-dimensional domain with a smooth boundary[1]. Let $f^+$ and $f^-$ be two measures defined on $\Omega$ to be transported. Let's assume that the material capacities of both the source and the target (commonly referred to as sink) are identical, denoted by $f^+(\Omega) = f^-(\Omega)$. From now on let's use the term "balanced" to describe this property. Consider the set $\mathcal{T}$ of measurable maps $T : \Omega \to \Omega$ defined as

$$\mathcal{T}(f^+, f^-) := \{T : T_\# f^+ = f^-\},$$

[1] These concepts can be intuitively defined in the following way. We add relevant references next to each item where more precise definitions can be found.

- **Open Set**: A set where each point has a neighborhood entirely contained within the set.
- **Bounded Set**: A set that can be enclosed within a finite-radius ball.
- **Convex Set**: A set where the line segment connecting any two points is entirely within the set.
- **Connected Set**: A set that cannot be separated into disjoint parts.
- **Smooth Boundary**: A boundary free of abrupt irregularities or singularities, approximable by a smooth curve or surface.

More details about these notions can be found in the book "Principles of Mathematical Analysis" [25].

where the *image measure* $T_{\#}f^+$ is defined as $T_{\#}(f^+)(A) := f^+(T^{-1}(A)) \quad \forall A$ measurable set in $\Omega$; we refer to $\mathcal{T}$ as the set of *transport maps*. This set represents all possible ways to transport mass from the source to the target while preserving the mass balance.

Let a *cost function* $c : \Omega \times \Omega \longrightarrow \mathbb{R}_+ := \mathbb{R}^+ \cup \{0\}$ define the expense associated with transporting mass from one point in the domain to another.

Having established this notation, we can proceed to define the first fundamental problem in OT [24]:

**Definition 2.1.1 Monge Problem:** *Given two balanced measures $f^+$ and $f^-$ defined on $\Omega$, and a cost function $c : \Omega \times \Omega \longrightarrow \mathbb{R}_+$, find $T \in \mathcal{T}(f^+, f^-)$ that minimizes the transportation cost functional*

$$\mathcal{I}_c(T) := \int_\Omega c(x, T(x)) df^+(x).$$

### 2.1.3 Kantorovich's Relaxation: From Deterministic to Probabilistic

Despite its elegance, Monge's problem often faced challenges regarding the existence and uniqueness of solutions. Constraints imposed by measure-preserving maps and intricacies within the cost function $c$ could render the problem ill-posed, raising doubts about the feasibility of obtaining the transport maps in $\mathcal{T}$.

Leonid Kantorovich's pivotal contributions during the mid-20th century [26] brought new perspectives into the field by addressing the ill-posed nature of Monge's problem. Kantorovich's pioneering insight shifted the focus from *deterministic* transport maps to *probabilistic* transport plans, thus introducing a more tractable formulation [27].

Consider the set $\Pi(f^+, f^-)$ of all probability distributions $\pi$ on $\Omega \times \Omega$ such that $\pi$ has marginals $f^+$ and $f^-$. Here, $\pi(x, y)$ represents the joint measure that characterizes the transportation plan between $x$ and $y$ within the domain $\Omega$. Kantorovich's formulation can then be stated as:

**Definition 2.1.2 Kantorovich Primal Problem:** *Given two balanced measures $f^+$ and $f^-$ defined on $\Omega$, and a cost function $c : \Omega \times \Omega \longrightarrow \mathbb{R}_+$, find a transport plan $\pi \in \Pi(f^+, f^-)$ that minimizes the functional*

$$\mathcal{K}_c(\pi) := \int_{\Omega\times\Omega} c(x, y) d\pi(x, y).$$

Kantorovich transformed the initial problem, permitting solutions to take the form of probability distributions in the joint space $\Omega \times \Omega$, rather than strict deterministic maps. This ingenious relaxation shifted the focus to *redistribution*, enabling the division of mass from the source to various destinations.

It is worth noting that for any cost function $c : \Omega \times \Omega \longrightarrow \mathbb{R}_+$ which is lower semicontinuous[2], Kantorovich's problem admits a solution, as

[2] **Lower Semicontinuous**: A function is lower semicontinuous if it does not decrease too abruptly; small variations in its input produce only small variations in its output, specifically in the downward direction. [28].

stated in the next theorem (and proved in the work of Kantorovich and Santambrogio [26, 29]):

**Theorem 2.1.1 Admissibility Theorem:** *For any lower semicontinuous cost function $c : \Omega \times \Omega \longrightarrow \mathbb{R}_+$, the Kantorovich Primal Problem admits a solution $\pi \in \Pi(f^+, f^-)$.*

One of the notable advantages of Kantorovich's formulation is its versatility in reformulation:

**Definition 2.1.3 Kantorovich Dual Problem:** *Given two balanced measures $f^+$ and $f^-$, and given a cost function $c : \Omega \times \Omega \longrightarrow \mathbb{R}$. Let $\mathcal{D}_c$ be the set:*

$$\mathcal{D}_c := \{(u, v) \in C_b(\Omega) \times C_b(\Omega)^3 : u(x) + v(y) \leq c(x, y) \, \forall \, (x, y) \in \Omega \times \Omega\}$$

*Find $(u^*, v^*) \in \mathcal{D}_c$ that maximizes the functional:*

$$\mathcal{J}(f^+, f^-)[u, v] := \int_\Omega u(x) df^+(x) + \int_\Omega v(y) df^-(y)$$

While the primal problem focuses on finding the OT plan $\pi$ that minimizes the transportation cost, the dual problem seeks to find functions $u$ and $v$ that maximize the total cost, subject to specific constraints. These dual functions $u$ and $v$ are known as *Kantorovich potentials*.

The duality between the two problems is guaranteed by the following theorem:

**Theorem 2.1.2 Kantorovich Duality:** *Given two balanced measures $f^+$ and $f^-$, and a cost function $c : \Omega \times \Omega \longrightarrow \mathbb{R}_+$ that is lower semicontinuous, the following equality holds:*

$$\min_{\pi \in \Pi(f^+, f^-)} \mathcal{K}_c(\pi) = \max_{(u,v) \in \mathcal{D}_c} \mathcal{J}(f^+, f^-)(u, v)$$

A proof of this theorem can be found in the work of Villani [30].

### 2.1.4 The $L^1$-OT Problem: $c(x, y) = |x - y|$

While Monge and Kantorovich proposed results defined for general spaces $\Omega$ and cost functions $c$, this section focuses on a more specific formulation known as the $L^1$-OT Problem. This formulation is obtained by setting $c(x, y) = |x - y|$, where $| \cdot |$ denotes the Euclidean distance.

Our emphasis on the $L^1$ version of the OT problem stems from its ability to be reformulated into the *Monge-Kantorovich Equations*, outlined in the following definition (see Definition 2.1.4 below). This specific setup serves as the basis for the dynamical formulation at the core of this thesis, introduced in the next section.

**Definition 2.1.4 Monge-Kantorovich Equations:** *Let $f^+$ and $f^-$ be two balanced measures. Assume that they admit $L^1$-densities[4]. Find $\mu^*$ and $u^*$*

[3] We use the notation $C_b(\Omega)$ to reference the space of continuous and bounded functions.

[4] This implies that the functions defining the measures are in $L^1$, i.e., the integral of their absolute values over the entire space is finite.

*that solve the following* ***Monge-Kantorovich equations****:*

$$-div(\mu^*\nabla u^*) = f \ in\ \Omega \tag{2.1}$$

$$|\nabla u^*| \leq 1 \ in\ \Omega \tag{2.2}$$

$$|\nabla u^*| = 1 \ a.e.\ in\ \mu^* > 0 \tag{2.3}$$

*with* $f = f^+ - f^-$.

The divergence $\text{div}(\mu^*\nabla u^*)$ represents the rate at which a quantity flows or spreads out from a given point. The function $\mu^*$ is called *OT density* map; the function $u^*$ is one of the *Kantorovich potentials* mentioned in Definition 2.1.3.

The link between these two problems is established through the resolution of another OT formulation called the *Beckman problem*. We refrain from presenting this problem here, as it involves a level of theory beyond the immediate scope of interest in this thesis. However, we encourage the interested reader to explore the details in the work of Brasco (2010) [31].

## 2.2 The Dynamical Monge-Kantorovich Model

In this chapter, we present a dynamical formulation for the $L^1$-OT problem, which serves as the foundation for the work presented in this thesis. This formulation, as originally conjectured by Facca et al. [12–14], provides solutions to the Monge-Kantorovich equations (Definition 2.1.4) in long-term configurations. The origin of this conjecture is inspired by the model proposed by Tero et al. (2007) [32] and its theoretical connections to Kantorovich's problem, as established by Bonifaci et al. (2012) [33]. Detailed explanations of the concepts of both works are given in Chapter 3.

[32]: Tero et al. (2007), 'A mathematical model for adaptive transport network in path finding by true slime mold'

[33]: Bonifaci et al. (2012), 'Physarum can compute shortest paths'

**Definition 2.2.1 Dynamical Monge-Kantorovich Problem:**

*Let $f^+$ and $f^-$ be two balanced measures. Assume that they are continuous with respect to the Lebesgue measure. Find $(\mu, u) : ([0, +\infty), \Omega) \longrightarrow (\mathbb{R}^+, \mathbb{R})$ that solve the following system of equations, complemented by homogeneous Neumann boundary conditions*[5]*:*

$$-div\,(\mu(t,x)\nabla u(t,x)) = f^+(x) - f^-(x) \tag{2.4}$$

$$\partial_t \mu(t,x) = [\mu(t,x)|\nabla u(t,x)|]^\beta - \mu(t,x) \tag{2.5}$$

$$\mu(0,x) := \mu_0(x) > 0. \tag{2.6}$$

Eq. (2.4) expresses that the divergence $\text{div}(\mu\nabla u)$ is equal to the difference between the source and sink measures, indicating spatial balance. This concept is characterized by the function $q = \mu\nabla u$, following the *Fick-Poiseuille rule*. The Fick-Poiseuille flow [35] is a concept related to the flow of substances in confined spaces, such as fluid flow in small tubes or pipes. It combines principles from Fick's first law (describing diffusion) and Poiseuille's law (explaining viscous flow in pipes). This rule describes how

[5] *Homogeneous Neumann boundary conditions* require that the outward normal derivative of the function $\mu$ is zero at the boundary ($\frac{\partial \mu}{\partial n} = 0$), as formally defined by Evans (2022) [34]. This implies that there is no mass flow across the boundary, which means that no mass enters or leaves the system through the boundaries

mass is transported in response to the concentration gradient and finds applications in microfluidics and related fields. Equation (2.5) dictates the dynamics of the system, where the parameter $\beta$ plays a crucial role in shaping various transportation mechanisms. Depending on its value, the transportation map $\mu$ will establish connections between the source and the sink, utilizing either broad or narrow "roads" in the space. A concise exploration of the impact of this parameter is presented in the forthcoming Section 2.2.1 (Branched and Congested OT) on the facing page. The conductivity distribution $\mu_0$ delineates the initial distribution of the transport density $\mu$.

Solving the Dynamical Monge Kantorovich Problem requires finding the steady state solution $(\mu^*, u^*) : \Omega \to \mathbb{R}_+ \times \mathbb{R}$ for Eqs. (2.4) and (2.5). This solution characterizes the system in the long term, i.e,

$$(\mu^*(x), u^*(x)) = \lim_{t \to +\infty} (\mu(t,x), u(t,x)). \tag{2.7}$$

**Conjecture:** ***DMK Solutions as Solutions to the MK Problem***

*It is conjectured that the OT density $\mu^*$ and the potential $u^*$, as defined in Eq. (2.7), serve as solutions to the Monge-Kantorovich Problem.*

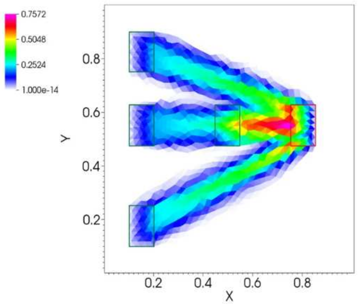


**Figure 2.1:** MK Problem: The plot illustrates the support of the solution $\mu^*$, represented by the color bar. The OT problem involves transporting $f^+$ supported on the green rectangles to the target $f^-$ supported by the red rectangle. The problem is defined in $\Omega = [0,1]^2$, with triangles representing the discretization of the numerical solution space. Plot extracted from our published work [36].

Although a formal proof is yet to be established, the authors have gathered significant theoretical and numerical evidence that supports the proposed idea [12–14].

Under certain continuity conditions on the partial derivatives of $\mu^*$ and $u^*$ [13, 14], the pair $(\mu^*, u^*)$ minimizes the *Lyapunov-candidate* functional $\mathscr{L}$ defined as

$$\mathscr{L}(\mu, u) := \frac{1}{2}\int_\Omega \mu |\nabla u|^2 dx + \frac{1}{2}\int_\Omega \frac{\mu^{P(\beta)}}{P(\beta)} dx, \tag{2.8}$$

where $P(\beta) = (2-\beta)/\beta$. Here, the left side of the functional measures total energy dissipation during transport, while the right side captures the cost of constructing transport infrastructure. The equilibrium solution $(\mu^*, u^*)$ balances energy minimization and transport capacity optimization. We usually call this the transportation cost associated with the solutions, and in the next chapters, we will use it to evaluate their quality.

Solutions to the DMK problem, and hence to the MK equations, are approximated using a finite element scheme that discretizes both space and time [12–14]. This approach combines P1 Finite Elements with forward Euler time stepping. P1 Finite Elements is a method for approximating solutions to partial differential equations (PDEs) by dividing the domain into a mesh of simple shapes (elements) like triangles and solving the equation within each element. Forward Euler time stepping is a numerical method for solving ordinary differential equations (ODEs) by iteratively advancing the solution in small steps forward in time [37]. See Figure 2.1 for a visual representation of the support of the transport density $\mu^*$ obtained as a solution to a synthetically generated MK problem.

### 2.2.1 Branched and Congested OT

In the framework of the Monge-Kantorovich formulation discussed in Definition 2.1.4, the transportation paths used by the solution form straight lines connecting the source and the sink distributions [12]. Notice this, for instance, in the solution shown in Figure 2.1. However, practical transport scenarios often require the ability to either incentivize or penalize mass concentration along these paths. The DMK model, introduced in Section 2.2, allows us to model such situations by adjusting the parameter $\beta$.

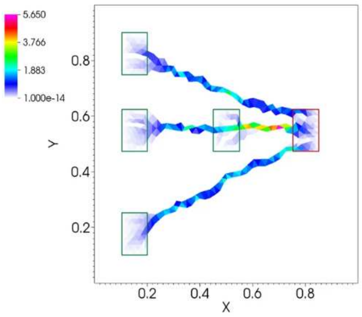


**Figure 2.2:** BT Problem: Solution to the problem introduced in Figure 2.1 found using the DMK model for $\beta = 1.5$. As mentioned, the solution that enforces traffic in a concentrated way. The topology of the support resembles a network, with a fractal-like structure. Plot extracted from our published work [36].

This $\beta$ parameter affects the transport of resources, primarily by dictating the relationship between the transport density map at different points of the domain $\Omega$. This is a consequence of the fact that if we define these as

$$A = q^{\beta}(x_1) + q^{\beta}(x_2), \text{and } \ B = (q(x_1) + q(x_2))^{\beta}, \forall x_1, x_2 \in \Omega,$$

then, for $\beta > 1$, we obtain $A > B$. This indicates that the combined transport of resources proves to be economically more advantageous than their separation. If $\beta \leq 1$, the reverse holds. These two scenarios correspond to the fields of *Branched Transportation* (BT) and *Congested Transportation* (CT) theories, respectively.

In the area of Branched Transportation (BT), a significant challenge lies in the numerical solution of these problems. The BTs are known to have an NP-hard nature [38]. Nonetheless, the presence of the superlinear growth given by $\beta$ incentivates an intense flux of mass from the source to the target distribution. This results in solutions characterized by a "fractal-like" topology, where repetitive patterns become common. An example of such type of solution can be seen in Figure 2.2 for the OT problem presented in Figure 2.1. A detailed description of such behavior can be found in [39, 40].

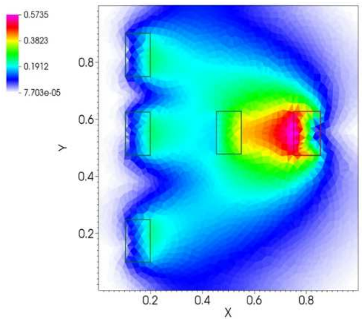


**Figure 2.3:** CT Problem: Solution to the problem introduced in Figure 2.1 found using the DMK model for $\beta = 0.5$. As mentioned, the solution avoids congestion which makes it cover a larger area of the domain. Plot extracted from our published work [36].

The Congested Transport Problem (CT) emerges as a distinct category of OT problems. In CT, the goal is to penalize mass concentration as the transition from $f^+$ to $f^-$ occurs [10]. These types of problems find numerous real-world applications, including the study of urban traffic dynamics and crowd motion [9, 10], where individuals choose which paths to transit by minimizing the number of other individuals they encounter. An example of such behavior can be seen in Figure 2.3 where the solution occupies a wide domain extension.

In the following sections, we look at the practical applications of the DMK model, intending to exploit its versatility to enhance its practicality for solving real-world optimal transport problems.

# Published Work

# From Continuous Spaces to Discrete Structures 3

This chapter examines situations where the solution space is continuous but a discrete solution is required. Building on the insights of Chapter 2, which introduced the DMK model for solving OT problems in continuous 2-dimensional spaces, we make several contributions. Our first paper focuses on proposing different methods to discretize transport density maps into graphs. In our second work, we use these graphs to analyze strategies to decrease the cost and time required to run the DMK to convergence. Furthermore, we take a step beyond by introducing a method that transforms the transport density maps into hypergraphs consisting of hyperedges that can have more than two nodes. Thus, this chapter establishes a link between solutions to the continuous problem and some discrete structures. In the subsequent sections, we will elaborate on these contributions in detail. References supporting these findings are available in the Appendix.

3.1 Preamble . . . . . . . . . . 17
3.1.1 Graphs . . . . . . . . . . . 17
3.1.2 The DMK Model on Graphs . . . . . . . . . . . 18
3.1.3 Biological Motivation . . 19
3.2 Publications . . . . . . . . 20
3.2.1 Network Extraction by Routing Optimization . 20
3.2.2 Convergence Properties of Optimal Transport-based Temporal Networks . . . 21
3.2.3 Convergence Properties of Optimal Transport-based Temporal Hypernetworks 22

## 3.1 Preamble

Before moving into the details of the papers, we provide some essential background. We begin with an introduction to the concept of graphs and related notions, followed by presenting a variant of the DMK model defined on these discrete structures.

### 3.1.1 Graphs

We will establish some formal definitions [41]:

**Definition 3.1.1** *An* ***(undirected) graph***[1]*, denoted as $G = (V, E)$, is composed of two fundamental components: $V$, a nonempty set representing vertices (or nodes), and $E$, a set of edges. These edges link either one or two vertices, referred to as their endpoints, establishing connections between them. We denote by $N$ the number of nodes $|V|$, and by $M$ the number of edges $|E|$.*

**Definition 3.1.2** *Graphs that have a number assigned to each edge are called* ***weighted graphs****. We represent them by $G = (V, E, W)$ where $W$ is the set of weights.*

**Definition 3.1.3** *A* ***subgraph*** *of a graph $G = (V, E)$ is a graph $H = (W, F)$, where $W \subseteq V$ and $F \subseteq E$.*

**Definition 3.1.4** *Consider a nonnegative integer $n$ and an undirected graph $G$. A* ***path*** *from a vertex $u$ to a vertex $v$ in $G$ is defined as a series of $n$ edges, denoted as $e_1, e_2, \ldots, e_n$, where there exists a sequence of vertices $x_0 = u, x_1, x_2, \ldots, x_{n-1}, x_n = v$ such that each edge $e_i$ connects the vertices*

[1] In this context, we use the terms "graph" and "network" interchangeably.

*$x_{i-1}$ and $x_i$, for $i \in [n]$*[2].

Intuitively, a path is a sequence of edges that starts at a vertex within a graph and continues by traversing the edges of the graph from vertex to vertex.

**Definition 3.1.5** *An undirected graph is said to be* ***connected*** *if there is a path between every pair of different vertices in the graph.*

Sometimes alternative representations are used to talk about graphs using matrices, e.g., via their *incidences*[3].

**Definition 3.1.6** *The* ***incidence matrix*** $\mathbf{B}$ *(of size $N \times M$) of an undirected graph $G$ is a binary matrix where each row represents a vertex, each column represents an edge, and each entry $\mathbf{B}_{ij}$ is defined as:*

$$\mathbf{B}_{ij} = \begin{cases} 1, & \text{if vertex } i \text{ is incident to edge } j, \\ 0, & \text{otherwise.} \end{cases}$$

Many of the models presented in the following section use a particular variation of the incidence matrix:

**Definition 3.1.7** *The* ***signed incidence matrix*** *(of size $N \times M$) of an undirected graph $G$ is a ternary matrix where each row represents a vertex, each column represents an edge with a previously chosen orientation, and each entry $\mathbf{B}_{ij}$ is defined as:*

$$\mathbf{B}_{ij} = \begin{cases} 1, & \text{if vertex } i \in V \text{ is the starting point of edge } j \in E, \\ -1, & \text{if vertex } i \in V \text{ is the ending point of edge } j \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Nodes that are incident to only a single edge are known as *leaves* [41]; these will be relevant in the following sections.

### 3.1.2 The DMK Model on Graphs

We now present a special variant of the DMK algorithm tailored for graphs.

Let $\mathbf{B}$ be a *signed incidence matrix* of the weighted graph $G = (V, E, W)$ for a previously chosen orientation of the edges. Let $f$ be an $N$-dimensional vector of source-sink values, with entries ensuring $\sum_{i \in V} f_i = 0$. Analogous to the continuous scenario, we have functions $\mu(t) \in \mathbb{R}^M$ and $u(t) \in \mathbb{R}^N$ representing the *conductivity* and *potential* at time $t$, respectively[4].

[2] We denote by $[n]$ the set $\{1, \ldots, n\}$.

[3] In graph theory, *incidence* indicates which vertices are connected to which edges in a graph.

[4] The functions $\mu(t)$ and $u(t)$ represent the discrete analogs of conductivity and potential in the continuous case (in the Definition 2.2.1).

**Definition 3.1.8 Graph DMK:**
*The graph* ***Dynamical Monge-Kantorovich*** *equations are given by:*

$$f_i = \sum_{e \in E} B_{ie} \frac{\mu_e}{\ell_e} \left(u_i - u_j\right), \qquad e = (i,j),\, \forall i \in V, \tag{3.1}$$

$$\mu_e' = \left[\frac{\mu_e}{\ell_e} |u_i - u_j|\right]^{\beta} - \mu_e \qquad \forall e = (i,j) \in E, \tag{3.2}$$

$$\mu_e(0) > 0, \tag{3.3}$$

*where $\ell_e > 0$ denotes the weight of the edge $e$ and $\beta \in (0,2)$ is a parameter that determines the optimization mechanism.*

Equation (3.1) is Kirchhoff's law; Eq. (3.2) is the discrete dynamics describing the feedback mechanism between conductivity and potentials; Eq. (3.3) is the initial condition. The stationary solution of this dynamical system can be mapped to the solutions of an optimization problem where the cost function can be interpreted as a network transportation cost [42]:

$$\begin{aligned}\mathcal{L}_\beta(\mu(t)) = \quad & \frac{1}{2} \sum_e \mu_e(t) \left(\frac{1}{\ell_e} \sum_j B_{ej} u_j(\mu(t))\right)^2 \ell_e \\ & + \frac{\beta}{2} \sum_e \frac{\mu_e(t)^{(2-\beta)/\beta}}{2-\beta} \ell_e \quad ,\end{aligned} \tag{3.4}$$

where $\mu(t) = \{\mu_e(t)\}_e$. The first term is the network operational cost, while the second represents the cost to build the network, similar to the function defined in Section 2.2.

In particular, the equilibrium point of $\mu(t)$ corresponds to a stationary point of the energy function. For $\beta = 1$, the energy function is convex, making the equilibrium point a global minimizer. However, for $\beta > 1$, the energy function loses its convexity, leading to a more complicated optimization landscape [12].

### 3.1.3 Biological Motivation

The DMK model and its discrete counterpart, the Graph DMK, draw inspiration from a different model [32], which was developed to explore the dynamics of the *Physarum polycephalum* (PP), a fascinating acellular slime mold. This model has been instrumental in understanding the PP's remarkable ability to navigate efficiently between food sources, with a particular emphasis on finding the shortest path. Experimental evidence [43] supports this finding, highlighting the PP's sophisticated pathfinding capabilities.

Building on that foundational work, Bonifaci et al. [33] demonstrated the equivalence of the Graph DMK model for $\beta = 1$ to a specific OT problem, defined on a graph.

**Definition 3.1.9** *Given a forcing term $f$ satisfying $\sum_{v \in V} f_v = 0$, find $Q = \{Q_e\}_{e \in E}$ such that it minimizes the quantity*

$$\sum_{e \in E} Q_e \ell_e, \tag{3.5}$$

*where $\ell_e$ is the weight of the edge $e$.*

They demonstrated that by utilizing the solutions $u$ and $\mu$ for the Graph DMK problem to define a *flux* function, $Q_e := \frac{\mu_e}{\ell_e}(u_i - u_j)$ for $e = (i, j)$, the resulting $Q$ solves the minimization problem outlined in Equation (3.5).

In the following chapters, building on this biological motivation, we compare our results with the behavior of real slime molds.

## 3.2 Publications

After establishing these foundational concepts, we provide an overview of the individual papers and their contributions.

**The Publication**

[42] **Baptista, D.***, Leite, D.*, Facca, E., Putti, M., & De Bacco, C. (2020). Network extraction by routing optimization. *Scientific reports, 10*(1), 20806.
* Equal contributions.

**About the Journal**

**Scientific Reports** is a multidisciplinary journal from the Nature Portfolio that publishes original research from across all areas of the natural sciences, psychology, medicine, and engineering.

### 3.2.1 Network Extraction by Routing Optimization

This section is based on the content of the publication "Network Extraction by Routing Optimization" [42].

As illustrated in Chapter 2, solutions to the original 2-dimensional DMK problem, particularly when $\beta \geq 1$, exhibit network-like structures. In this work, we introduce a methodology to transform these solutions into graphs.

Our approach involves several steps. The first step is to solve the routing problem by computing the steady-state solutions to the DMK equations. The routing optimization problem seeks the steady-state solution $(\mu^*, u^*) : \Omega \rightarrow \mathbb{R}^+ \times \mathbb{R}$ of Eq. (2.5)[5], namely $(\mu^*(x), u^*(x)) = \lim_{t \to +\infty}(\mu(t, x), u(t, x))$.

In the second stage of our process, we present a pre-extraction method to produce an initial network based on the solution of the routing optimization problem. The process begins by defining nodes based on the discretization of the space, typically using the vertices of a finite element mesh. Edges between nodes are then determined based on the values of the optimal transport density $\mu^*$, following specific rules for selecting nodes, edges, and assigning weights. More details on these rules can be found in [42]. These resulting networks are called *pre-extracted networks*; an example of which is shown in Figure 3.1.

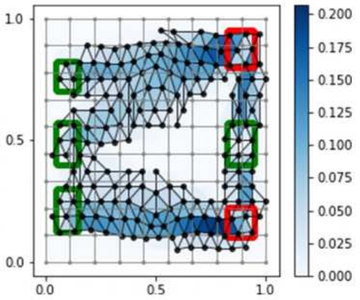


**Figure 3.1:** Pre-extracted network showing the solution of the OT problem between the source $f^+$ (green rectangles) and the sink $f^-$ (red rectangles). The network, depicted in black, is constructed from the support of $\mu^*$, with intensity indicated by the color bar. This plot is extracted from our published work [42].

The networks obtained in the previous step often have more nodes and edges than necessary. In this final step, we remove them by formulating and solving a new routing problem on the pre-extracted graph, using the Graph DMK model (Definition 3.1.8). This process allows us to identify the nodes and edges that are most relevant to the new transport process.

[5] Eq. (2.5) corresponds to the *adaptation equation* mentioned in Definition 2.2.1:

$$\partial_t \mu(t, x) = [\mu(t, x)|\nabla u(t, x)|]^\beta - \mu(t, x).$$

The unused elements are then removed from the graph, resulting in what we refer to as a *filtered graph*.

We provide an implementation of this methodology as a Python tool called *Nextrout*. This tool contains the entire pipeline, simplifying the complex process of network extraction and promoting accessibility for practitioners.

**Personal Contributions to the Paper** I proposed various graph representations during the design phase and introduced a comprehensive quality measure for evaluating the extracted graphs. Additionally, I contributed to writing several sections of the network extraction code and conducted extensive experiments on synthetic data. Moreover, a substantial portion of the paper's content was authored by me.

### 3.2.2 Convergence Properties of Optimal Transport-based Temporal Networks

This section is based on the material covered in the publication "Convergence Properties of Optimal Transport-based Temporal Networks" [36].

As discussed in Chapter 2, the original DMK model seeks to solve an OT problem iteratively. It starts with an initial guess $\mu_0$ and updates it through its equations until reaching the equilibrium state $\mu^*$, thereby solving the transportation problem between source $f^+$ and the target $f^-$. Building on our previous work, we present a method that converts this sequence of updated functions into a family of graphs that evolve over time. Our study of these sequences aims to reduce the cost and time of running the DMK until convergence.

**The Publication**

[36] **Baptista, D.**, & De Bacco, C. (2022). Convergence properties of optimal transport-based temporal networks. In *Complex Networks & Their Applications X: Volume 1, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021 10* (pp. 578-592). Springer International Publishing.

These evolving sequences of graphs, called *temporal graphs*, can be formally defined as:

**Definition 3.2.1** *A* ***temporal graph*** *is a sequence $\{G_t\}_{t\in T}$, where each $G_t$ is a graph, and $T$ is the set of times, often represented as discrete time points or a continuous time interval.*

**About the Conference**

The **International Conference on Complex Networks and their Applications** brings together researchers from diverse scientific communities who are focused on studying various aspects of complex networks. The conference proceedings are published by the multinational publisher, Springer.

Each $G_t$ in the sequence represents the state of the graph at a specific time $t$. Temporal graphs are used to model dynamic systems where the structure of the graph changes over time, capturing the evolution of relationships or interactions among its nodes.

Let $\mu(x,t)$ be a transport density (or conductivity) function that varies with both time and space. This function is obtained as a solution to the DMK model and can be represented as a sequence, denoted as $\{\mu_t\}_{t=0}^T$, typically up to a convergent state at time $T$. Each $\mu_t$ represents an update of our initial guess $\mu_0$, computed by following the rules described in Eqs. (3.1-3.3). We can apply the algorithm described in Section 3.2.1 to extract a network from each of these functions $\mu_t$. This process leads to the creation of a new sequence, denoted as $\{G(\mu_t)\}_{t=0}^T$, which forms a temporal graph.

Once these sequences of networks $\{G(\mu_t)\}_{t=0}^T$ have been built, we study the interplay between the network structure and transport efficiency in synthetically generated OT problems. We compare how the candidate Lyapunov functional $\mathscr{L}$ evaluated on the sequence $\{\mu_t\}_{t=0}^{T}$ compares to the total length[6] of the graphs in the sequence $\{G(\mu_t)\}_{t=0}^{T}$.

Our main finding is that for $\beta > 1$, the total length of graphs in the temporal sequence significantly decreases at times $t_G$ when the total cost is nearing the convergent value ($\mu^*$), yet the number of iterations remains far from convergence ($t_G \ll T$). This suggests that the DMK solver can be stopped at these times, with minimal impact on its optimal properties.

Furthermore, the behavior of the total length of the graph sequences derived from the updates also suggests a dynamic process for optimal network design. This process consists of an initial optimization of transport paths, followed by a consolidation phase in which these paths are compressed, resulting in more efficient network structures.

We further investigate different temporal sequences, focusing in particular on the application of our analysis to networks derived from images capturing changes in the shape of *Physarum polycephalum*. This analysis builds on insights from a methodology introduced in [44][7]. Our observations reveal distinct phases in temporal networks: first an exploration phase, followed by a consolidation phase. These results mirror the patterns observed in synthetic experiments, validating the dynamic mechanism identified for understanding temporal network evolution.

[6] The total length of a graph $G$ is defined as the total sum of the lengths of its edges.

[7] This paper presents a specialized graph extraction method for network-like images, to be discussed in detail in the following chapter.

**Personal Contributions to the Paper** In this collaborative project, I extended the model proposed in the previous section [42] to incorporate temporal sequences, and designed both theoretical developments and empirical experiments. Through experiments with synthetic data and exploration of *Physarum polycephalum* data, I contributed to the validation and testing of our theoretical constructs. In addition, I actively participated in the drafting of the manuscript.

**The Publication**

[45] **Baptista, D.**, & De Bacco, C. (2023). Convergence properties of optimal transport-based temporal hypergraphs. *Applied Network Science*, 8(1), 3.

**About the Journal**

**Applied Network Science** (ANS) is an open-access journal with a rigorous peer-review process, providing researchers and practitioners in the field with an expanded platform to share their work. It is part of the publisher Springer.

### 3.2.3 Convergence Properties of Optimal Transport-based Temporal Hypernetworks

The content of this section is derived from the publication "Convergence Properties of Optimal Transport-based Temporal Hypernetworks" [45].

Building on our previous analysis of the DMK model's solutions and their connections to graphs, we now extend our study to hypergraphs. Hypergraphs, defined as a generalization of graphs, allow edges to interact with more than two nodes simultaneously. Formally, hypergraphs can be defined as [46]:

**Definition 3.2.2** *A* ***hypergraph*** *is a tuple $H = (V_H, E_H)$, where $V_H$ is the nodeset and $E_H$ is the set of hyperedges. Each* ***hyperedge****, denoted as $h_i \in E_H$, is mathematically represented as a non-empty subset of $V$: $h_i \subseteq V$.*

Similar to networks, hypergraphs can have a temporal component. Formally:

**Definition 3.2.3** *A **temporal hypergraph** is a sequence $\{H_t\}_{t\in T}$, each $H_t$ being a hypergraph, and $T$ being the set of times, often expressed as discrete time points or a continuous time interval.*

Let $\mu^*$ be the conductivity derived as a solution to Eqs. (3.1-3.3). Building on our previous work presented on Section 3.2.2 (Convergence Properties of Optimal Transport-based Temporal Networks) on page 21, we propose a method to transform this 2-dimensional function into a hypergraph. This transformation is achieved as follows: consider $G(\mu) = (V_G, E_G)$ as the network extracted according to the methodology proposed in Section 3.2.1. Now we define $H(\mu)$ as the tuple $(V_H, E_H)$, where $V_H = V_G$ and $E_H = E_G \cup T_G$, with $T_G = (u,v,w) : (u,v),(v,w),(w,u) \in E_G$. In simple terms, $H(\mu)$ includes the graph $G(\mu)$ and all its associated triangles. Figure 3.2 illustrates a hypergraph $H(\mu)$ derived from the solution $\mu$ of a synthetically generated transportation problem. The triangles used in the construction are represented in red.

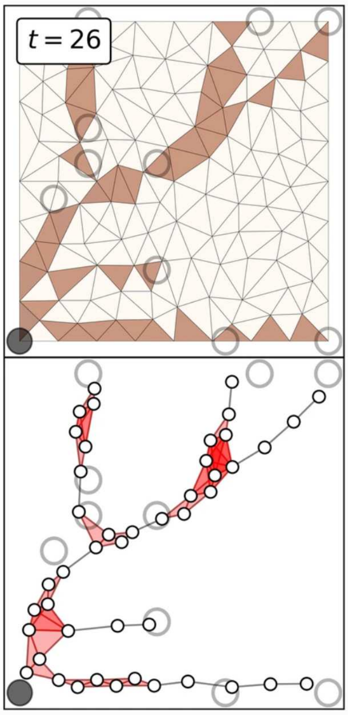


**Figure 3.2:** Hypergraph $H(\mu)$ extracted from the solution $\mu$ of a synthetically generated transportation problem. The black circle denotes the source, and the unfilled circles represent the targets. In red, the hyperedges of size 3. The time $t = 26$ when convergence is reached. This plot is extracted from our published work [45].

This hypergraph construction method applies not only to the convergent state of $\mu^*$ but also to any time step before convergence. Therefore, we can extract a hypernetwork from any intermediate update $\mu_t$. This allows us to represent OT sequences $\{\mu_t\}_{t=0}^{T}$ as temporal hypernetworks $\{H(\mu_t)\}_{t=0}^{T}$.

After constructing the hypergraph sequence, we analyze its coverage on the domain $\Omega$, where the OT problem is defined. This metric, known as *area coverage*, represents the total sum of the area covered by the hypergraph's edges. In our construction, where edges consist of at most three nodes, it equals the sum of the area enclosed by the triangles within the hypergraph.

Our results indicate that the transport costs $\mathcal{L}$ tend to converge faster than the expansion of the coverage in the hypergraphs. This suggests that the DMK model quickly reduces the cost $\mathcal{L}$ of roads connecting source and destination distributions without an immediate reduction in coverage. However, the reduction in coverage becomes more pronounced as the evolution proceeds.

Moreover, we perform a comparative analysis of these higher-order structures with their traditional network equivalents, assessing them in light of conventional graph properties. Our findings indicate that specific transportation schemes, modulated by the parameter $\beta$[8], can benefit from hypernetwork representations. In particular, as the traffic rate increases, the benefits of using higher-order representations decrease. This phenomenon can be attributed to the fact that high traffic rates tend to demand more compact transport schemes, leading to a reduction in the number of nodes required to transport resources.

Following a methodology similar to that used for graphs, we infer hypernetworks from images. We are particularly interested in examining the progression of the area coverage for sequences $H(\mu_t)$ extracted from images documenting the growth of the slime mold *Physarum polycephalum*. Significantly, our analysis shows that denser regions of the slime mold structure tend to decrease in thickness as the organism evolves toward a more uniform state. This observed pattern is consistent with the trends identified in our synthetic data.

[8] The parameter $\beta$, as presented in Section 2.2.1 (Branched and Congested OT) on page 13, dictates the preferred mode of resource transportation from the source to the destination.

**Personal Contributions to the Paper** I modified the model to generate temporal hypernetworks and extensively analyzed their higher-order properties. I carried out a thorough comparison of these properties with those described in a previous publication, which focused on graphs [36].

# Applications on Discrete Spaces 4

4.1 Preamble . . . . . . . . . . 25
4.1.1 The Wasserstein distance 25
4.2 Publications . . . . . . . . 26
4.2.1 Community Detection in Networks by Dynamical Optimal Transport Formulation . . . . . . . . 26
4.2.2 Principled Network Extraction from Images . . . . . . . . . . . . . . . . 27
4.2.3 Immiscible Color Flows in Optimal Transport Networks for Image Classification . . . . . . . 28

Having established a good understanding of the DMK model and its implications for network and hypernetwork extraction in continuous scenarios, we now turn our attention to situations where the solution space is discrete.

In this section, our focus is on showcasing the versatility of the Graph DMK algorithm through three case studies: clustering in networks, network extraction from images, and image classification. These studies serve to illustrate how the algorithm can effectively address diverse machine learning tasks across various domains.

Before presenting the specifics of each case study, we introduce some important preliminary concepts. These fundamental concepts provide the necessary context for understanding the subsequent papers and form the cohesive framework that ties the sections together.

## 4.1 Preamble

This section lays the foundation for the applications discussed later in this chapter. We begin by introducing the necessary theory and general framework for using OT in various applications.

### 4.1.1 The Wasserstein distance

OT allows the comparison of probability distributions, a concept explored in Chapter 2. This involves converting domain-specific data into probability distributions $f^+$ and $f^-$ while defining a base cost $C$ that captures the complexities of the transportation process. Ultimately, OT aims to achieve an optimal alignment between input data, guided by geometric constraints. In the following paragraph, we formalize these concepts and introduce the mathematical counterpart that supports the utility of these tools in our discussion: the *Wasserstein distance*.

By now, it should be clear that solving the discrete OT problem involves finding a function $\gamma^*$ that minimizes the expression:

$$\gamma^* \in \arg\min_{\gamma \in \Pi} \sum_{ij} \gamma_{ij} C_{ij}, \tag{4.1}$$

where $\Pi$ is defined as:

$$\Pi = \left\{ \gamma \,\middle|\, \gamma_{ij} \geq 0 \,\forall i, j; \sum_j \gamma_{ij} = f_i^+ \,\forall i; \sum_i \gamma_{ij} = f_j^- \,\forall j \right\}.$$

Here, $C_{ij}$ represents the cost of transporting mass from location $i$ to location $j$.

The expression at the core of the Eq. (4.1)

$$W_1(f^+, f^-) := \sum_{ij} \gamma^*_{ij} C_{ij}, \tag{4.2}$$

serves as our similarity measure between the input data, effectively quantifying their correspondence level. This similarity measure is known as the *Wasserstein distance*[1] *between the distribution $f^+$ and $f^-$*, a fundamental concept in OT theory.

We use the *flux* function $Q$, derived from the Graph DMK, as the solution to (4.1). Notably, the solution depends parametrically on $\beta$, representing the chosen transport scheme, thus $Q$ becomes $Q(\beta)$. This leads us to extend the Wasserstein distance definition, resulting in a new function denoted as $W_1^\beta$.

## 4.2 Publications

Having laid down this fundamental concept, we proceed to give an outline of each paper and its respective contributions.

### 4.2.1 Community Detection in Networks by Dynamical Optimal Transport Formulation

**The Publication**

[47] Leite, D.*, **Baptista, D.***, Ibrahim, A. A., Facca, E., & De Bacco, C. (2022). Community detection in networks by dynamical optimal transport formulation. *Scientific Reports*, 12(1), 16811.
* Equal contributions.

**About the Journal**

This paper was also published in the **Scientific Reports** journal. More about the journal in Section 3.2.1 (Network Extraction by Routing Optimization) on page 20.

The content of this section is derived from the publication "Community Detection in Networks by Dynamical Optimal Transport Formulation" [47].

In this section, we introduce an OT-based algorithm to discover communities within networks. Consider a weighted graph, $G = (V, E, W)$. We leverage the insights derived from the neighborhood of a node, denoted as $N(i) := \{j \in V \,|\, (i, j) \in E\}$, to determine its community affiliation. This determination depends on a comparison between the distribution defined over $N(i)$ and those associated with neighboring nodes nearby.

As mentioned, we assess the topological similarities between nodes in the network. We achieve this by defining a discrete distribution, denoted as $m_i$, which encapsulates the topological features of each node $i$. We then compare this distribution with those of its neighboring nodes. We postulate that when nodes $i$ and $j$ belong to the same community, their distributions, $m_i$ and $m_j$, should exhibit some degree of similarity. This similarity is quantified by the Wasserstein cost $W_1^\beta(m_i, m_j)$ (as introduced in Eq. (4.2)). It enables us to define a critical metric: the discrete *Ollivier–Ricci curvature* [48], defined as $\kappa_\beta(i, j) := 1 - \frac{W_1^\beta(m_i, m_j)}{d_{ij}}$. This curvature will be instrumental in adjusting the edge weights via the *Ricci flow algorithm* [49]:

$$w_{ij} := d_{ij} - \kappa_\beta(i, j) \cdot d_{ij}.$$

[1] This corresponds to the discrete version of the $W_1$ distance. The generalized $W_p$ distance, as defined in [27], is expressed as:

$$W_p(\mu, \nu) := \left( \inf_{\pi \in \Pi(\mu,\nu)} \int_X d(x, y)^p \, d\pi(x, y) \right)^{1/p},$$

where $p \in [1, \infty)$, and $\Pi(\mu, \nu)$ is the space of all the probability distributions $\pi$ defined on $X \times X$, such that they have marginal distributions $\mu$ and $\nu$.

This iterative process, presented as the *ORC-Nextrout* algorithm, dynamically reshapes the weight distribution of the graph $G$, prioritizing the contraction of intracommunity edges and the expansion of intercommunity ones.

Our algorithm is tested in three different scenarios: synthetic networks, semi-synthetic networks, and real-world networks. To measure the effectiveness of our method in community recovery, we use the *Adjusted Rand Index* (ARI) [50]. The ARI facilitates a comparative evaluation of the community partitions generated by our algorithm with those based on ground-truth clustering.

In our evaluation, our algorithm outperforms the Sinkhorn algorithm [6] [2], especially in scenarios where community detection poses a moderate challenge - neither too easy nor too hard. This highlights the adaptability of our algorithm, allowing users to fine-tune the $\beta$ selection based on performance metrics such as maximizing the ARI, tailored to their specific applications.

The effectiveness of our model is highlighted by evaluations in diverse real-world datasets rich in node metadata, allowing for a thorough assessment of community recovery. These datasets span several domains: co-occurrence networks of characters in "Les Misérables" [51], a social network of bottlenose dolphins [52], a network of Division I American football games in 2000 [22], and a network of US political books around the 2004 presidential election [53]. Comparisons show that OT-based algorithms outperform their counterparts, and match node metadata well in two of the four datasets.

In a final examination, we widen the scope of our comparison between OT-based methods and Infomap[3]. This extended assessment takes place in semi-synthetic scenarios, where random noise is added to existing connections within networks derived from the Les Misérables and Dolphins datasets. In the majority of cases, ORC-Nextrout consistently outperforms the other algorithms in terms of accuracy, demonstrating its robustness in this type of scenario.

**Personal Contributions to the Paper** I contributed to establishing the theoretical framework of our work, developed and tested the codebase, and conducted experiments with real-world data to assess the performance of our method. Additionally, I created some of the figures and improved various sections of the paper to ensure the effective communication of our ideas.

### 4.2.2 Principled Network Extraction from Images

The information in this section is based on the paper "Principled Network Extraction from Images" [44].

Continuing our exploration of the DMK model's versatility, we now shift our focus from complex networks to image analysis. We present a method for extracting graphs from images representing flows.

Exploiting the inherent "discretization" of images by their pixels, we use the RGB color values assigned to them as *conductivities*. Inspired

**The Publication**

[44] **Baptista, D.**, & De Bacco, C. (2021). Principled network extraction from images. *R. Soc. Open Sci.*, 8(7), 210025.

**About the Journal**

The **Royal Society: Open Science** (*R. Soc. Open Sci.*) is an open-access multidisciplinary journal published by the Royal Society, the United Kingdom's National Academy of Sciences.

[2] Sinkhorn, short for Sinkhorn Distance, is a variant of the OT problem that introduces regularization to make computations more tractable. It was introduced by Marco Cuturi in 2013 [6]. The Sinkhorn algorithm iteratively scales the rows and columns of the transportation matrix to approximate the optimal solution of the OT problem.

[3] The Infomap algorithm, also known as the map equation [54, 55], leverages principles of flow and information theory to define a theoretical framework for concisely describing the trajectory of a random walker on a network. It optimizes the map equation, which finds the balance between capturing community structures and minimizing the description length of the random walker's movements.

[42]: Baptista et al. (2020), ‘Network extraction by routing optimization’

by principles established in the continuous case [42], we construct *pre-extracted networks* $G^{pe} = (V, E)$. In this setup, we treat the centroids of pixels as nodes and establish edges between them based on pixel locations and greyscale values. Specifically, an edge is formed when adjacent pixels have a sufficiently high color intensity.

Since this network may still contain redundancies such as dangling nodes or redundant edges, we use the Graph DMK model (defined in Section 3.1.8). This process identifies the most relevant edges to connect pixels of interest, resulting in a new filtered network, denoted as $G^f$.

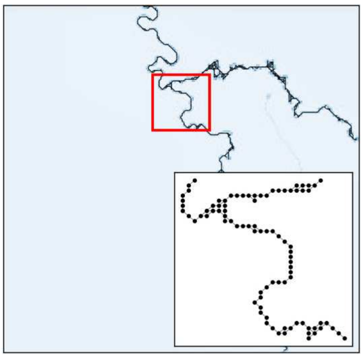


**Figure 4.1:** The black lines represent the network extracted from an image illustrating a river. The inset provides a closer view of a section of the network. This plot is extracted from our published work [44].

Running this dynamic process yields tree-like structures. Yet, network representations in images often contain loops. To address this, we iterate the dynamics multiple times, each time selecting a different eligible node as the source (and designating the rest as sinks). This iterative approach results in a set of filtered networks denoted as $\{G_1^f, \ldots, G_{N_{\text{runs}}}^f\}$. Combining these filtered networks through superposition yields the unique representation we seek. Figure 4.1 shows a reference image together with the network extracted from it.

We put our network-extraction model to the test using three image datasets, each portraying various types of network-like structures commonly observed in biology and ecology. These datasets depict: (i) the *Physarum polycephalum* slime mold [56], which initially inspired our dynamics; (ii) the retinal vascular system [57]; and (iii) river networks obtained from [58]. We compare our approach to methods like NEFI [59] and Minimum Steiner Tree[4]-based techniques. Our evaluation rests on the ability of the algorithms to faithfully recover the network-like structures depicted in the underlying images.

[4] As introduced in [51]:

**Definition 4.2.1** *In a connected weighted graph, a* ***minimum Steiner tree*** *is a tree that connects a specified subset of vertices (the terminals) with the smallest total sum of edge weights.*

Our algorithm outperforms other network extraction tools. It excels at producing networks that closely approximate the structures depicted in the images (see Figure 4.1). In particular, it demonstrates flexibility in recognizing different network shapes, including curved geometries such as those found in river networks.

**Personal Contributions to the Paper** I contributed to designing and implementing the algorithm for network extraction from images, as well as conducting real-data experiments and documenting results. Additionally, I authored many sections of the paper.

**The Publication**

[60] Lonardi, A.*, **Baptista, D.***, & De Bacco, C. (2023). Immiscible color flows in optimal transport networks for image classification. *Frontiers in Physics*, 11, 1089114.
* Equal contributions.

**About the Journal**

**Frontiers in Physics** is a distinguished journal that publishes rigorously peer-reviewed research encompassing the entire field of physics, ranging from experimental investigations to computational and theoretical studies. It is part of the Swiss publishing company, *Frontiers Media SA*.

### 4.2.3 Immiscible Color Flows in Optimal Transport Networks for Image Classification

The content presented in this section is taken from the research paper "Immiscible Color Flows in Optimal Transport Networks for Image Classification" [60].

Here, we introduce an image classification algorithm based on a generalized version of the Graph DMK model, which extends its capabilities to handle multiple sources and sinks in the OT problem.

To begin, let $\mathbf{G}$ and $\mathbf{H}$ represent two images as matrices of sizes $m \times M$ and $n \times M$. Here, $M$ denotes the number of color channels in the images, and

$m$ and $n$ are their sizes, respectively. We start by introducing an auxiliary *bipartite*[5] network, denoted as $K_{m,n}(V_1, V_2, E_{12})$, which is the initial step in solving the OT problem. Here, $V_1$ and $V_2$ represent the pixels of the two images under consideration. The set of edges, $E_{12}$, initially includes all possible connections between the pixels from both images. However, for computational efficiency, we focus on those edges that carry the most relevant information. This selection process involves a combination of thresholding and balancing techniques to retain only the most significant connections.

Once the bipartite graph is constructed, we proceed to solve the OT problem. We achieve this by injecting color mass from the first image into the nodes $i \in V_1$, as defined by matrix $\mathbf{G}$, and extracting it from nodes $j \in V_2$ of the second image, as defined by matrix $\mathbf{H}$.

Given that we have three sources of information for each pair of images $\mathbf{G}$ and $\mathbf{H}$, we employ a variation of the Graph DMK algorithm, known as *Multicommodity DMK* [61]. This extension allows us to handle multiple source and target distributions effectively. The following equations can describe the formulation of Multicommodity DMK:

**Definition 4.2.3 Multicommodity DMK:**
*Let* $\mathbf{B}$ *be a signed incidence matrix of the graph* $G$[6]*. Let* $\{f^{a+}, f^{a-}\}_{a\in[M]}$ *be a family of probability distributions defined on* $V$*. Assume that* $f^a := f^{a+} - f^{a-}$ *satisfies* $\sum_{i\in V} f_i^a = 0, \forall a$*. The Multicommodity Dynamical Monge-Kantorovich equations are given by:*

$$f_i^a = \sum_{e\in E} B_{ie}\frac{\mu_e}{\ell_e}\left(u_i^a - u_j^a\right) \qquad e = (i,j),\ \forall i \in V, a \in [M] \tag{4.3}$$

$$\mu'_e = \mu_e^\beta\left[\frac{||u_i - u_j||_2^2}{\ell_e^2}\right] - \mu_e \qquad \forall e = (i,j) \in E,\text{[7]} \tag{4.4}$$

$$\mu_e(0) > 0, \tag{4.5}$$

*where* $\ell_e > 0$ *denotes the weight of the edge* $e$ *and* $\beta$ *is the parameter that determines the optimization mechanism.*

Notice that, as in the case presented in the previous section, Equation (4.3) corresponds to Kirchhoff's law. Equation (4.4) characterizes the discrete dynamics governing the feedback mechanism between conductivity and flow, while Equation (4.5) outlines the initial condition.

The principal deviation between this model and the one previously discussed is the introduction of multiple source and target distributions for transportation. Additionally, it is worth highlighting that Equation (4.4) employs an alternative exponentiation mechanism compared to the single commodity model. This choice is motivated by the favorable theoretical properties associated with this approach, as detailed in the provided reference [61].

Similarly to that outlined in Definition 3.1.8 , our primary quantities of interest are $\mu_e(t)$ and $u_i(t)$, representing the *conductivity* and *potential*, respectively. These quantities allow us to define a set of *fluxes*, each corresponding to a specific commodity: $Q_e^a := \frac{\mu_e(u_i^a - u_j^a)}{\ell_e}$ for an edge

[5] The definition of a bipartite graph, as outlined in [41], is as follows:

**Definition 4.2.2** *A graph G is considered* ***bipartite*** *if it can be partitioned into two separate sets of vertices, denoted as* $V_1$ *and* $V_2$*, in such a way that every edge in the graph connects a vertex from* $V_1$ *to a vertex in* $V_2$*. Importantly, this partitioning ensures that no edge within the graph links two vertices within either* $V_1$ *or* $V_2$*.*

[6] Notice that the graph $G$ used in this model refers specifically to the auxiliary bipartite graph $K$ introduced in the previous paragraph.

[7] The quantity $||u_i||_2$ represents the 2-norm of $u_i = (u_i^1, \ldots, u_i^M)$ and is mathematically defined as:

$$||u_i||_2 := \left(\sum_a [u_i^a]^2\right)^{1/2}.$$

$e = (i, j)$. These fluxes, in turn, facilitate the computation of a modified *Wasserstein function*: $W_1^\Gamma(G, H) := \sum_{e \in E} ||Q_e||_2^\Gamma C_e$, with $\Gamma = \frac{2-\beta}{3-\beta}$. This Wasserstein measure stands as a pivotal component in quantifying the similarity between the two images **G** and **H**.

To validate the effectiveness of our multicommodity dynamics, we conduct supervised classification tasks using a k-nearest neighbor (k-NN) [62] classifier. We compare the classification accuracy of our model against several alternatives, including i) the Sinkhorn algorithm (using a more stable Sinkhorn scheme [63]); ii) a unicommodity dynamics executed on grayscale images (i.e., with color information compressed into one single commodity, $M = 1$); and iii) the Sinkhorn algorithm applied to grayscale images. All methods are assessed on two datasets: the *Jena Flowers 30 Dataset* [64] and the *Fruit Dataset* [65].

Our method outperforms the other algorithms in terms of classification accuracy. Additionally, our model assigns a lower cost to correctly classified images compared to its unicommodity counterparts, indicating its effectiveness in capturing image color-based similarities and differences.

**Personal Contributions to the Paper** I contributed to the development of the theoretical foundations, participated in the conceptualization and implementation of our method. Additionally, I desgined and executed the experiments needed to validate the model's performance.

# Discussion

# Discussion 5

**Summary** In this manuscript, we study the DMK model, a recently developed optimization algorithm designed to tackle transport problems through dynamical equations. Unlike its predecessors, which primarily address discrete problems, the DMK model introduces an efficient computational method capable of solving challenges in a continuous two-dimensional space. However, despite its advantages, this OT algorithm has two key limitations. While many practical scenarios indeed demand solutions within a continuous space, there remains a prevalent need for discrete solutions, such as graphs. Furthermore, the DMK model, originally introduced as a theoretical framework, requires a deeper understanding of its practical applicability to real-world scenarios. This thesis addresses these limitations by offering practical solutions and insights, thereby enhancing the usability and relevance of the algorithm in real-world applications.

In Chapter 3, we focus on overcoming the first limitation by proposing an approach to network topology extraction. Our method provides a flexible solution capable of processing the numerical solutions of routing optimization problems. It consists of a sequence of three main algorithmic steps: computing steady-state solutions of the DMK equations, extracting optimal network solutions (graph pre-extraction), and filtering redundant structures (graph filtering). Building on this framework, we introduce a method that converts not only the optimal states of the dynamics but also the intermediate updates into temporal graphs. Our main finding highlights an important reduction in the total length of the graphs at convergence points of the total continuous cost, indicating the potential to reduce the number of iterations of the DMK solver with minimal impact on optimality. In addition, our analysis suggests a dynamic process for optimal network design, consisting of an exploration and a compression step. We also propose a method for transforming graphs into temporal hypergraphs. We use this to show that transport costs also converge faster than the coverage area. These transformations (either to graphs or to hypergraphs) of the sequences generated by the DMK model in the search for optimal solutions to transport problems provide new insights into the behavior of these functions and help improve the applicability of the model.

In Chapter 4, we focus on the practical applications. Here we concentrate on scenarios where the solution space is discrete and specifically study a variant of the DMK model known as Graph DMK. We build algorithms that integrate the insights gained from addressing the first limitation and apply them to machine learning tasks such as network clustering, network extraction from images, and image classification.

First, we introduce an OT-based algorithm to discover communities within networks. Our study shows that by exploiting the hyperparameter $\beta$, which governs the transport scheme and influences the flow of information within the network, we could significantly improve the classification accuracy of the algorithm.

Secondly, we propose an efficient method for extracting graphs from images representing flows of mass. Building on the network extraction method presented earlier, we adapt it to use images representing natural transport processes (such as rivers) by exploiting their inherent discretization into pixels. This approach offers a new perspective on graph extraction from real-world images.

Finally, we move one step further into the domain of image analysis by introducing an OT-based classification method that exploits RGB information. Using the Multicommodity DMK algorithm, a variant of Graph DMK, we approach image classification as an OT problem with multiple source and target distributions determined by the different color channels. This approach enhances our ability to classify images effectively and make better use of the information available.

These applications demonstrate the versatility and effectiveness of the DMK model in different domains, highlighting its potential impact and relevance in real-world scenarios.

**Future Research Directions** In terms of future work, this thesis lays a solid foundation for further exploration of other OT algorithms. While the current investigation sheds light on the solutions provided by the DMK model, there is potential for other OT iterative methods to benefit from the discretizations outlined in the previous sections.

Extending the scope of the DMK model opens up new avenues of research. In addition to the structures discussed here, there is potential to explore the extension of the DMK model to other frameworks. While our focus has been on translating solutions into graphs and hypergraphs, or directly using its graph version, there is room to consider introducing a new variation of the DMK model defined directly on other discrete structures, such as generalized hypergraphs (consisting of hyperedges of arbitrary size).

Furthermore, the thesis points to the need for further refinement and innovation in algorithms for community detection and image analysis. Using the knowledge gained from OT-inspired structures, ongoing efforts can be directed toward improving the accuracy and efficiency of these algorithms. From the perspective of community detection, one can think of ways to exploit more broadly the definition of similarity between nodes that uses the Wasserstein measure that we consider in our work. For instance, one could incorporate additional information as covariates on nodes, or alternative definitions of node neighborhoods. From the perspective of image analysis, we have provided here an example of color as the main variable with multiple types of information (the three RGB channels). But one can think of other types of information that could be represented in a multicommodity formalism, and investigate how this could be exploited to improve classification accuracy. These efforts are in line with the broader goal of the thesis, which is to bridge theoretical advances with practical applications, ensuring that theoretical advances contribute directly to addressing real-world challenges.

# Bibliography

Here are the references in citation order.

[1] Christopher Coutts, Ryan Wenger, and Michael Duncan. 'Exploratory Analysis of Revealed Pedestrian Paths as Cues for Designing Pedestrian Infrastructure'. In: *Journal of Urban Planning and Development* 145.4 (2019), p. 05019017 (cited on page 1).

[2] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*. Vol. 2. Springer, 1984 (cited on page 1).

[3] Dimitri P Bertsekas. 'Auction algorithms for network flow problems: A tutorial introduction'. In: *Computational optimization and applications* 1 (1992), pp. 7–66 (cited on page 1).

[4] Carsten Gottschlich and Dominic Schuhmacher. 'The shortlist method for fast computation of the earth mover's distance and finding optimal solutions to transportation problems'. In: *PloS one* 9.10 (2014), e110214 (cited on pages 1, 3).

[5] Haibin Ling and Kazunori Okada. 'An efficient earth mover's distance algorithm for robust histogram comparison'. In: *IEEE transactions on pattern analysis and machine intelligence* 29.5 (2007), pp. 840–853 (cited on page 1).

[6] Marco Cuturi. 'Sinkhorn distances: Lightspeed computation of optimal transport'. In: *Advances in neural information processing systems* 26 (2013) (cited on pages 1, 27).

[7] Yaron Hollander and Ronghui Liu. 'The principles of calibrating traffic microsimulation models'. In: *Transportation* 35 (2008), pp. 347–362 (cited on page 2).

[8] Giuseppe Buttazzo and Filippo Santambrogio. 'A model for the optimal planning of an urban area'. In: *SIAM journal on mathematical analysis* 37.2 (2005), pp. 514–530 (cited on page 2).

[9] Guillaume Carlier and Ivar Ekeland. 'Equilibrium structure of a bidimensional asymmetric city'. In: *Nonlinear Analysis: Real World Applications* 8.3 (2007), pp. 725–748 (cited on pages 2, 13).

[10] Guillaume Carlier, Cristian Jimenez, and Filippo Santambrogio. 'Optimal transportation with traffic congestion and Wardrop equilibria'. In: *SIAM Journal on Control and Optimization* 47.3 (2008), pp. 1330–1350 (cited on pages 2, 13).

[11] Harun Pirim, Umar Al-Turki, and Bekir Sami Yilbas. 'Optimization in Supply Chain'. In: *Supply Chain Management and Optimization in Manufacturing*. Cham: Springer International Publishing, 2014, pp. 43–60. DOI: 10.1007/978-3-319-08183-0_4 (cited on page 2).

[12] Enrico Facca, Franco Cardin, and Mario Putti. 'Towards a Stationary Monge–Kantorovich Dynamics: The Physarum Polycephalum Experience'. In: *SIAM Journal on Applied Mathematics* 78.2 (2018), pp. 651–676 (cited on pages 2, 7, 11–13, 19).

[13] Enrico Facca et al. 'Numerical solution of Monge–Kantorovich equations via a dynamic formulation'. In: *Journal of Scientific Computing* 82.3 (2020), p. 68 (cited on pages 2, 11, 12).

[14] Enrico Facca, Franco Cardin, and Mario Putti. 'Branching structures emerging from a continuous optimal transport model'. In: *Journal of Computational Physics* 447 (2021), p. 110700 (cited on pages 2, 11, 12).

[15] Paolo Toth and Daniele Vigo. 'Fast local search algorithms for the handicapped persons transportation problem'. In: *Meta-heuristics: theory and applications* (1996), pp. 677–690 (cited on page 3).

[16] Judy E Scott and Carlton H Scott. 'Drone delivery models for medical emergencies'. In: *Delivering superior health and wellness management with IoT and analytics* (2020), pp. 69–85 (cited on page 3).

[17] Gabriel Peyré, Marco Cuturi, et al. 'Computational optimal transport: With applications to data science'. In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607 (cited on page 3).

[18] Zheng Ge et al. 'Ota: Optimal transport assignment for object detection'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 303–312 (cited on page 3).

[19] Liqun Chen et al. 'Graph optimal transport for cross-domain alignment'. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1542–1553 (cited on page 3).

[20] Geoffrey Schiebinger et al. 'Reconstruction of developmental landscapes by optimal-transport analysis of single-cell gene expression sheds light on cellular reprogramming'. In: *BioRxiv* (2017), p. 191056 (cited on page 3).

[21] Alexander Tong et al. 'Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics'. In: *International conference on machine learning*. PMLR. 2020, pp. 9526–9536 (cited on page 3).

[22] Michelle Girvan and Mark EJ Newman. 'Community structure in social and biological networks'. In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826 (cited on pages 4, 27).

[23] Krishna B Athreya and Soumendra N Lahiri. *Measure theory and probability theory*. Vol. 19. Springer, 2006 (cited on page 7).

[24] Gaspard Monge. 'Mémoire sur la théorie des déblais et des remblais'. In: *Mem. Math. Phys. Acad. Royale Sci.* (1781), pp. 666–704 (cited on pages 8, 9).

[25] Walter Rudin. *Principles of mathematical analysis*. 1953 (cited on page 8).

[26] L Kantorovich. 'On the transfer of masses (in Russian)'. In: *Doklady Akademii Nauk*. Vol. 37. 1942, p. 227 (cited on pages 9, 10).

[27] Cédric Villani et al. *Optimal transport: old and new*. Vol. 338. Springer, 2009 (cited on pages 9, 26).

[28] Walter Rudin. 'Real and complex analysis. 1987'. In: *Cited on* 156 (1987), p. 16 (cited on page 9).

[29] Filippo Santambrogio. 'Optimal transport for applied mathematicians'. In: *Birkäuser, NY* 55.58-63 (2015), p. 94 (cited on page 10).

[30] Cédric Villani. *Topics in optimal transportation*. Vol. 58. American Mathematical Soc., 2021 (cited on page 10).

[31] Lorenzo Brasco. 'Geodesics and PDE methods in transport models'. PhD thesis. Université Paris Dauphine-Paris IX, 2010 (cited on page 11).

[32] Atsushi Tero, Ryo Kobayashi, and Toshiyuki Nakagaki. 'A mathematical model for adaptive transport network in path finding by true slime mold'. In: *Journal of theoretical biology* 244.4 (2007), pp. 553–564 (cited on pages 11, 19).

[33] Vincenzo Bonifaci, Kurt Mehlhorn, and Girish Varma. 'Physarum can compute shortest paths'. In: *Journal of theoretical biology* 309 (2012), pp. 121–133 (cited on pages 11, 19).

[34] Lawrence C Evans. *Partial differential equations*. Vol. 19. American Mathematical Society, 2022 (cited on page 11).

[35] Adolph Fick. 'V. On liquid diffusion'. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 10.63 (1855), pp. 30–39. DOI: 10.1080/14786445508641925 (cited on page 11).

[36] Diego Baptista and Caterina De Bacco. 'Convergence properties of optimal transport-based temporal networks'. In: *Complex Networks & Their Applications X: Volume 1, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021 10*. Springer. 2022, pp. 578–592 (cited on pages 12, 13, 21, 24).

[37] Carlos A Felippa. 'Introduction to finite element methods'. In: *University of Colorado* 885 (2004) (cited on page 12).

[38] Geoffrey M Guisewite and Panos M Pardalos. 'Algorithms for the single-source uncapacitated minimum concave-cost network flow problem'. In: *Journal of Global Optimization* 1 (1991), pp. 245–265 (cited on page 13).

[39] Qinglan Xia. 'Optimal paths related to transport problems'. In: *Communications in Contemporary Mathematics* 5.02 (2003), pp. 251–279 (cited on page 13).

[40] Qinglan Xia. 'Numerical simulation of optimal transport paths'. In: *2010 Second International Conference on Computer Modeling and Simulation*. Vol. 1. IEEE. 2010, pp. 521–525 (cited on page 13).

[41] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 5th. McGraw-Hill Higher Education, 2002 (cited on pages 17, 18, 29).

[42] Diego Baptista et al. 'Network extraction by routing optimization'. In: *Scientific reports* 10.1 (2020), p. 20806 (cited on pages 19, 20, 22, 28).

[43] Toshiyuki Nakagaki, Hiroyasu Yamada, and Ágota Tóth. 'Maze-solving by an amoeboid organism'. In: *Nature* 407.6803 (2000), pp. 470–470 (cited on page 19).

[44] Diego Baptista and Caterina De Bacco. 'Principled network extraction from images'. In: *Royal Society Open Science* 8.7 (2021), p. 210025 (cited on pages 22, 27, 28).

[45] Diego Baptista and Caterina De Bacco. 'Convergence properties of optimal transport-based temporal hypergraphs'. In: *Applied Network Science* 8.1 (2023), p. 3 (cited on pages 22, 23).

[46] Federico Battiston et al. 'Networks beyond pairwise interactions: Structure and dynamics'. In: *Physics Reports* 874 (2020). Networks beyond pairwise interactions: Structure and dynamics, pp. 1–92. DOI: https://doi.org/10.1016/j.physrep.2020.05.004 (cited on page 22).

[47] Daniela Leite et al. 'Community detection in networks by dynamical optimal transport formulation'. In: *Scientific Reports* 12.1 (2022), p. 16811 (cited on page 26).

[48] Yann Ollivier. 'Ricci curvature of Markov chains on metric spaces'. In: *Journal of Functional Analysis* 256.3 (2009), pp. 810–864 (cited on page 26).

[49] Jürgen Jost and Shiping Liu. 'Ollivier's Ricci curvature, local clustering and curvature-dimension inequalities on graphs'. In: *Discrete & Computational Geometry* 51.2 (2014), pp. 300–322 (cited on page 26).

[50] Lawrence Hubert and Phipps Arabie. 'Comparing partitions'. In: *Journal of classification* 2 (1985), pp. 193–218 (cited on page 27).

[51] Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. Vol. 1. AcM Press New York, 1993 (cited on pages 27, 28).

[52] David Lusseau et al. 'The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait?' In: *Behavioral Ecology and Sociobiology* 54 (2003), pp. 396–405 (cited on page 27).

[53] V. Krebs. *Books about US Politics Dataset*. www.orgnet.com. Unpublished dataset (cited on page 27).

[54] Martin Rosvall and Carl T Bergstrom. 'Maps of random walks on complex networks reveal community structure'. In: *Proceedings of the national academy of sciences* 105.4 (2008), pp. 1118–1123 (cited on page 27).

[55] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. 'The map equation'. In: *The European Physical Journal Special Topics* 178.1 (2009), pp. 13–23 (cited on page 27).

[56] Michael Dirnberger, Kurt Mehlhorn, and Tim Mehlhorn. 'Introducing the slime mold graph repository'. In: *Journal of Physics D: Applied Physics* 50.26 (2017), p. 264001 (cited on page 28).

[57] AD Hoover, Valentina Kouznetsova, and Michael Goldbaum. 'Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response'. In: *IEEE Transactions on Medical imaging* 19.3 (2000), pp. 203–210 (cited on page 28).

[58] *Openseamap*. Accessed from 7–28 September 2020 (cited on page 28).

[59] Michael Dirnberger, Tim Kehl, and A Neumann. 'NEFI: Network extraction from images'. In: *Scientific reports* 5.1 (2015), p. 15669 (cited on page 28).

[60] Alessandro Lonardi, Diego Baptista, and Caterina De Bacco. 'Immiscible color flows in optimal transport networks for image classification'. In: *Frontiers in Physics* 11 (2023), p. 65 (cited on page 28).

[61] Alessandro Lonardi, Mario Putti, and Caterina De Bacco. 'Multicommodity routing optimization for engineering networks'. In: *Scientific reports* 12.1 (2022), p. 7474 (cited on page 29).

[62] Evelyn Fix and Joseph Lawson Hodges. 'Discriminatory analysis. Nonparametric discrimination: Consistency properties'. In: *International Statistical Review/Revue Internationale de Statistique* 57.3 (1989), pp. 238–247 (cited on page 30).

[63] Bernhard Schmitzer. ‘Stabilized sparse scaling algorithms for entropy regularized transport problems’. In: *SIAM Journal on Scientific Computing* 41.3 (2019), A1443–A1481 (cited on page 30).

[64] Marco Seeland et al. *Jena Flowers 30 Dataset*. Version V1. 2017. DOI: 10.7910/DVN/QDHYST (cited on page 30).

[65] Priscila Alves Macanhã et al. ‘Handwritten Feature Descriptor Methods Applied to Fruit Classification’. In: *Information Technology - New Generations*. Ed. by Shahram Latifi. Cham: Springer International Publishing, 2018, pp. 699–705 (cited on page 30).

# Alphabetical Index

$L^1$-OT Problem, 10
$L^1$-densities, 11
*Physarum polycephalum*, 28

abstract, i
acknowledgements, v
Adjusted Rand Index (ARI), 27
Admissibility Theorem, 10
almost everywhere, 8

balanced measures, 9
Beckman Problem, 11
binary matrix, 18
bipartite graph, 29
bounded set, 8
Branched Transportation (BT) theory, 13

community detection, 26
Congested Transportation (CT) theory, 13
Conjecture, 12
connected graph, 18
connected set, 8
convex set, 8
cost function, 9

discrete dynamics, 19
dual problem, 10
Dynamical Monge-Kantorovich model, 11

Euclidean distance, 10

Fick-Poiseuille rule, 12
flux, 20
Fruit Dataset, 30

graph, 17
Graph DMK, 19
graph sequence, 21, 23

homogeneous Neumann boundary conditions, 11
hypergraph, 22

image measure, 9
incidence matrix, 18
incidences, 18
Infomap, 27
integral, 8

Jena Flowers 30 Dataset, 30
joint distributions, 9

Kantorovich, 9
Kantorovich Dual Problem, 10
Kantorovich Duality, 10
Kantorovich potentials, 10
Kantorovich Primal Problem, 9
Kirchhoff's law, 19

leaves, 18
Lebesgue measure, 7
lower semicontinuous, 10
Lyapunov functional $\mathscr{L}$, 12

measurable map, 8
measure, 7
Minimum Steiner Tree, 28
Monge problem, 9
Monge-Kantorovich equations, 11
Monge-Kantorovich Problem, 11
Multicommodity Graph DMK, 29

NEFI, 28
network, 17
NP-hard, 13

Ollivier-Ricci curvature, 26
open set, 8

path, 17
Physarum polycephalum, 19
probabilistic transport plans, 9
probability distributions, 9

retinal vascular system, 28
Ricci flow algorithm, 26
river networks dataset, 28

signed incidence matrix, 18
Sinkhorn algorithm, 27, 30
slime mold, 28
smooth boundary, 8
subgraph, 17

temporal graph, 21, 23
ternary matrix, 18
transport maps, 9

unicommodity DMK, 30

Wasserstein cost, 26
Wasserstein distance, 25, 30
weighted graph, 17

zusammenfassung, iii

# Appendix

www.nature.com/scientificreports

scientific reports

# Network extraction by routing optimization

Diego Baptista[1,4], Daniela Leite[1,4], Enrico Facca[2], Mario Putti[3] & Caterina De Bacco[1]✉

**Routing optimization is a relevant problem in many contexts. Solving directly this type of optimization problem is often computationally intractable. Recent studies suggest that one can instead turn this problem into one of solving a dynamical system of equations, which can instead be solved efficiently using numerical methods. This results in enabling the acquisition of optimal network topologies from a variety of routing problems. However, the actual extraction of the solution in terms of a final network topology relies on numerical details which can prevent an accurate investigation of their topological properties. In fact, in this context, theoretical results are fully accessible only to an expert audience and ready-to-use implementations for non-experts are rarely available or insufficiently documented. In particular, in this framework, final graph acquisition is a challenging problem in-and-of-itself. Here we introduce a method to extract network topologies from dynamical equations related to routing optimization under various parameters' settings. Our method is made of three steps: first, it extracts an optimal trajectory by solving a dynamical system, then it pre-extracts a network, and finally, it filters out potential redundancies. Remarkably, we propose a principled model to address the filtering in the last step, and give a quantitative interpretation in terms of a transport-related cost function. This principled filtering can be applied to more general problems such as network extraction from images, thus going beyond the scenarios envisioned in the first step. Overall, this novel algorithm allows practitioners to easily extract optimal network topologies by combining basic tools from numerical methods, optimization and network theory. Thus, we provide an alternative to manual graph extraction which allows a grounded extraction from a large variety of optimal topologies. The analysis of these may open up the possibility to gain new insights into the structure and function of optimal networks. We provide an open source implementation of the code online.**

Investigating optimal network topologies is a relevant problem in several contexts, with applications varying from transportation networks[1–4], communication systems[5–7], biology[8,9] and ecology[10–12]. Depending on the specified objective function and set of constraints of a routing optimization problem[13], optimal network topologies can be determined by different processes ranging from energy-minimizing tree-like structures ensuring steeper descent through a landscape as in river basins[10] to the opposite scenario of loopy structures that favor robustness to fluctuations and damage as in leaf venation[12,14], the retina vascular system[15,16] or noise-cancelling networks[7].

In many applications, optimal networks can arise from an underlying process defined on a continuous space rather than a discrete network as in standard combinatorial optimization routing problems[17–20]. Optimal routing networks try to move resources by minimizing the transportation cost. This cost may be taken to be a function of the traveled distance, such as in Steiner trees, or proportional to the dissipated energy, such as optimal channel networks or resistance networks. The common denominator of these configurations is that they have a tree-like shape, i.e., optimal routing networks are loopless[1,21]. Recent developments in the mathematical theory of optimal transport[11,13] have proved that this is indeed the case and that complex fractal-like networks arise from branched optimal transport problems[22]. While the theory starts to consolidate, efficient numerical methods are still in a pre-development stage, in particular in the case of branched transport, where only a few results are present[23,24], reflecting the obstacle that all these problems are NP-hard. Recent promising results[25,26] map a computationally hard optimization problem into finding the long-time behavior of a system of dynamic partial differential equations, the so-called Dynamic Monge-Kantorovich (DMK) approach, which is instead numerically accessible, computationally efficient, and leads to network shapes that resemble optimal structures[27]. Working in discretized continuous space, and in many network-based discretizations such as lattice-like networks as well, requires the use of threshold values for the identification of active network edges. This has the main consequence that there

[1]Max Planck Institute for Intelligent Systems, Cyber Valley, 72076 Tübingen, Germany. [2]Centro di Ricerca Matematica Ennio De Giorgi, Scuola Normale Superiore, Piazza dei Cavalieri, 3, Pisa, Italy. [3]Department of Mathematics "Tullio Levi-Civita", University of Padua, via Trieste 63, Padua, Italy. [4]These authors contributed equally: Diego Baptista and Daniela Leite. ✉email: caterina.debacco@tuebingen.mpg.de

might be no obvious final resulting network, an output that would be trivial when starting from an underlying search space formed by predefined selected network structures. For example, the output of a numerically discretized (by, e.g., the Finite Element method) routing optimization problem in a 2D space is a real-valued function on a set of $(x, y)$ points defined on a grid or triangulation, which already has a graph structure. Despite the underlying graph, this grid function contains numerous side features, such as small loops and dangling vertices, that prevent the recognition of a clear optimal network structure. Obtaining this requires a suitable identification of vertices and edges that should contain the optimal network properties embedded in the underlying continuous space. In other words, the output of a routing optimization problem in continuous space carries unstructured information about optimality that is hard to interpret in terms of network properties. Extracting a network topology from this unstructured information would allow, on one hand, better interpretability of the solution and enable the comparison with networks resulting from discrete space. On the other hand, the use of tools from network theory to investigate optimality properties, for instance, to perform clustering or classification tasks based on a set of network features.

One can frame this problem as that of properly compressing the information contained in the "raw" solution of a routing problem in continuous space into an interpretable network structure while preserving the important properties connected to optimality. This is a challenging task, as compression might result in losing important information. The problem is made even more complex because one may not know in advance what are the relevant properties for the problem at hand, a knowledge that could help drive the network extraction procedure. This is the case for any real network, where the intrinsic optimality principle is elusive and can only be speculated about by observing trajectories, an approach adopted for instance when processing images in biological networks[28–31].

Several works have been proposed to tackle domain-specific network extraction. These methods include using segmentation techniques on a set of image pixels to extract a skeleton[28,29,32] that is then converted into a network; a pipeline combining different segmentation algorithms building from OpenCV[33], which is made available with an intuitive graphical interface[34]; graph-based techniques[35] that sample junction-points from input images; methods that use deep convolutional neural networks[36] or minimum cost path computations[37] to extract road networks from images. These are mainly using image processing techniques as the input is an image or photograph, which might not necessarily be related to a routing optimization problem. In this work, we propose a new approach for the extraction of network topologies and build a protocol to address this problem. This can take in input the numerical solution of a routing optimization problem in continuous space as described in[25–27] and then processes it to finally output the corresponding network topology in terms of a weighted adjacency matrix. However, it can also be applied to more general inputs, such as images, which may not necessarily come from the solution of an explicit routing transportation problem. Specifically, our work features a collection of numerical routines and graph algorithms designed to extract network structures that can then be properly analyzed in terms of their topological properties. The extraction pipeline consists in a sequence of three main algorithmic steps: (i) compute the steady-state solutions of the DMK equations (*DMK-Solver*); (ii) extract the optimal network solution of the routing optimization problem (*graph pre-extraction*); (iii) filter the network removing redundant structure (*graph filtering*). While for this work we test and demonstrate our algorithm on routing scenarios coming from DMK, which constitute our main motivation, we remark that only the first step is specific to these, whereas the last two steps are applicable beyond these settings. The graph pre-extraction step consists of a set of rules aiming at building a network from an input that is not explicitly a topological structure made of nodes and edges. The filtering step is based on a principled mathematical model inspired by that of the first step, which leads to an efficient algorithmic implementation. Our network filter has a nice interpretation in terms of a cost function that interpolates between an operating cost and an infrastructure one, contrarily to common approaches used in image processing for filtering, which often relies on heuristics. Our numerical approach is based on finite element-like solvers that transform the problem into a finite sequence of linear systems with dimension equal to the number of nodes in the network. Using a careful combination of efficient numerical solvers, the high computational efficiency of our implementation allows addressing large scale problems, out of reach for standard methods of combinatorial optimization. In addition, the algorithmic complexity of our approach is independent of the number of sources and sinks, unlike more standard methods based on Steiner tree solvers[38,39].

A successful execution will return a representation of the network in terms of an edge-weighted undirected network. The resulting weights are related to the optimal flow, solution of the routing problem. Once the network is obtained, practitioners can deploy arbitrary available network analysis software[40–43] or custom-written scripts to investigate properties of the optimal topologies. For instance, given that our model easily adapts to receive images as input, a promising application is that of extracting optimal network topologies from biological networks, in particular in systems that display a dynamic behavior of self-optimization, as recently found this being the case for neuronal networks[44]. Note that our optimal transport-based approach naturally calculates Wasserstein-type distances between discrete measures on the network. This can be used, like other geometric approaches in network analysis, to address different network-related applications, for example for geometry-based community detection algorithms[45–47]. While our primary goal is to provide a framework and tool to solve the research question of how to extract network topologies resulting from routing optimization problems in continuous space or any other image containing a network structure, we also aim at encouraging non-expert practitioners to automatically extract networks from such problems or from more general settings beyond that. Thus we make available an open-source algorithmic implementation and executables of this work at https://github.com/Danielaleite/Nextrout.

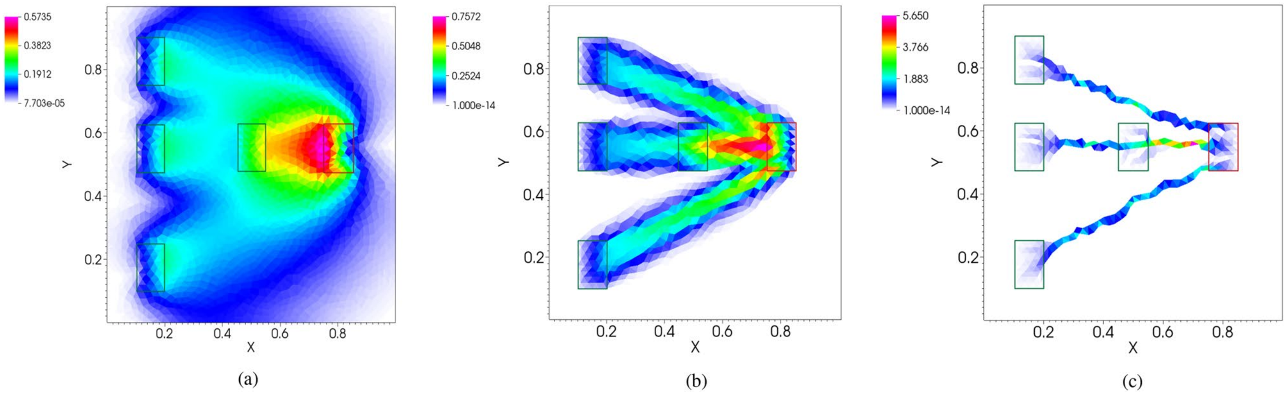


**Figure 1.** Different values of $\beta$ in Eq. (2) lead to different settings of a routing optimization problem. Colors denote different intensities of conductivity $\mu$ as described by the color bar on the left. (**a**) $\beta < 1$ enforces avoiding mass congestion ($\beta = 0.75$); (**b**) $\beta = 1$ is shortest path-like, the mass goes straight from source to sink; (**c**) $\beta > 1$ encourages traffic consolidation ($\beta = 1.2$). Red rectangle denotes the sink, green ones the four sources.

## The routing optimization problem

In this section, we describe the main ideas and establish notation. We start by introducing the dynamical system of equations corresponding to the DMK routing optimization problem as proposed by Facca et al.[25–27] In these works, the authors first generalize the discrete dynamics of the slime mold *Physarum Polycephalum* (PP) to a continuous domain; then they conjecture that, like its discrete counterpart, its solution tends to an equilibrium point which is the solution of the Monge-Kantorovich optimal mass transport[48] as time goes to infinity.

We denote the space where the routing optimization problem is set as $\Omega \in \mathbb{R}^n$, an open bounded domain that compactly contains $f(x) = f^+(x) - f^-(x) \in \mathbb{R}$, the forcing function, describing the flow generating sources $f^+(x)$ and sinks $f^-(x)$. It is assumed that the system is isolated, i.e., no fluxes are entering or exiting the domain from the boundary. This imposes the constraint $\int_\Omega f(x)dx = 0$ to ensure mass balance. It is assumed that the flow is governed by a transient Fick-Poiseuille type flux $q = -\mu\nabla u$, where $\mu(t, x)$, $u(t, x)$ are called *conductivity* or *transport density* and *transport potential*, respectively.

The continuous set dynamical Monge-Kantorovich (DMK) equations are given by:

$$-\nabla \cdot (\mu(t,x)\nabla u(t,x)) = f^+(x) - f^-(x)\,, \tag{1}$$

$$\frac{\partial \mu(t,x)}{\partial t} = [\mu(t,x)\nabla u(t,x)]^\beta - \mu(t,x)\,, \tag{2}$$

$$\mu(0,x) = \mu_0(x) > 0\,, \tag{3}$$

where $\nabla = \nabla_x$. Equation (1) states the spatial balance of the Fick-Poiseuille flux and is complemented by no-flow Neumann boundary conditions; Eq. (2) enforces the system dynamics in analogy with the discrete PP model and Eq. (3) provides the initial configuration of the system. The parameter $\beta$ captures different routing transportation mechanisms. A value of $\beta < 1$ enforces optimal solutions to avoid traffic congestion; $\beta = 1$ is shortest path-like; while $\beta > 1$ encourages consolidating the flow so to use a smaller amount of network-like infrastructure, and is related to branched transport[11,49]. Within a network-like interpretation, qualitatively, $\mu(x, t)$ describes the capacity of the network edges. With hydraulic interpretation, we can think of the edges as pipes, small cylindrical channels where the mass is passing through, and the capacity is proportional to the size of the pipe diameter. Thus, its initial distribution $\mu_0(x)$ describes how the initial capacities are distributed.

In this work, solving the routing optimization problem consists of finding the steady state solution $(\mu^*, u^*) : \Omega \to \mathbb{R}_{\geq 0} \times \mathbb{R}$ of Eq. (1), i.e. $(\mu^*(x), u^*(x)) = \lim_{t\to+\infty}(\mu(t, x), u(t, x))$. Numerical solution of the above model can be obtained by means of a double discretization in time and space[25–27]. The resulting solver (called from now on *DMK-Solver*) has been shown to be efficient, robust and capable of identifying the typically singular structures that arise from the original problem. In Fig. 1, some visual examples of the numerical $\mu^*$ obtained for different values of $\beta$ are shown. The same authors showed that the *DMK-Solver* is able to emulate the results for the discrete formulation of the PP model proposed by Tero et al.[50]

Under appropriate regularity assumptions, it can be shown[26,27] that the equilibrium solution of the above problem $(\mu^*(x), u^*(x))$ is a minimizer of the following functional:

$$\mathscr{L}(\mu, u) = \frac{1}{2}\int_\Omega \mu|\nabla u|^2 dx + \int_\Omega \frac{\mu^{P(\beta)}}{P(\beta)}dx\,, \tag{4}$$

where $P(\beta) = (2 - \beta)/\beta$. In words, this functional is the sum of the total energy dissipated during transport (the first term is the Dirichlet energy corresponding to the solution of the first PDE) plus a nonlinear (sub-additive) function of the total capacity of the system at equilibrium. In terms of costs, this functional can be interpreted as the cost of transport, assumed to be proportional to the total dissipated energy, and the cost of building the transport infrastructure, assumed to be a nonlinear function (with power $2-\beta$) of the total transport capacity of the system.

We exploit the robustness of this numerical solver to extract the solutions of DMK equations corresponding to various routing optimization problems. We here focus on the case $\beta \geq 1$, where the approximate support of $\mu^*$ displays a network-like structure. This is the first step of our extraction pipeline, which we denote as *DMK-Solver*. The numerical solution of these equations does not allow for a straightforward network representation. Indeed, depending on various numerical details related to the spatial discretization and other parameters, one usually obtains a visually well-defined network structure (see Fig. 1) whose rendering as a graph object is however uncertain and non-unique. This in turns can hinder a proper investigation of the topological properties associated to routing optimization problems, motivating the main contribution of our work: the proposal of a graph extraction pipeline to automatically and robustly extract network topologies from the solutions' output of *DMK-Solver*. We reinforce that our contribution is not limited to this application, but is also able to extract network-like shapes from any kind of image where a color or greyscale thresholds can be used to identify the sought structure.

Our extraction pipeline then proceeds with two main steps: *pre-extraction* and *graph filtering*. The first one tackles the problem of translating a solution from the continuous scenario into a graph structure, while the second one addresses the problem of removing redundant graph structure resulting from the previous step. A pseudo-code of the overall pipeline is provided in Algorithm 1. In that pseudo-code, mesh-related parameters specify how the mesh for the discretization of space is built. Specifically, we could specify *ndiv*, the number of divisions in the $x$ axis and *nref*, the number of refinements, i.e. the number of times each triangle on the grid generated by a specific *ndiv* is subdivided into four triangles.

**Algorithm 1** Generating optimal networks from solutions of dynamical systems: the pipeline

```
Input: parameters to set up network extraction problem:
  i) Set the space $\Omega$: $\{T_i\}_i$ grid triangulation and mesh-related parameters.
  ii) Set up routing optimization problem: $\beta \geq 1$, $\mu_0$, $f^+$, $f^-$ and threshold $\delta$.
  iii) Set up the discrete routing optimization (for graph filtering): $\beta_d$, $\delta_d$ and $\tau_{BC}$.
Output: $G(V,E,W)$ final network
1.   Run DMK-Solver: outputs $(\mu^*, u^*)$
2.   Graph pre-extraction: outputs $G(V,E,W)$ with possible redundancies
3.   Graph filtering: removes redundancies from $G(V,E,W)$
```

Our final goal is to translate the solution pair $(\mu^*, u^*)$ into a proper network structure using several techniques from graph theory. With these networks at hand, a practitioner is then able to investigate topologies associated with this novel representation of routing optimization solutions.

## Graph preliminary extraction

In this section, we expand on the *graph pre-extraction* step: extracting a network representation from the numerical solution output of the *DMK-Solver*. This involves a combination of numerical methods for discretizing the space and translating the values of $\mu^*$, and $u^*$ into edge weights of an auxiliary network, which we denote as $G=(V,E,W)$, where $V$ is the set of nodes, $E$ the set of edges and $W$ the set of weights.

The DMK solver outputs the solution on a triangulation of the domain $\Omega$ (here also named *grid*) and denoted as $\Delta_\Omega = \{T_i\}_i$, with $\cup T_i = \Omega$. The numerical solution, piecewise constant on each triangle $T_i$, is considered assigned to the triangle barycenter (center of gravity) at position $\mathbf{b}_i = (x_i, y_i) \in \Omega$. Note that in this work we focus on a $2D$ space, but the procedure can be generalized to $3D$. This means that the result is a set of pairs $\{(\mu^*(\mathbf{b}_i), u^*(\mathbf{b}_i))\}_i$. We can track any function of these two quantities. For simplicity, we use $\mu^*$ (see Fig. 1 for various examples), but one could use $u^*$ or a function of these two. This choice does not affect the procedure, although the resulting network might be different.

We neglect information on the triangles where the solution is smaller than a user-specified threshold $\delta \in \mathbb{R}_{\geq 0}$, in order to work only with the most relevant information. Formally, we only keep the information on $T_i$ such that $\mu^*(\mathbf{b}_i) \geq \delta$. We observed empirically that in many cases, several triangles contain a value of $\mu^*$ that is orders of magnitude smaller than others, see for instance the scale of Fig. 1. Since we want to build a network that connects these barycenters, we remark that this procedure depends on the choice of the threshold $\delta$: if $\delta_1 < \delta_2$, then $G(\delta_2) \subset G(\delta_1)$. On one hand, the smaller $\delta$, the more likely $G$ is to be connected, but at the cost of containing many possibly loop-forming edges and nodes (the extreme case $\delta = 0$ uses the whole grid to build the final network); on the other hand, the higher $\delta$, the smaller the final network is (both in terms of the number of nodes and edges). Thus one needs to tune the parameter $\delta$ such that resulting paths from sources to sinks are connected while avoiding the inclusion of redundant information.

The set of relevant triangles does not correspond to a straightforward meaningful network structure, i.e. a set of nodes and edges connecting neighboring nodes. In fact, we want to remove as much as possible the biases introduced by the underlying triangulation and thus we start by connecting the triangle barycenters. For this, we need rules for defining nodes, edges and weights on the edges. Here, we propose three methods for defining the graph nodes and edges and two functions to assign the weights. The overall *graph pre-extraction* routine

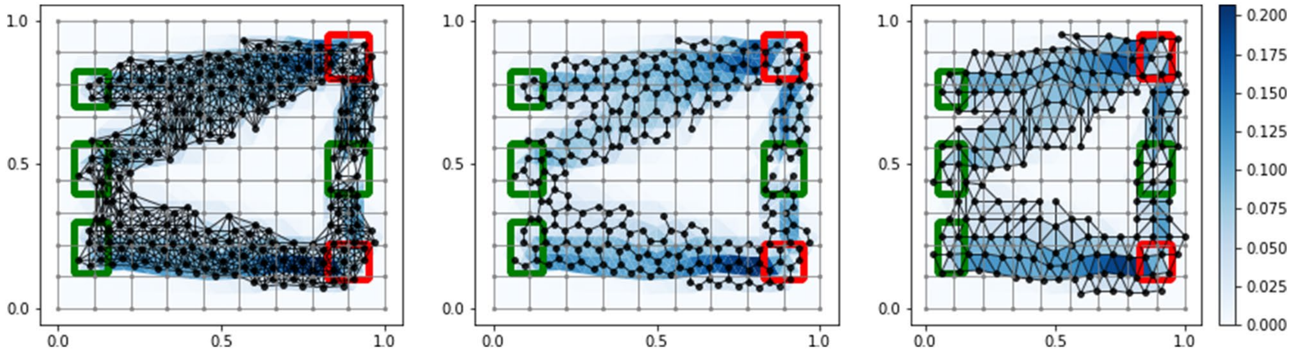


**Figure 2.** *Graph pre-extraction* rules. Left: edge-or-node sharing (I); center: edge-only sharing (II); right: original triangulation (III). We monitor the conductivity $\mu$ and use parameters $\mu_0 = 1,\ \beta = 1.02,\ \delta = 0.0001$. Weights $w_{ij}$ are chosen with AVG (i), *f* is chosen such that sources and sinks are inside green and red rectangles respectively.

is given by choosing one of the former and one of the latter, and it can be applied also to more general inputs beyond solutions of the *DMK-Solver*.

**Rules for selecting nodes and edges.** Selecting $V$ and $E$ requires defining the neighborhood $\sigma(T_i)$ of a triangle in the original triangulation $\Delta_\Omega$ (for $i$ such that $\mu^*(\mathbf{b}_i) \geq \delta$). We consider three different procedures:

(I) Edge-or-node sharing: $\sigma(T_i)$ is the set of triangles that either share a grid edge or a grid node with $T_i$.
(II) Edge-only sharing: $\sigma(T_i)$ is the set of triangles that share a grid edge with $T_i$. Note that $|\sigma(T_i)| \leq 3,\ \forall i$.
(III) Original triangulation: let $v,\ w,\ s$ be the grid nodes of $T_i$; then add $v, w, s$ to $V$ and $(v,\ w),\ (w,\ s),\ (s,\ v)$ to $E$. Note that in this case we make direct use of the graph associated to the triangulation and consider $\sigma(T_i)$ as in rule (II).

It is worth mentioning that since the grid $\Delta_\Omega$ is non-uniform and $\mu^*$ is not constant, we cannot control *a priori* the degree $d_i$ of a node $i$ in the graph $G$ generated for a particular threshold $\delta$. We give examples of networks resulting from these three definitions in Fig. 2 and a pseudo-code for them in Algorithm 2.

**Algorithm 2** Graph extraction

```
Input: $(\mu, u)$ solution of the DMK-Solver, $\delta$ threshold, $\{T_i\}_i$ grid triangulation
Output: a network $G(V,E,W)$
Initialize: $V, E = \emptyset$
for $i$ s.t. $\mu(\mathbf{b}_i) \geq \delta$ do
  if rule = (I) or rule = (II)
  $V \leftarrow V \cup \{i\}$
  for $T_j \in \sigma(T_i)$ s.t. $\mu(\mathbf{b}_j) \geq \delta$ do
    $V \leftarrow V \cup \{j\}$
    $E \leftarrow E \cup \{e_{ij}\} := \{(i,j)\}$
    $w_{ij} := w(\mu(\mathbf{b}_i), \mu(\mathbf{b}_j))$
  end for
  else if rule = (III)
  $V \leftarrow V \cup \{i_1, i_2, i_3\}$, where $V(T_i) = \{i_1, i_2, i_3\}$
  $E \leftarrow E \cup \{(i_1,i_2),(i_1,i_3),(i_2,i_3)\}$
  $w_{i_1 i_2} := w(\mu(\mathbf{b}_i), \mu(\mathbf{b}_{j_1})), w_{i_1 i_3} := w(\mu(\mathbf{b}_i), \mu(\mathbf{b}_{j_2})), w_{i_2 i_3} := w(\mu(\mathbf{b}_i), \mu(\mathbf{b}_{j_3}))$, where $j_1, j_2, j_3 \in \sigma(T_i)$
end for
```

**Rules for selecting weights.** The weights $w_{ij}$ are assigned to edges $e_{ij} := (i,j) \in E$ by the function $w(\mu(\mathbf{b}_i), \mu(\mathbf{b}_j))$, considering the density defined on the original triangles. We consider two possibilities for this function:

(i) Average (AVG): $w_{ij} = \frac{\mu(\mathbf{b}_i)+\mu(\mathbf{b}_j)}{2}$.
(ii) Effective reweighing (ER): $w_{ij} = \frac{\mu(\mathbf{b}_i)}{d_i} + \frac{\mu(\mathbf{b}_j)}{d_j}$.

While using the average as in (i) captures the intuition, it may overestimate the contribution of a triangle when this has more than one neighbor in $G$ with the risk of calculating a total density larger than the original output of the *DMK-Solver*. To avoid this issue, we consider an *effective* reweighing as in (ii), where each triangle contribution by the degree $d_i = |\sigma_i|$ of a node $i \in V$ is reweighted, with $\sigma_i$ the set of neighbors of $i$. This guarantees the recovery of the density obtained from *DMK-Solver*, since $\frac{1}{2}\sum_{i,j} w_{ij} = \frac{1}{2}\sum_i \left[\mu(\mathbf{b}_i) + \sum_{j\in\sigma_i} \frac{\mu(\mathbf{b}_j)}{d_j}\right] = \sum_i \mu(\mathbf{b}_i)$, where in the sum we neglected isolated nodes, i.e. $i$ s.t. $d_i = 0$. Note that in the case of choosing the original triangulation for node and edge selection (case (III) above), the ER rule does not apply; in that case, we use AVG, i.e. given an edge $e$, its weight is the average between its two neighboring triangles.

## Graph filtering

The output of the graph extraction step is a network closer to our expectation of obtaining an optimal network topology resulting from a routing optimization problem. However, this network may contain redundant structures like dangling nodes or small irrelevant loops (see Fig. 2). These are not related to any intrinsic property of optimality, but rather are a feature of the discretization procedure resulting from the graph pre-extraction step. It is thus important to filter the network by removing these redundant parts. However, how to perform this removal in an automated and principled way is not an obvious task. One has to be careful in removing enough structure, while not compromising the core optimality properties of the network. This removal is then a problem in-and-of-itself, we name it *graph filtering* step. We now proceed by explaining how we tackle it in a principled way and discuss its quantitative interpretation in terms of minimizing a cost function interpolating between an operating and an infrastructural cost.

**The *Discrete DMK-Solver*.** Going beyond heuristics and inspired by the problem presented in "The routing optimization problem" section, we consider as a solution for the graph filtering step, the implementation of a second routing optimization algorithm to the network $G$ output of the pre-extraction step, i.e. in discrete space. Several choices for this could be drawn, for instance, from routing optimization literature[51], but we need to make sure that this second optimization step does not modify any of the intrinsic properties related to optimality resulting from the *DMK-Solver*. We thus propose to use a *discrete* version of the *DMK-Solver* (*discrete-DMK-Solver*). This was proven to be related to the *Basis Pursuit* (BP) optimization problem[52]. In fact, BP is related[53] to the PP dynamical problem in discrete space and the *discrete-DMK-Solver* gives a solution to the PP in discrete space[52]. The discretization results in a reduction of the computational costs for solutions of BP problems, compared to standard combinatorial optimization approaches[52]. Being an adaptation to discrete settings of our original optimization problem, it is a natural candidate for a graph filtering step, preserving the solution's properties.

The problem is stated as follows. Consider the *signed incidence* matrix $\mathbf{B} \in \mathbb{M}_{N\times M}$ of a weighted graph $G = (V, E, W)$, with entries $B_{ie} = \pm 1$ if the edge $e$ has node $i$ as start/end point, 0 otherwise; $N = |V|$ and $M = |E|$. Denote $\ell = \{\ell_e\}_e$ the vector of edge lengths, $\mathbf{f}$ a $N$-dim vector of source-sink values with entries satisfying $\sum_{i\in V} f_i = 0$; this is the discrete analogues of the source-sink function $\mathbf{f}(x)$ introduced in Section "The routing optimization problem"; the functions $\mu(t) \in \mathbb{R}^M$ and $u(t) \in \mathbb{R}^N$ correspond to the *conductivity* and *potential* respectively, similarly to the continuous case, but this time they are vectors with entries $\mu_e(t)$ and $u_i(t)$ defined on edges and nodes respectively. The PP discrete dynamics corresponding to the original routing optimization problem can be written as:

$$f_i = \sum_e B_{ie} \frac{\mu_e(t)}{\ell_e} \sum_j B_{ej}\, u_j(t)\,, \tag{5}$$

$$\mu'_e(t) = \left[\frac{\mu_e(t)}{\ell_e} |\sum_j B_{ej}\, u_j(t)|\right]^{\beta_d} - \mu_e(t)\,, \tag{6}$$

$$\mu_e(0) > 0\,, \tag{7}$$

where $|\cdot|$ is the absolute value element-wise. Equation (5) corresponds to Kirchoff's law, Eq. (6) is the discrete dynamics with $\beta_d$ a parameter controlling for different routing optimization mechanisms (analogously to $\beta$ in Eq. 2); Eq. (7) is the initial condition. The importance of this system stems in having an interesting theoretical correspondence: its equilibrium point corresponds to the minimizer of a cost function analogous to Eq. (4) that, similarly to the continuous case, can be interpreted as global energy functional. This is:

$$\mathscr{L}_\beta(\mu(t)) = \frac{1}{2}\sum_e \mu_e(t) \left(\frac{1}{\ell_e}\sum_j B_{ej}\, u_j(\mu(t))\right)^2 \ell_e + \frac{1}{2}\sum_e \frac{\mu_e(t)^{P(\beta)}}{P(\beta)}\, \ell_e\,, \tag{8}$$

where $P(\beta) = (2-\beta)/\beta$ and $u(\mu(t))$ is a function implicitly defined as the solution of Eq. (5). The first term corresponds to the energy dissipated during transport, it can be interpreted as the operating costs, whereas the second is the infrastructural cost. The equilibrium point of $\mu_e(t)$ is stationary at the previous energy function, and for $\beta_d = 1$ it acts also as the global minimizer due to its convexity. For $\beta_d > 1$ the energy is not convex, thus in general the functional will present several local minima towards which the dynamics will be attracted. The case $\beta_d < 1$ does not act as a filter because it encourages trajectories to spread through the network, instead of removing edges, and so not interesting to our purposes. Discretization in time of Eq. (6) by the implicit Euler

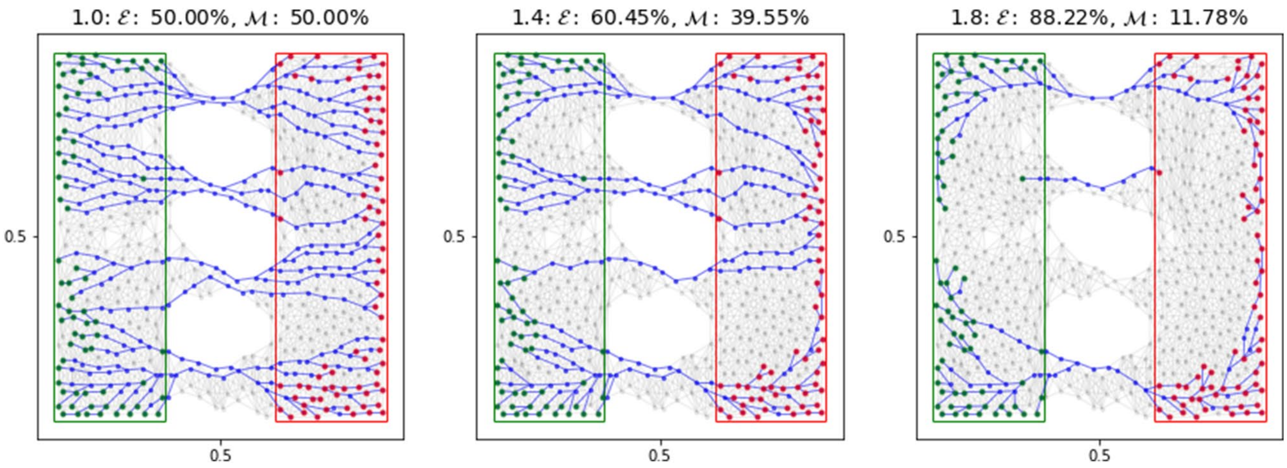


**Figure 3.** *Graph filtering* rules. Left: $\beta_d = 1.0$; center: $\beta_d = 1.4$; right: $\beta_d = 1.8$. The numbers on top denote the percentage contributions of operating and infrastructural cost to the energy as in Eq. (8). Green and red dots represent sources and sinks respectively ($\tau_{BC} = 10^{-1}$); blue edges are those $e$ with $\mu_e^* \geq \delta_d = 10^{-3}$. The filtering input is generated from *DMK-Solver* with $\beta = 1.05$. The apparent lack of symmetry of the network's branches is due to numerical discretization of the domain, solver and threshold $\delta$. As the relative size of the terminal set decreases compared to the size of remaining part of the domain, this lack of symmetry becomes negligible.

scheme combined with Newton method leads to an efficient numerical solver, see Facca et al.[52] for more details. The above scheme gives the solution to the BP problem and represents the *discrete-DMK-Solver*. Similarly to the graph pre-extraction step, the filtering is also valid beyond networks related to solutions of the *DMK-Solver*. It applies to more general inputs if defined on a discrete space, for instance, images. Finally, notice that the filter generates a graph with a new set of nodes and edges, both subsets of the corresponding ones in $G$, result of the pre-extraction. The weights of the final graph can then be assigned with same rules as in "Rules for selecting nodes and edges"; in addition, one can consider as weights the values of $\mu_e^*$ resulting from the BP problem (we named this weighing method "BPW"). Alternatively, one can ignore the weights of BP and keep (for the edges remaining after the filter) the weights as in the previous pre-extraction step (labeled as "IBP"). Analogously to what done on the original triangulation, we discard the edges $e$ for which $\mu_e < \delta_d$. In our experiments we use as initial density distribution $\mu_e(0) = w(e), \forall e \in E$, where $w$ correspond to the weight of the edge $e$ in the pre-extracted graph. Figure 3 shows an example of three filtering settings on the same input.

### Selecting sources and sinks.

The *discrete-DMK-Solver* requires in input a set of source and sink nodes ($S^+$ and $S^-$) that identify the support of the forcing vector $\mathbf{f}$ introduced in "The Discrete DMK-Solver". However, the graph pre-extraction output $G$ might contain redundant nodes (or edges) as mentioned before. In principle, among the nodes $i \in V$, all of those contained in the support of $f(x = \mathbf{b}_i)$, i.e. contained in the supports of sources and sinks of the original routing optimization problem in Eq. (1), are *eligible* to be treated as sources or sink in the resulting network. However, several paths connecting source and sink nodes may be redundant and clearly not compatible with an optimal routing network (see Supplementary Fig. S2 for such an example). Therefore, it is important to select "representatives" for sources and sinks, such that the final network is heuristically closer to optimality. Here we propose a criterion to select source and sink nodes from the eligible ones in each of the connected components $\{C_m\}_m$ of $G$, using a combination of two network properties. Starting from the complete graph formed by all the nodes characterized by a significant (above the threshold) density, source and sink nodes and rates are defined as follows. A node $i \in S^+$, i.e. is a source $f_i > 0$, if either i) is in the convex hull of the set of eligible sources or ii) its betweenness centrality is smaller than a given threshold $\tau_{BC}$. Similarly for sink nodes in $S^-$. This is because, on one side, nodes in the convex hull capture the outer shape structure of the source and sink sets defined in the continuous problem; on the other side, nodes with *small* values of the betweenness centrality capture the end-points of $G$ inside the source and sink sets, analogously to leaves (i.e., degree-one nodes). Note that, due to the high graph connectivity, degree centrality is not appropriate for selecting these ending parts. We present these ideas in more detail in the Supplementary Fig. S2. Once we have identified the sets of source and sink vertices, we need to assign a proper value $f_i$ such that Kirchhoff law is satisfied in each of the different connected components $C_m$. It is reasonable to assume that each connected component is "closed", i.e. $\sum_{i \in C_m} f_i = 0, \forall C_m$. Denoting with $|S|$ the number of elements in a set $S$ and $V(C_m)$ the set of nodes in $C_m$, we then distribute the mass-fluxes uniformly by setting $f_i = \frac{1}{|S^+ \cap V(C_m)|}$ for $i \in S^+$, and $f_i = -\frac{1}{|S^- \cap V(C_m)|}$ for $i \in S^-$ sinks ($f_i = 0$ otherwise) so that the total original source and sink flux is assigned to the overall source/sink nodes of all $C_m$. Note that this procedure maintains the overall system and each connected component "closed", as stated above.

### Computational complexity.

The numerical implementation of our graph extraction algorithm is based on finite element-like solvers that transform the problem into a finite sequence of linear systems. This implies that we need to run a variable number $N_T$ of iterations in time, each requiring $N_N$ Newton steps. Every Newton step requires the approximate solution of a linear system of dimension $N$ by pre-conditioner conjugate gradient solver, which has complexity $O(N \log N)$[54]. The time complexity of our graph extraction algorithm is then $O(N_T \times N_N \times N \log N)$. In practice, because of exponential convergence of the time discretization towards equilibrium[52], $N_T$ is typically constant approximately $< 10^2$, instead $N_N \sim 5$. In the worst cases $N_T\, N_N \sim N^{0.3}$. Thus the total complexity is $O(N \log N)$.

The time complexity of other related approaches such as the ORC-based algorithms is dominated by the computation of the Wasserstein distance, which typically takes $O(M\, k \times k^3 \log k)$, where $k = 2M/N$ is the average network degree, when using linear programming and can be further improved using wavelet earth-mover-distance approximation approaches[55]. While $M > N$, in sparse networks such as those used in our experiments, $M \sim N$.

Other approaches that solve similar problems are based on Steiner tree solvers[39] and have a complexity which depends on the number of sources and sinks, in addition to the system size. Instead, our method complexity does not depend on them, but only on the network size.

## Model validation

Our extraction pipeline proceeds by compressing routing information in the raw output of the *DMK-Solver* (although what follows is not restricted to this case) on a lean network structure. This might lead us to lose relevant information in the process. Hence, we need to devise *a posteriori* estimates that provide quantitative guidance on the “leanness” and information loss of the final network. Here we propose metrics to evaluate the compression performance of the various graph pre-extraction and filtering protocols. The raw information is made of a set of weights $w(T_i)$ representing the values $(\mu^*, u^*)$ on each of the triangles $T_i \in \Omega$. We consider as the truth benchmark the distribution of $w$, or any other quantity of interest, supported on the subgrid $\Delta_\Omega^\delta \subset \Delta_\Omega$ formed by all triangles where $w$ is larger than the threshold value $\delta$, i.e., $\Delta_\Omega^\delta := \{T_i \in \Delta_\Omega : w(T_i) \geq \delta\}$. We expect that a good compression scheme should preserve both the total *amount* of the weights from the original solution in $\Delta_\Omega^\delta$ and the information of *where* these weights are located inside the domain $\Omega$. Also, we want this compression to be parsimonious, i.e. to store the least amount of information as possible. We test against these two requirements by proposing two metrics that measure: i) an information difference between the raw output of the *DMK-Solver* and the network extracted using our procedure, capturing the information of where the weights are located in space; ii) the amount of information needed to store the network.

Our first proposed metric relies on partitioning $\Omega$ in several subsets and then calculating the difference in the extracted network weights and the uncompressed output, *locally* within each subset. More precisely, we partition $\Omega$ into $P$ non intersecting subsets $C_\alpha \subset \Omega$, with $\alpha = 1, \ldots, P$ and $\cup_1^P C_\alpha = \Omega$. For example, we define $C_\alpha = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, for $x_i, x_{i+1}, y_j$ and $y_{j+1}$, consecutive elements of $N$-regular partitions of [0, 1], and $P = (N-1)^2$. Denote with $w_\delta(T_i)$ the weight on the triangle $T_i \in \Delta_\Omega^\delta$, resulting from the *DMK-Solver* (usually a function of $\mu^*$ and $u^*$). If we denote the *local weight* of $\Delta_\Omega^\delta$ inside $C_\alpha$ as $w_\alpha = \sum_{i:\mathbf{b}_i \in C_\alpha} w_\delta(T_i)$, then we propose the following evaluation metric:

$$\hat{w}_q(G) : = \frac{1}{P}\left[\sum_{\alpha=1}^{P}\left(|\sum_{e\in E}\mathbb{I}_\alpha(e)\, w_e - w_\alpha\,|\right)^{q}\right]^{\frac{1}{q}}, \tag{9}$$

where $\mathbb{I}_\alpha(e)$ is an indicator of whether an edge $e = (i,j) \in E$ is inside an element $C_\alpha$ of the partition, i.e. $\mathbb{I}_\alpha(e) = 1, 0, 1/2$ if both $\mathbf{b}_i, \mathbf{b}_j$ are in $C_\alpha$, none of them are, or only one of them is, respectively. In words, $\hat{w}_q(G)$ is a distance between the weights of the network extracted by our procedure and the original weights, output of the *DMK-Solver*, over each of the local subsets $C_\alpha$. This metric penalizes networks that either place large-weight edges where they were not present in the original triangulation, or low-weight ones where they were instead present originally. In this work, we consider the Euclidean distance, i.e. $q = 2$, but other choices are also possible. Note that $\hat{w}_q(G)$ does not say anything about how much information was required to store the processed network. If we want to encourage parsimonious networks, i.e. networks with few redundant structures, then we should include in the evaluation the monitoring of $L(G) = \sum_{e\in E} \ell_e$, the total path length of the compressed network, where the edge length $\ell_e$ can be specified based on the application. Standard choices are uniform $\ell_e = 1$, $\forall e$ or the Euclidean distance between $\mathbf{b}_i$ and $\mathbf{b}_j$. Intuitively, networks with small values of both $\hat{w}_q(G)$ and $L(G)$ are both accurate and parsimonious representations of the original DMK solutions defined on the triangulation.

We evaluate numerous graph extraction pipelines in terms of these two metrics on various routing optimization problem settings and parameters. In Fig. 4 we show the main results for a distribution of 170 networks obtained with $\beta \in \{1.1,\ 1.2,\ 1.3\}$ and $\beta_d = 1.1$. Similar results were obtained for other parameter settings. Networks are generated as follows: first, we choose a set of 5 different initial transport densities $\mu_0$, grouped in parabola-like, delta-like and uniform distributions, and a set of 12 different configurations for sources/sinks (mainly rectangles placed in different positions along the domain, see Supplementary Information for more details). Then, for each of these setups, we run our procedure: (i) first the *DMK-Solver* calculates the solution of the continuous problems; (ii) then we apply the *graph pre-extraction* procedure according rules of “Rules for selecting nodes and edges" and weights as in “Rules for selecting weights”; iii) finally, we run the *graph filtering* step and consider various weight functions, as described in Fig. 4.

We observe that not applying the final filtering step and considering rule I with ER to build the graph (I-ER-None), the values of $\hat{w}_2(G)$ are smaller than other cases. This is expected as by filtering we remove information and thus achieve better performance with this metric when compared to no filtering. However, we pay a price in terms of total relative length as $L(G)/L_{max}$ is larger for this case. When working with rule II, we notice the appearance of many non-optimal small disconnected components, and this effect deteriorates if filtering is activated. Corresponding statistics show low values for both $\hat{w}_2(G)$ and $L(G)/L_{max}$. We argue that this is because rule

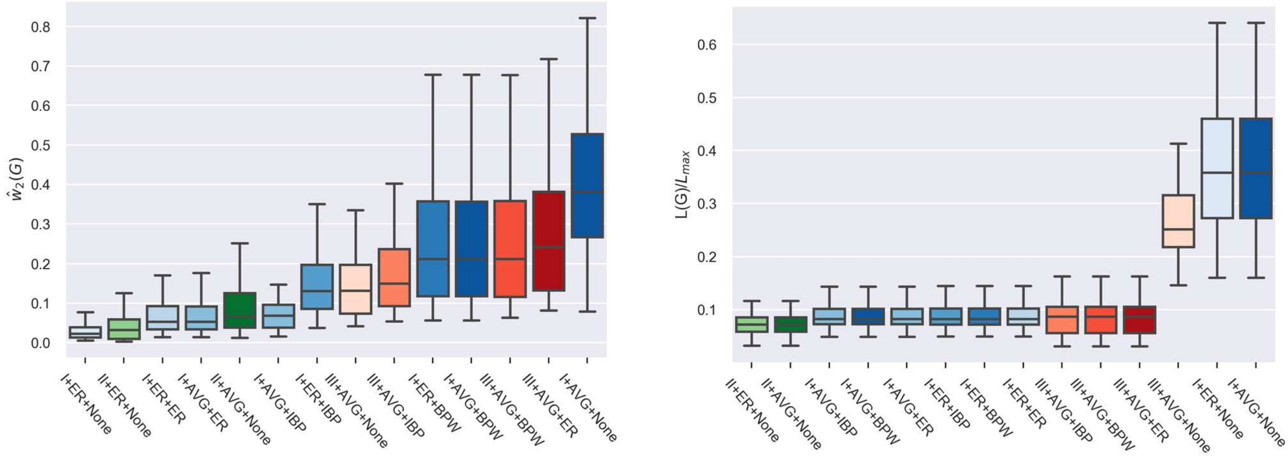


**Figure 4.** Graph extraction performance evaluation. We plot results for the different combinations of the graph extraction rule in terms of: (left) the metric $\hat{w}_q(G)$ of Eq. (9); (right) total network length $L(G)$ normalized by $L_{max}$, the max length over the 170 networks. Each bar denotes a possible combination as follows: roman numbers denote one of the three rules I–III ("Rules for selecting nodes and edges"); first label after the number denotes one of the rules to assign weights i–ii ("Rules for selecting weights"), which is applied to the output of the first step; the second (and last) label denotes the same rule but applied after the filtering step, "None" means that nothing is done, i.e. no filter applied, "IBP" means filter applied but with no reweighing, i.e. when an edge is removed by the filter we simply lose information without relocating its weight. Bars are color-coded so that rules I–III have three different primary colors and their corresponding routines have different shades of that main color. Here, we keep track of the conductivity $\mu$ and show medians and quartiles of a distribution over 170 networks generated with $\beta \in \{1.1,\ 1.2,\ 1.3\}$, $\beta_d = 1.1$ and $\delta = 0.01$.

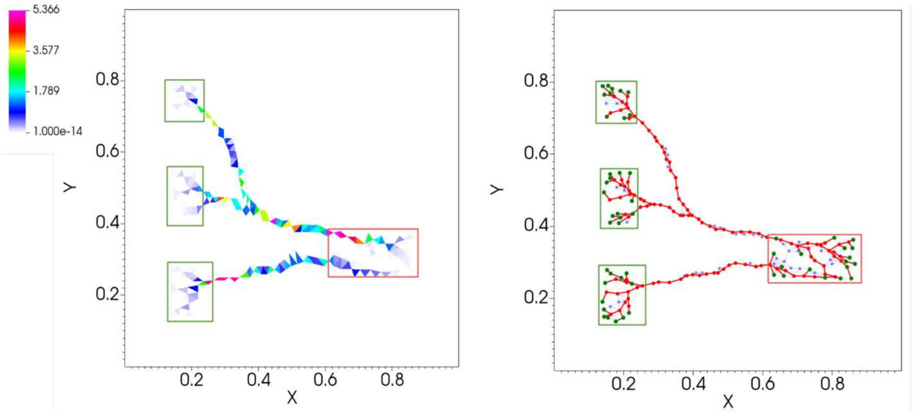


**Figure 5.** Network extraction example. We show a network generated from a routing optimization problem with parameters $\beta_c = 1.4$, $\beta_d = 1.3$ and $\delta = 0.001$; (left) raw output of the *DMK-Solver*; (right) final network extracted using the routine I-ER-ER.

II produces, by construction, fewer redundant objects than rule I in the initial phase. This might have a similar effect as a filter but is done *a priori* during the pre-extraction, because rule II produces in this phase a limited number of effective neighbors. However, this comes at a price of higher variability with the sampled networks, as the variance of $\hat{w}_2(G)$ is higher than for the other combinations. Among the possibilities with filtering applied, we observe that rule I performs better than rule III, while all the weighting rules give a similar performance in terms of both metrics. Any combination involving rule I plus filtering has a similar performance as rule II in terms of both metrics but with smaller variability. Finally, these combinations perform differently in terms of the number of disconnected components (not shown here), with rule II producing more spurious splittings, as already mentioned. Depending on the application at hand, a practitioner should select one of these combinations based on their properties as discussed in this section. We give an example of a network generated with I-ER-ER in Fig. 5.

## Application: network analysis of a vein network

We demonstrate our protocol on a biological network of fungi foraging for resources in space. The network structure corresponds to the fungi response to food cues while foraging[56]. Edges are veins or venules and connect adjacent nodes. This and those of other types of fungi are well known networks typically studied using

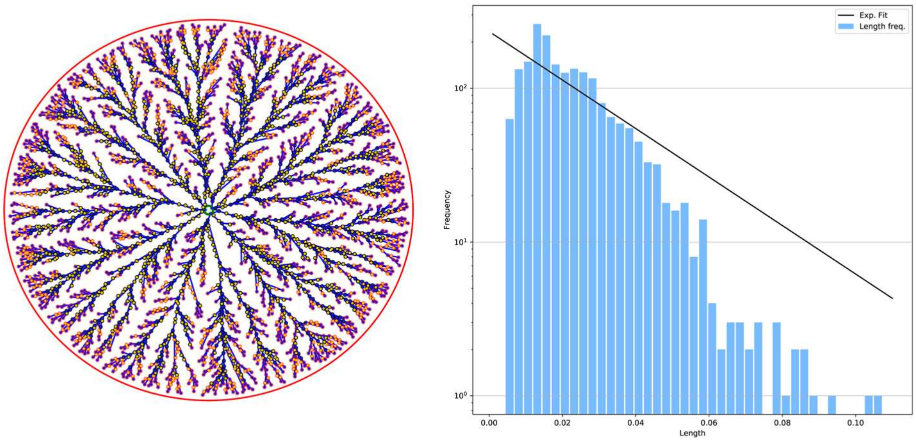


**Figure 6.** Application to fungi network. We generate a synthetic network similar to the image of Fig. 1a reported in Boddy et al.[31] and Fig. 4a in Obara et al.[29] for the of *Phanerochaete velutina* fungus[56] and Fig. 1 in their supplementary for the *Coprinus picaceus*. Fitted parameters are: $P_0 = 234.00, \gamma = 36.32$. Here $f^+(x, y) = 1$, if $(x - 0.5)^2 + (y - 0.5)^2 \leq 0.01$; $f^+(x, y) = 0$, otherwise; $f^-(x, y) = -1$, if $0.01 < (x - 0.5)^2 + (y - 0.5)^2 \leq 0.45$; $f^-(x, y) = 0$, otherwise. The network on the left corresponds to the filtered graph. Yellow nodes are degree-2 nodes that we omitted when computing the length distribution. Green and red outlines are used to denote nodes in $S^+$ and $S^-$, respectively.

image segmentation methods[28–31]. It is thus interesting to compare results found by these techniques and by our approach, under the conjecture that the underlying dynamic driving the network structure could be the same as the optimality principles guiding our extraction pipeline. In particular, we are interested in analyzing the distribution $P(\ell)$ of the vein lengths, i.e. the network edges. The benchmark $P(\ell)$ distribution obtained by Baumgarten and Hauser[28] using image processing techniques is an exponential of the type $P(\ell) = P_0\, e^{-\gamma \ell}$. Accordingly, as shown in Fig. 6, we find that an exponential fit (with values $P_0 = 234.00, \gamma = 36.32$) well captures the left part of the distribution, i.e. short edges. Differences between fit and observed data can be seen in the right-most tail of the graph, corresponding to longer path lengths, where the data decay faster than the fit. However, we find that the exponential fit is nevertheless better than other distributions, such as the gamma and log-normal proposed in Dirnberger and Mehlhorn[57] for the *P. Polycephalum*. Drawing definite quantitative conclusions is beyond the scope of our work, as this example aims at a qualitative illustration of possible applications that can be addressed with our model. In general, however, it seems not possible to choose a single distribution that well fits both center and tails of the distribution for various datasets of this type[57].

To conclude, we demonstrate the flexibility of our graph extraction method on a more general input than the one extracted from *DMK-Solver*. Specifically, we consider as example an image of *P. Polycephalum* taken from data publicly available in the Slime Mold Graph Repository (SMGR) repository[58]. We first downsample an image of the SMGR's *KIST Europe data set*, using *OpenCV* (left) and a color scale defined on the pixels as an artificial $\mu^*$ function. We build a graph using the *graph pre-extraction* and *graph filtering* steps as shown in Fig. 7. Notice that our protocol in its standard settings with filtering can only generate tree-like structures. Therefore, if we want to obtain a network with loops as we did in Fig. 7, we should consider a modification of our routine, which can be done in a fully automatized way, as explained in more details in the Supplementary S4. In short, after the *graph pre-extraction* step, where loops are still present, we extract a tree-like structure close to the original loopy graph and give this in input to the filtering. We can then add *a posteriori* edges that connect terminals that were close by in the graph obtained from the pre-extraction step but removed by the filter, thus recovering loops. In case obtaining loops is not required, our routine can be used with no modifications. Adapting our filtering model to allow for loopy structures in a principled way, analogously to what done in "Graph preliminary extraction", will be subject of future work.

## Discussion

We propose a graph extraction method for processing raw solutions of routing optimization problems in continuous space into interpretable network topologies. The goal is to provide a valuable tool to help practitioners bridging the gap between abstract mathematical principles behind optimal transport theory and more interpretable and concrete principles of network theory. While the underlying routing optimization scheme behind the first step of our routine uses recent advances of optimal transport theory, our tool enables automatic graph extraction without requiring expert knowledge. We purposely provide a flexible routine for graph extraction so that it can be easily adapted to serve the specific needs of practitioners from a wider interdisciplinary audience. We thus encourage users to choose the parameters and details of the subroutines to suitably customize the protocol based on the application of interest. To help guiding this choice, we provide several examples here and in the Supplementary Information. We anticipate that this work will find applications beyond that of automating graph extraction from routing optimization problems. We remark that two of the three steps of our protocol apply to inputs that might not necessarily come from solutions of routing optimization. Indeed, the pipeline can be

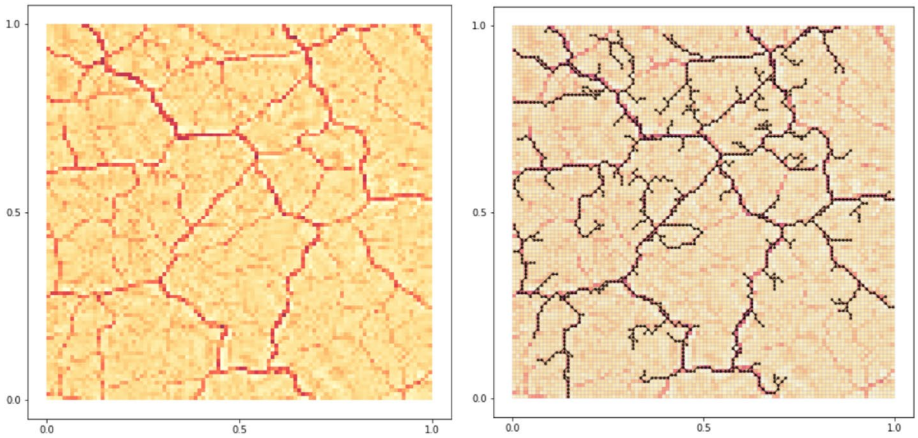


**Figure 7.** Application to images. We take an image of the *P. Polycephalum* from the SMGR repository[58].The picture used is a 1200x1200-pixel section of an original image of size 5184x3456 pixels (see Supplementary S4 for details) and extract a network with step 2 and 3 of our protocol. As a pre-processing step, we downsample the image using *OpenCV* (left) and use the color scale defined on the pixels as an artificial $\mu^*$ function. Using this information and the grid structure associated to the image's pixels, we first build a graph $G$ with the *graph pre-extraction* step described in "Graph preliminary extraction"; then, we obtain a graph $G_f$ (right) using the *graph filtering* step of "Graph filtering", for an appropriate selection of sources and sinks, and adding a correction to retrieve loops. Notice that our protocol in its standard settings with filtering can only generate tree-like structures. Therefore, if we want to obtain a network with loops, we should consider a minor modification of our routine, which can be done in a fully automatized way, as explained in more detail in the Supplementary S4.

applied to any image setting where an underlying network needs to be extracted. This can have relevant impact in applications involving biological systems like neuronal networks, for which we observe an increasing amount of data from imaging experiments. The advantage of our setting with respect to more conventional machine learning methods is that the final structure extracted with our approach minimizes a clearly defined energy functional, that can be interpreted as the combination of the total dissipated energy during transport and the cost of building the transport infrastructure. We foresee that this minimizing interpretation together with the simplification of the pipeline from abstract modeling to final concrete network outputs will foster cross-breeding between fields as our tool will inform network science with optimal transport principles and vice-versa. In addition, we expect to advance the field of network science by promoting the creation of new network databases related to routing optimization problems. For instance, an interesting direction for future work is to extend our optimal transport-based method to address other network-related applications such as geometry-based community detection.

## Code availability

open source codes and executables are available at https://github.com/Danielaleite/Nextrout.

Received: 4 May 2020; Accepted: 5 November 2020
Published online: 30 November 2020

## References

1. Banavar, J. R., Colaiori, F., Flammini, A., Maritan, A. & Rinaldo, A. Topology of the fittest transportation network. *Phys. Rev. Lett.* **84**, 4745 (2000).
2. Corson, F. Fluctuations and redundancy in optimal transport networks. *Phys. Rev. Lett.* **104**, 048703 (2010).
3. Li, G. *et al.* Towards design principles for optimal transport networks. *Phys. Rev. Lett.* **104**, 018701 (2010).
4. Yeung, C. H., Saad, D. & Wong, K. M. From the physics of interacting polymers to optimizing routes on the london underground. *Proc. Nat. Acad. Sci.* **110**, 13717–13722 (2013).
5. Guimerà, R., Díaz-Guilera, A., Vega-Redondo, F., Cabrales, A. & Arenas, A. Optimal network topologies for local search with congestion. *Phys. Rev. Lett.* **89**, 248701 (2002).
6. Donetti, L., Hurtado, P. I. & Munoz, M. A. Entangled networks, synchronization, and optimal network topology. *Phys. Rev. Lett.* **95**, 188701 (2005).
7. Ronellenfitsch, H., Dunkel, J. & Wilczek, M. Optimal noise-canceling networks. *Phys. Rev. Lett.* **121**, 208301 (2018).
8. Gazit, Y., Berk, D. A., Leunig, M., Baxter, L. T. & Jain, R. K. Scale-invariant behavior and vascular network formation in normal and tumor tissue. *Phys. Rev. Lett.* **75**, 2428 (1995).
9. Garlaschelli, D., Caldarelli, G. & Pietronero, L. Universal scaling relations in food webs. *Nature* **423**, 165 (2003).
10. Balister, P. *et al.* River landscapes and optimal channel networks. *Proc. Nat. Acad. Sci.* **115**, 6548–6553 (2018).
11. Santambrogio, F. Optimal channel networks, landscape function and branched transport. *Interfaces Free Bound.* **9**, 149–169 (2007).
12. Katifori, E., Szöllősi, G. J. & Magnasco, M. O. Damage and fluctuations induce loops in optimal transport networks. *Phys. Rev. Lett.* **104**, 048704 (2010).
13. Santambrogio, F. Optimal transport for applied mathematicians. *Birkäuser, NY* **55**, 58–63 (2015).
14. Messinger, S. M., Mott, K. A. & Peak, D. Task-performing dynamics in irregular, biomimetic networks. *Complexity* **12**, 14–21 (2007).
15. Fruttiger, M. Development of the mouse retinal vasculature: angiogenesis versus vasculogenesis. *Investig. Ophthalmol. Vis. Sci.* **43**, 522–527 (2002).

16. Schaffer, C. B. *et al.* Two-photon imaging of cortical surface microvessels reveals a robust redistribution in blood flow after vascular occlusion. *PLoS Biol.* **4**, e22 (2006).
17. Altarelli, F., Braunstein, A., Dall'Asta, L., De Bacco, C. & Franz, S. The edge-disjoint path problem on random graphs by message-passing. *PloS one***10**, e0145222 (2015).
18. Bayati, M. *et al.* Statistical mechanics of steiner trees. *Phys. Rev. Lett.* **101**, 037208 (2008).
19. De Bacco, C., Franz, S., Saad, D. & Yeung, C. H. Shortest node-disjoint paths on random graphs. *J. Stat. Mech: Theory Exp.* **2014**, P07009 (2014).
20. Braunstein, A. & Muntoni, A. P. The cavity approach for steiner trees packing problems. *J. Stat. Mech: Theory Exp.* **2018**, 123401 (2018).
21. Ronellenfitsch, H. & Katifori, E. Global optimization, local adaptation, and the role of growth in distribution networks. *Phys. Rev. Lett.* **117**, H364-5 (2016).
22. Brancolini, A. & Solimini, S. Fractal regularity results on optimal irrigation patterns. *Journal de Mathématiques Pures et Appliquées* **102**, 854–890 (2014).
23. Xia, Q. Motivations, ideas and applications of ramified optimal transportation. *ESAIM Math. Model. Numer. Anal.* **49**, 1791–1832 (2015).
24. Pegon, P., Santambrogio, F. & Xia, Q. A fractal shape optimization problem in branched transport. *Journal de Mathématiques Pures et Appliquées* **123**, 244–269 (2019).
25. Facca, E., Cardin, F. & Putti, M. Towards a stationary Monge–Kantorovich dynamics: the physarum polycephalum experience. *SIAM J. Appl. Math.* **78**, 651–676 (2018).
26. Facca, E., Daneri, S., Cardin, F. & Putti, M. Numerical solution of Monge–Kantorovich equations via a dynamic formulation. *J. Sci. Comput.* **82**, 1–26 (2020).
27. Facca, E. & Cardin, F. Branching structures emerging from a continuous optimal transport model. *J. Comput. Phys.***Submitted** (2020).
28. Baumgarten, W. & Hauser, M. Detection, extraction, and analysis of the vein network. *J. Comput. Interdis. Sci.* **1**, 241–249 (2010).
29. Obara, B., Grau, V. & Fricker, M. D. A bioimage informatics approach to automatically extract complex fungal networks. *Bioinformatics* **28**, 2374–2381 (2012).
30. Bebber, D. P., Hynes, J., Darrah, P. R., Boddy, L. & Fricker, M. D. Biological solutions to transport network design. *Proc. R. Soc. B Biol. Sci.* **274**, 2307–2315 (2007).
31. Boddy, L., Wood, J., Redman, E., Hynes, J. & Fricker, M. D. Fungal network responses to grazing. *Fungal Genet. Biol.* **47**, 522–530 (2010).
32. Dehkordi, M. T., Sadri, S. & Doosthoseini, A. A review of coronary vessel segmentation algorithms. *J. Med. Signals Sens.* **1**, 49 (2011).
33. Bradski, G. & Kaehler, A. *Learning OpenCV: Computer vision with the OpenCV library* (OReilly Media Inc, New York, 2008).
34. Dirnberger, M., Kehl, T. & Neumann, A. Nefi: Network extraction from images. *Sci. Rep.* **5**, 15669 (2015).
35. Chai, D., Forstner, W. & Lafarge, F. Recovering line-networks in images by junction-point processes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition***1894–1901** (2013).
36. Wang, J., Song, J., Chen, M. & Yang, Z. Road network extraction: a neural-dynamic framework based on deep learning and a finite state machine. *Int. J. Remote Sens.* **36**, 3144–3169 (2015).
37. Wegner, J. D., Montoya-Zegarra, J. A. & Schindler, K. Road networks as collections of minimum cost paths. *ISPRS J. Photogr. Remote Sens.* **108**, 128–137 (2015).
38. Hwang, F. K. & Richards, D. S. Steiner tree problems. *Networks* **22**, 55–89 (1992).
39. Liu, L., Song, Y., Zhang, H., Ma, H. & Vasilakos, A. V. Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks. *IEEE Trans. Comput.* **64**, 818–831 (2013).
40. Hagberg, A., Swart, P. & S Chult, D. Exploring network structure, dynamics, and function using networkx. Tech. Rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008).
41. Xu, K., Tang, C., Tang, R., Ali, G. & Zhu, J. A comparative study of six software packages for complex network research. In *2010 Second International Conference on Communication Software and Networks*, 350–354 (IEEE, 2010).
42. Batagelj, V. & Mrvar, A. Pajek-program for large network analysis. *Connections* **21**, 47–57 (1998).
43. Bastian, M., Heymann, S. & Jacomy, M. Gephi: an open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media* (2009).
44. Yin, C. *et al.* Network science characteristics of brain-derived neuronal cultures deciphered from quantitative phase imaging data. *Sci. Rep.* **10**, 1–13 (2020).
45. Sia, J., Jonckheere, E. & Bogdan, P. Ollivier-ricci curvature-based method to community detection in complex networks. *Sci. Rep.* **9**, 1–12 (2019).
46. Ni, C.-C., Lin, Y.-Y., Luo, F. & Gao, J. Community detection on networks with ricci flow. *Sci. Rep.* **9**, 1–12 (2019).
47. Xue, Y. & Bogdan, P. Reliable multi-fractal characterization of weighted complex networks: algorithms and implications. *Sci. Rep.* **7**, 1–22 (2017).
48. Evans, L. C. & Gangbo, W. *Differential equations methods for the Monge–Kantorovich mass transfer problem* Vol. 653 (American Mathematical Soc, New York, 1999).
49. Xia, Q. On landscape functions associated with transport paths. *Discrete Contin. Dyn. Syst* **34**, 1683–1700 (2014).
50. Tero, A., Kobayashi, R. & Nakagaki, T. A mathematical model for adaptive transport network in path finding by true slime mold. *J. Theor. Biol.* **244**, 553–564 (2007).
51. Oliveira, C. A. & Pardalos, P. M. *Mathematical aspects of network routing optimization* (Springer, Berlin, 2011).
52. Facca, E., Cardin, F. & Putti, M. Physarum dynamics and optimal transport for basis pursuit. arXiv preprint arXiv:1812.11782 (2018).
53. Straszak, D. & Vishnoi, N. K. Irls and slime mold: Equivalence and convergence. arXiv preprint arXiv:1601.02712 (2016).
54. Chan, R. H. & Ng, M. K. Conjugate gradient methods for toeplitz systems. *SIAM Rev.* **38**, 427–482 (1996).
55. Li, W., Ryu, E. K., Osher, S., Yin, W. & Gangbo, W. A parallel method for earth movers distance. *J. Sci. Comput.* **75**, 182–197 (2018).
56. Fricker, M., Boddy, L. & Bebber, D. Network organisation of mycelial fungi. In *Biology of the fungal cell*, 309–330 (Springer, 2007).
57. Dirnberger, M. & Mehlhorn, K. Characterizing networks formed by p. polycephalum. *J. Phys. D Appl. Phys.* **50**, 224002 (2017).
58. Dirnberger, M., Mehlhorn, K. & Mehlhorn, T. Introducing the slime mold graph repository. *J. Phys. D Appl. Phys.* **50**, 264001 (2017).

## Acknowledgements

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Diego Baptista and Daniela Leite.

## Author contributions

All authors contributed to developing the models, analyzing the results and reviewing the manuscript. E.F, M.P. and C.D.B. conceived the experiment(s), D.B. and D.L. conducted the experiments.

## Funding

Open Access funding enabled and organized by Projekt DEAL.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** is available for this paper at https://doi.org/10.1038/s41598-020-77064-4.

**Correspondence** and requests for materials should be addressed to C.D.B.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# Convergence Properties of Optimal Transport-Based Temporal Networks

Diego Baptista( ) and Caterina De Bacco

Max Planck Institute for Intelligent Systems, Cyber Valley,
Tuebingen 72076, Germany
diego.theuerkauf@tuebingen.mpg.de

**Abstract.** We study network properties of networks evolving in time based on optimal transport principles. These evolve from a structure covering uniformly a continuous space towards an optimal design in terms of optimal transport theory. At convergence, the networks should optimize the way resources are transported through it. As the network structure shapes in time towards optimality, its topological properties also change with it. The question is how do these change as we reach optimality. We study the behavior of various network properties on a number of network sequences evolving towards optimal design and find that the transport cost function converges earlier than network properties and that these monotonically decrease. This suggests a mechanism for designing optimal networks by compressing dense structures. We find a similar behavior in networks extracted from real images of the networks designed by the body shape of a slime mold evolving in time.

**Keywords:** Optimal transport theory · Graph theory · Network structure · Network design

## 1 Introduction

Optimal Transport (OT) theory studies optimal ways of transporting resources in space [1,2]. The solutions are optimal paths that connect sources to sinks (or origins to destinations) and the amount of flow traveling through them. In a general setting one may start from a continuous space in 2D, arbitrarily set sources and sinks, and then look for such optimal paths without any predefined underlying topology. Empirically, in many settings, these paths resemble network-like structures that embed optimality, in that traffic flowing along them is minimizing a transport cost function. Among the various ways to compute these solutions [3], a promising and computationally efficient recent approach is that of Facca et al. [4–6], which is based on solving a set of equations (the so-called *Dynamical Monge-Kantorovich* (DMK) equations). This starts from an initial guess of the optimal paths that are then updated in time until reaching a steady state configuration. At each time step, one can automatically extract a principled network from a network-like structure using the algorithm proposed

R. M. Benito et al. (Eds.): COMPLEX NETWORKS 2021, SCI 1072, pp. 578–592, 2022.
https://doi.org/10.1007/978-3-030-93409-5_48

in Baptista et al. [7]. This in turn allows observing a sequence of network structures that evolves in time towards optimality, as the dynamical equations are iterated. While we know that the transport cost function is decreasing along this trajectory, we do not know how network properties on these structures evolve. For instance, in terms of the total number of edges or nodes, one may intuitively expect a monotonically decreasing behavior, from a topology covering uniformly the whole space, towards a compressed one only covering a subset of it efficiently. Analyzing the properties of networks that provide optimal transport efficiency is relevant in many contexts and has been explored in several works [8–11]. However, these studies usually consider pre-existing underlying topologies that need to be optimized. Moreover, they focus on network properties at convergence. Here instead we consider the situation where a network can be designed in a continuous 2D space, i.e. with no pre-defined underlying topology, and monitor the whole evolution of network properties, in particular away from convergence. While this question has been explored in certain biological networks [12–15], a systematic investigation of this intuition is still missing. In this work, we address this problem by considering several optimization settings, extracting their optimal networks, and then measuring core network properties on them. We find that network sequences show similar convergence patterns of those exhibited by their continuous counterparts. However, topological features of optimal networks tend to develop slightly slower than total cost function minimization. We also find that, in some cases, this delay in convergence presented by the networks might give better representations than those extracted at other cost-based convergence times. Finally, we analyze real data of the *P. polycephalum* slime mold evolving its network-like body shape in time as it explores the space foraging. We use networks extracted from images generated in wet-lab experiments [16], and analyze their topological features. Pattern matches can be found between synthetic graphs and this family of real networks.

Understanding how network topology evolves towards optimality may shed light on broader questions about optimal network properties and how to obtain them.

## 2 The Model

*The Dynamical-Monge Kantorovich Set of Equations.* We now present the main ideas of how to extract sequences of networks that converge towards an optimal configuration, according to optimal transport theory. We start by introducing the dynamical system of equations regulating this, as proposed by Facca et al. [4–6]. We assume that the problem is set on a continuous 2-dimensional space $\Omega \in \mathbb{R}^2$, i.e. there is no pre-defined underlying network structure. Instead, one can explore the whole space to design an optimal network topology, determined by a set of nodes and edges, and the amount of flow passing through each edge. Sources and sinks of a certain mass (e.g. passengers in a transportation network, water in a water distribution network) are displaced on it. We denote these with a "forcing" function $f(x) = f^+(x) - f^-(x) \in \mathbb{R}$, describing the flow generating sources $f^+(x)$

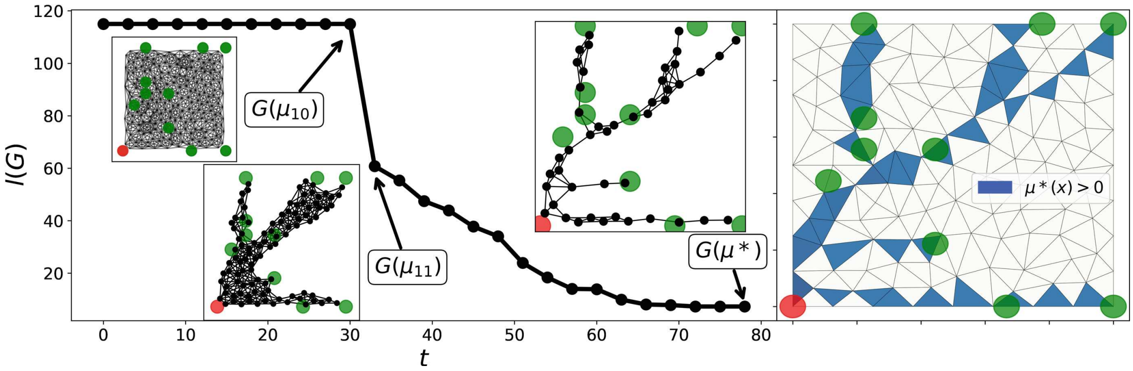


**Fig. 1. Temporal networks.** On the left, the total length $l(G)$ (i.e. sum of the edge lengths), as a function of time $t$; the networks inside the insets correspond to different time steps. On the right, optimal transport density $\mu^*$; triangles are a $[0,1]^2$ discretization. In both plots, red and green circles correspond to the support of $f^+$ and $f^-$, i.e. sources and sinks, respectively. This sequence is obtained for $\beta = 1.5$.

and sinks $f^-(x)$ (also known as *source* and *target* distributions, respectively). It is assumed that $\int_\Omega f(x)dx = 0$ to ensure mass balance. We suppose that the flow is governed by a transient Fick-Poiseuille type flux $q = -\mu\nabla u$, where $\mu, u$ and $q$ are called *conductivity* (or *transport density*), *transport potential* and *flux*, respectively. Intuitively, the conductivity can be seen as proportional to the size of the edges where mass can flow, the potential could be seen as pressure on nodes, thus determining the flux passing on it.

The set of *Dynamical Monge-Kantorovich* (DMK) equations is given by:

$$-\nabla \cdot (\mu(t,x)\nabla u(t,x)) = f^+(x) - f^-(x)\,, \tag{1}$$

$$\frac{\partial \mu(t,x)}{\partial t} = [\mu(t,x)\nabla u(t,x)]^\beta - \mu(t,x)\,, \tag{2}$$

$$\mu(0,x) = \mu_0(x) > 0\,, \tag{3}$$

where $\nabla = \nabla_x$. Equation (1) states the spatial balance of the Fick-Poiseuille flux and is complemented by no-flow Neumann boundary conditions; Eq. (2) enforces the dynamics of this system and Eq. (3) is the initial configuration, this can be thought of as an initial guess of the solution. The parameter $\beta$ (traffic rate) tunes between various optimization setting: for $\beta < 1$ we have congested transportation where traffic is minimized, $\beta > 1$ is branched transportation where traffic is encouraged to consolidate along fewer edges, and $\beta = 1$ is shortest path-like. In this work we only consider the branched transportation regime $1 < \beta < 2$, as this is the only one where meaningful network structures can be extracted [7].

Solutions $(\mu^*, u^*)$ of Eqs. (1)–(3) minimize the transportantio n cost function $\mathcal{L}(\mu, u)$ [4–6], defined as:

$$\mathcal{L}(\mu,u) := \mathcal{E}(\mu,u) + \mathcal{M}(\mu,u) \tag{4}$$

$$\mathcal{E}(\mu,u) := \frac{1}{2}\int_\Omega \mu|\nabla u|^2 dx, \quad \mathcal{M}(\mu,u) := \frac{1}{2}\int_\Omega \frac{\mu^{\frac{(2-\beta)}{\beta}}}{2-\beta}dx \quad . \tag{5}$$

$\mathcal{L}$ can be thought of as a combination of $\mathcal{M}$, the total energy dissipated during the transport (or network operating cost) and $\mathcal{E}$, the cost to build the network infrastructure (or infrastructural cost).

### 2.1 Network Sequences

The conductivity $\mu$ at convergence regulates where the mass should travel for optimal transportation. This is a function of a 2-dimensional space, it can be turned into a principled network $G(\mu)$ (a set of nodes, edges, and weights on them) by using the method proposed by [7], which in turn determines the design of the optimal network. While the authors of that work considered only values at convergence, this method is still valid at any time step, in particular at time steps before convergence. This then leads to a sequence of networks evolving in time as the DMK equations are iterated. Figure 1 shows three networks built using this method at different time steps. The leftmost inset is the densest representation that one can build from the shown discretization of the space (a triangulation), as in the plot on the right side: all the nodes are connected to all of their closest neighbors. This is what happens at initial time steps where the network is built from mass uniformly displaced in space, as per uniform initial condition. On the other hand, the rightmost network is built from a $\mu$ at convergence, consolidated on a more branched structure.

Formally, let $\mu(x,t)$ be a *transport density* (or *conductivity*) function of both time and space obtained as a solution of the DMK model. We denote it as the sequence $\{\mu_t\}_{t=0}^{T}$, for some index $T$ (usually taken to be that of the convergent state). Every $\mu_t$ is the $t$-th update of our initial guess $\mu_0$, computed by following the rules described in Eqs. (1)–(3). This determines a sequence of networks $\{G(\mu_t)\}_{t=0}^{T}$ extracted from $\{\mu_t\}_{t=0}^{T}$ with [7]. Figure 1 shows three different snapshots of one of the studied sequences.

*Convergence Criteria.* Numerical convergence of the DMK equations (1)–(3) can be arbitrarily defined. Typically, this is done by fixing a threshold $\tau$, and stopping the algorithm when the cost does not change more than that between successive time steps. However, when this threshold is too small ($\tau = 10^{-12}$ in our experiments), the cost or the network structure may consolidate to a constant value way in advance, compared to the algorithmic one. Thus, to meaningfully establish when is network optimality reached, we consider as convergence time the first time step when the transport cost, or a given network property, reaches a value that is smaller or equal to a certain fraction $p$ of the value reached by the same quantity at algorithmic convergence (in the experiments here we use $p = 1.05$). We refer to $t_{\mathcal{L}}$ and $t_P$ for the convergence in times in terms cost function or a network property, respectively.

*Network Properties.* We analyze the following main network properties for the different networks in the sequences and for different sequences. Denote with $G$ one of the studied networks belonging to some sequence $\{G(\mu_t)\}_{t=0}^{T}$. We study the following properties relevant to the design of networks for optimal transport of resources through it.

- $|N|$, total number of nodes;
- $|E|$, total number of edges;
- *total length* $l(G) = \sum_e l(e)$, i.e. the sum of the lengths of every edge. Here $l(e)$ is the Euclidean distance between the nodes endpoints of $e$;
- Average degree, the mean number of neighbors per node;
- $bif(G)$, the number of bifurcations; a *bifurcation* is a node with degree greater than 2;
- $leav(G)$, the number of leaves; a leave is a node with degree equal to 1.

## 3 Results on Synthetic Data

To study the behavior of network structures towards optimality, we perform an extensive empirical analysis as follows. We generate synthetic data considering a set of optimal transport problems, determined by the configuration of sources and sinks. In fact, the final solutions strongly depend on how these are displaced in space. We consider here a scenario where we have one source and many sinks, which is a relevant situation in many applications. For instance, in biology, this would be the case for a slime mold placed on a point in space (the source) and looking for multiple sources of food (the sinks). Formally, consider a set of points $S = \{s_0, s_1, ..., s_M\}$ in the space $\Omega = [0,1]^2$, and $0 < r$ a positive number. We define the distributions $f^+$ and $f^-$ as

$$f^+(x) \propto \mathbb{1}_{R_0}(x), \quad f^-(x) \propto \sum_{i>0} \mathbb{1}_{R_i}(x)$$

where $\mathbb{1}_{R_i}(x) := 1$, if $x \in R_i$, and $\mathbb{1}_{R_i}(x) := 0$, otherwise; $R_i$ is the circle of center $s_i$ and radius $r$ (the value of $r$ is automatically selected by the solver based on the discretization of the space); and the proportionality is such that $f^+$ and $f^-$ are both probability distributions. The transportation cost is that of Eq. (4).

*Data Generation.* We generate 100 transportation problems by fixing the location of the source $s_0 = (0,0)$ (i.e. the support of $f^+$ at $(0,0)$), and sampling 15 points $s_1, s_2, ..., s_M$ uniformly at random from a regular grid (see supplementary information). By choosing points from vertices of a grid, we ensure that the different sinks are sufficiently far from each other, so they are not redundant. We start from an uniform, and thus non-informative, initial guess for the solution, $\mu_0(x) = 1, \forall x$. We fix the maximum number of iterations to be 300. We say that the sequence $\{\mu_t\}_{t=0}^T$ *converges* to a certain function $\mu^*$ at iteration $T$ if either $|\mu_T - \mu_{T-1}| < \tau$, for a *tolerance* $\tau \in (0,1]$, or $T = 300$. For the experiments reported in this manuscript, the tolerance $\tau$ is set to be $10^{-12}$. We consider different values of $\beta \in [1.1, 1.9]$, thus exploring various cost functions within the branched transportation regime. Decreasing $\beta$ from 2 to 1 results in traffic being more penalized, or consolidation of paths into fewer highways less encouraged. In total, we obtain 900 network sequences, each of them containing between 50 and 80 networks.

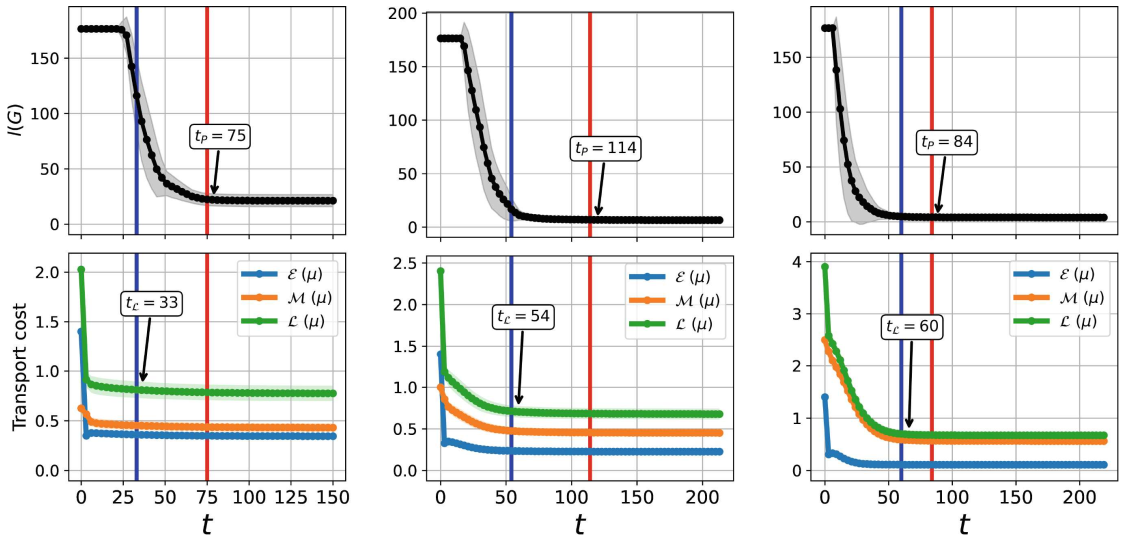


**Fig. 2. Total length and Lyapunov cost.** Mean (markers) and standard deviations (shades around the markers) of the total length (top plots) and of the Lyapunov cost, energy dissipation $\mathcal{E}$ and structural cost $\mathcal{M}$ (bottom plots), as functions of time $t$. Means and standard deviations are computed on the set described in Sect. 3. From left to right: $\beta = 1.2, 1.5$ and $1.8$. Red and blue lines denote $t_P$ and $t_\mathcal{L}$.

*Convergence: Transport Cost vs Network Properties.* Figure 2 shows a comparison between network properties and the cost function minimized by the dynamics.

We observe that $t_P > t_\mathcal{L}$ in all the cases, i.e. convergence in the cost function is reached earlier than convergence of the topological property. Similar behaviors are seen for other values of $\beta \in [1.1, 1.9]$ and for other network properties (see supplementary information). For smaller values of $\beta$ convergence in transport cost is reached faster, when the individual network properties are still significantly different from their value at convergence, see Fig. 3 for an example in terms of total path length at $\beta = 1.2$. In this case, while the cost function does not change much after $t_\mathcal{L}$, the network properties do instead. This may be because the solutions for $\beta$ close to 1 have non-zero $\mu$ on many edges but most of them have low values. Indeed, we find that the most important edges, measured by the magnitude of $\mu$ on them, are those corresponding to the topology found at a later time, when the network properties also converge, as shown in Fig. 3 (bottom). This indicates that the dynamics first considers many edges, and distributes the fluxes optimally along fewer main roads. At the end, close to convergence, it focuses instead on removing redundant edges, those that have little flux traveling.

Finally, we notice how $t_\mathcal{L}$ is smaller for $\beta$ close to 1. This reflects the fact that in this case it is easier to find a solution to the optimization problem, as for increasing $\beta$ the configuration space gets roughed with many local optima [4–6].

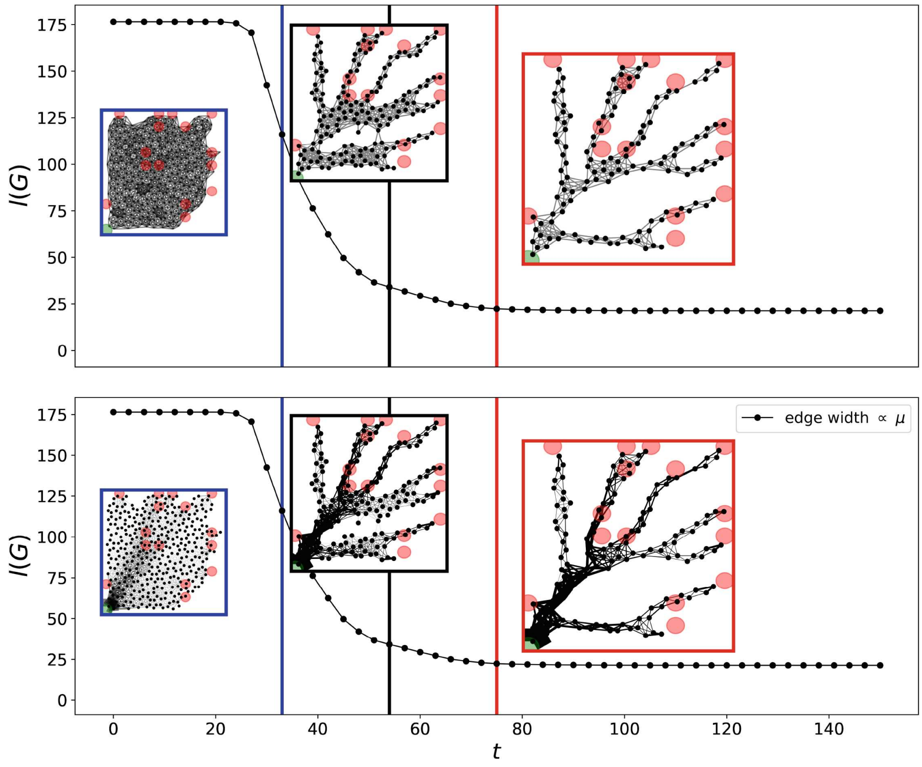


**Fig. 3. Network topologies for different convergence criteria.** Mean (markers) and standard deviations (shades around the markers) of total length $l(G)$ as a function of time. The red, black and blue vertical lines (and networks in the insets) correspond to $t_P$, the average of $t_P$ and $t_{\mathcal{L}}$, and $t_{\mathcal{L}}$, respectively. Networks without (top row) and with (bottom) edge weights proportional to their $\mu$ are plotted at those times three time steps. Hence, the networks on top highlight the topological structure, while those on the bottom the flux passing through edges.

*Convergence Behavior of Network Properties.* Figure 4 shows how the various network properties change depending on the traffic rate. The plots show their mean values computed across times, for a fixed value of $\beta$. Notice that quantities like the total length, the average degree, the number of bifurcations, the number of edges and the number of nodes decrease in time, signaling that sequences reach steady minimum states. These are reached at different times, depending on $\beta$, with convergence reached faster for lower $\beta$. Moreover, mean values of these properties converge to decreasing values, as $\beta$ increases. This is explained by a cost function increasingly encouraging consolidations of paths on fewer edges. Finally, the magnitude of the gap between the different mean values of each property for different $\beta$ depends on the individual property. For instance, the average degree changes more noticeably between two consecutive values of $\beta$ than the total path length, which shows a big gap between the value at $\beta = 1.1$ and all of the subsequent $\beta > 1.1$, that have instead similar value of this property. This also shows that certain properties better reveal the distinction between different optimal traffic regimes. The number of leaves behaves more distinctly. In fact, it exhibits two different patterns: either it remains constantly equal

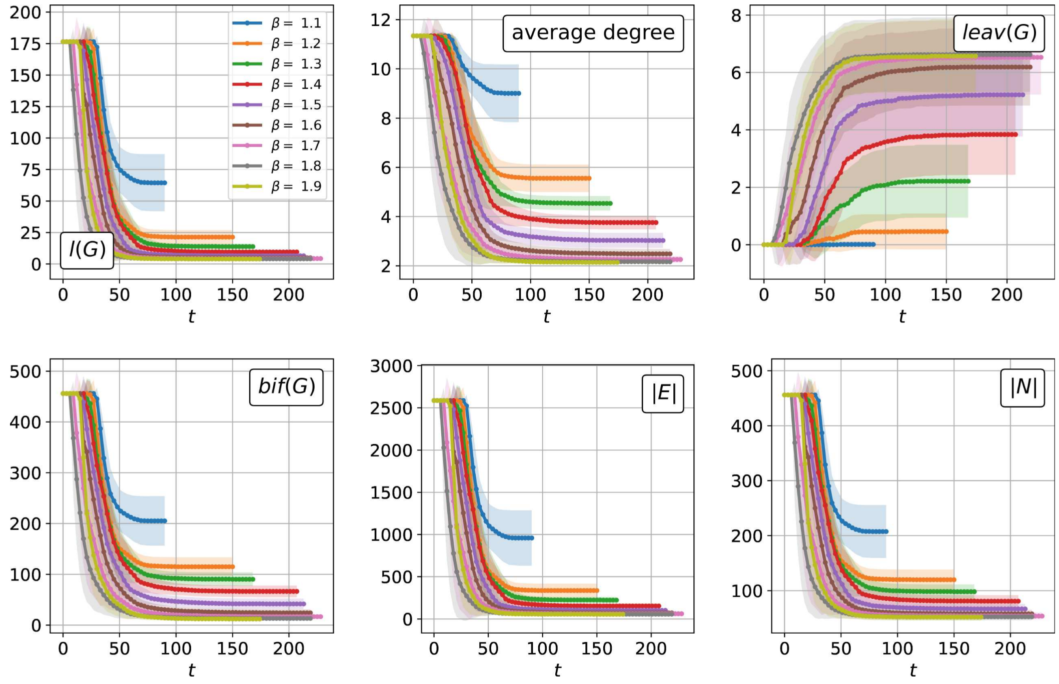


**Fig. 4. Evolution of network properties.** Mean (markers) and standard deviations (shades around the markers) of total length $l(G)$ (upper left), average degree (upper center), number of leaves $leav(G)$ (upper right), number of bifurcations $bif(G)$ (lower left), number of edges $|E|$ (lower center) and number of nodes $|N|$ (lower right), computed for different values of $\beta$ as a function of time. Notice that $|E|$ keeps changing for $t > 70$ but the scale makes it hard to perceive.

to 0 ($\beta = 1.1$) or it increases, and with different rates, as time gets larger ($\beta > 1.1$). This number increases with $\beta$, as in this regime paths consolidate into fewer edges, thus leaving more opportunities for leaves. To help intuition of the different optimal designs for various $\beta$, we plot the extracted networks at convergence in Fig. 5. The positions of source and sinks are the same in all cases. The network obtained for higher $\beta = 1.8$ contains fewer edges and nodes than the others cases. On average, these networks have $bif(G) \approx 13$ , $leav(G) \approx 7$, $l(G) \approx 4$ and an average degree $\approx 2$. These reveal various topological features on the converged networks of this traffic regime, that make it more distinct than others. For instance, having approximately 7 leaves implies that the dynamics builds networks with as many leaves as approximately half the number of sinks ($M = 15$) in this transportation problem, while on the other hand, we can see that $bif(G) \approx M$, i.e., the number of bifurcations is almost as large as the number of sinks.

### 3.1 Results on Real Networks of *P. polycephalum*

In this section, we compare the properties of the sequences $\{G(\mu_t)\}_{t=0}^{T}$ to those extracted from real images of the slime mold *P. polycephalum*. This organism has been shown to perform an optimization strategy similar to that modeled by the DMK equations of Sect. 2, while foraging for food in a 2D surface [17–19].

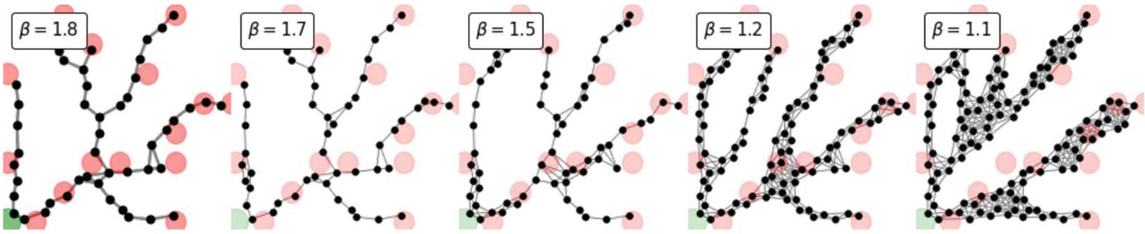


**Fig. 5. Example of optimal networks for various cost functions.** Networks extracted from the solutions of the same transportation problem but various $\beta$. Green and red and circles denote source and sinks.

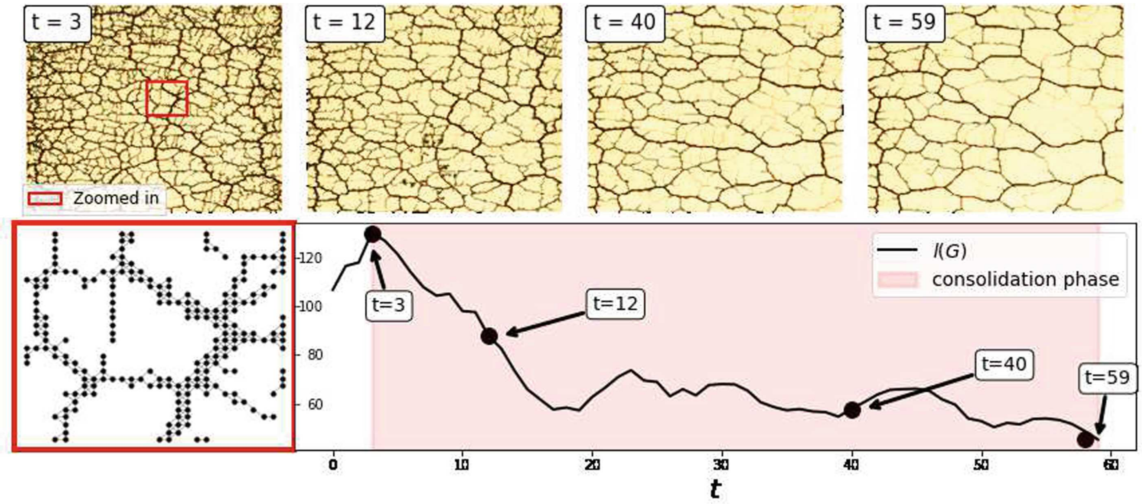


**Fig. 6. Network evolution of *P. polycephalum*.** On top: *P. polycephalum* images and networks extracted from them. Bottom left: a zoomed-in part of the graph shown inside the red rectangle on top. Bottom right: total length as a function of time. The red shade highlights a tentative consolidation phase towards optimality.

We extract these networks from images using the method proposed by [20]. This pipeline takes as input a network-like image and uses the color intensities of its different pixels to build a graph, by connecting adjacent meaningful nodes. We choose 4 image sequences from the Slime Mold Graph Repository [16]. Every sequence is obtained by taking pictures of a *P. polycephalum* placed in a rectangular Petri dish and following its evolution in time. Images are taken every 120 s from a fixed position.

We study the evolution of the total length for every sequence. We show in Fig. 6 the total length of the temporal network extracted from one of the mentioned image sequences (namely, image set *motion12*; see supplementary information), together with different network snapshots. As we can see from the lower rightmost plot, the evolution of the total length of the extracted networks resembles that of the synthetic network sequences analyzed above. This suggests that the DMK-generated sequences resemble the behavior of this real system in this time frame. This could mean that the DMK dynamics realistically represents a consolidation phase towards optimality of real slime molds [16]. Similar results are obtained for other sequences (see supplementary information).

## 4 Conclusions

We studied the properties of sequences of networks converging to optimal structures. Our results show that network sequences obtained from the solution of diverse transportation problems often minimize network properties at slower rates compared to the transport cost function. This suggests an interesting behavior of the DMK dynamics: first, it focuses on distributing paths into main roads while keeping many network edges. Then, once the main roads are chosen, it removes redundant ones where the traffic would be low. Measuring convergence of network properties would then reveal a more compressed network cleaned from redundant paths. The insights obtained in this work may further improve our understanding of the mechanisms governing the design of optimal transport networks.

We studied here a particular set of transportation problems, one source, and multiple sinks. In this case, all the main network properties studied here show similar decaying behavior. However this analysis can be replicated for more complex settings, like multiple sources and multiple sinks [21] or in multilayer networks [22], as in urban transportation networks. Potentially, this may unveil different patterns of the evolution of the topological properties than those studied in this work.

Results on real networks suggest that the networks generated by the DMK dynamics (inspired by the *P. polycephalum*) resemble realistic features. Strongly monotonic phases are not only typical of the mentioned slime molds but also a pattern in the artificially generated data. Alternative realistic behaviors may be seen by considering a modified version of the model described in Eq. (1) by adding non-stationary forcing terms. This may highlight a behavior different than the one observed in a consolidation phase, where a network converges to an optimal design and then does not change further. This is an interesting direction for future work.

**Acknowledgements.** The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Diego Baptista.

## Supplementary Information (SI)

## 1 Synthetic Data

*Details of the Studied Transport Problems.* As mentioned in the main manuscript, we consider a set of points $S = \{s_0, s_1, ..., s_M\}$ in the space $\Omega = [0,1]^2$, and $0 < r$ a positive number, and we use this to define the distributions $f^+$ and $f^-$ as

$$f^+(x) \propto \mathbb{1}_{R_0}(x), \quad f^-(x) \propto \sum_{i>0} \mathbb{1}_{R_i}(x)$$

where $R_i$ is the circle of center $s_i$ and radius $r$. The points $s_1, ..., s_M$, the support of the sink, are sampled uniformly at random from a regular grid. The used grid and different realizations of the sampling are shown in Fig. 7.

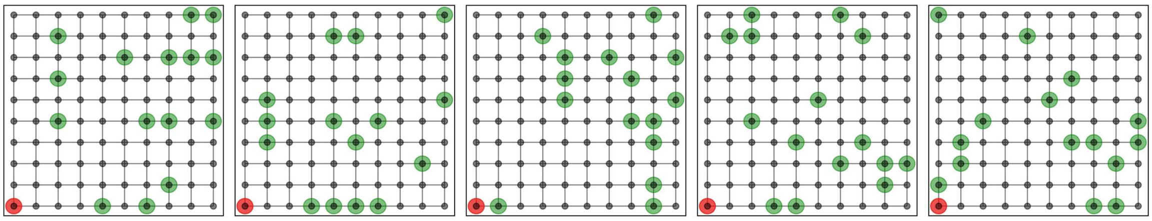


**Fig. 7. Support of $f^-$.** The nodes of the grid constitutes the set of candidates from which the support of $f^-$

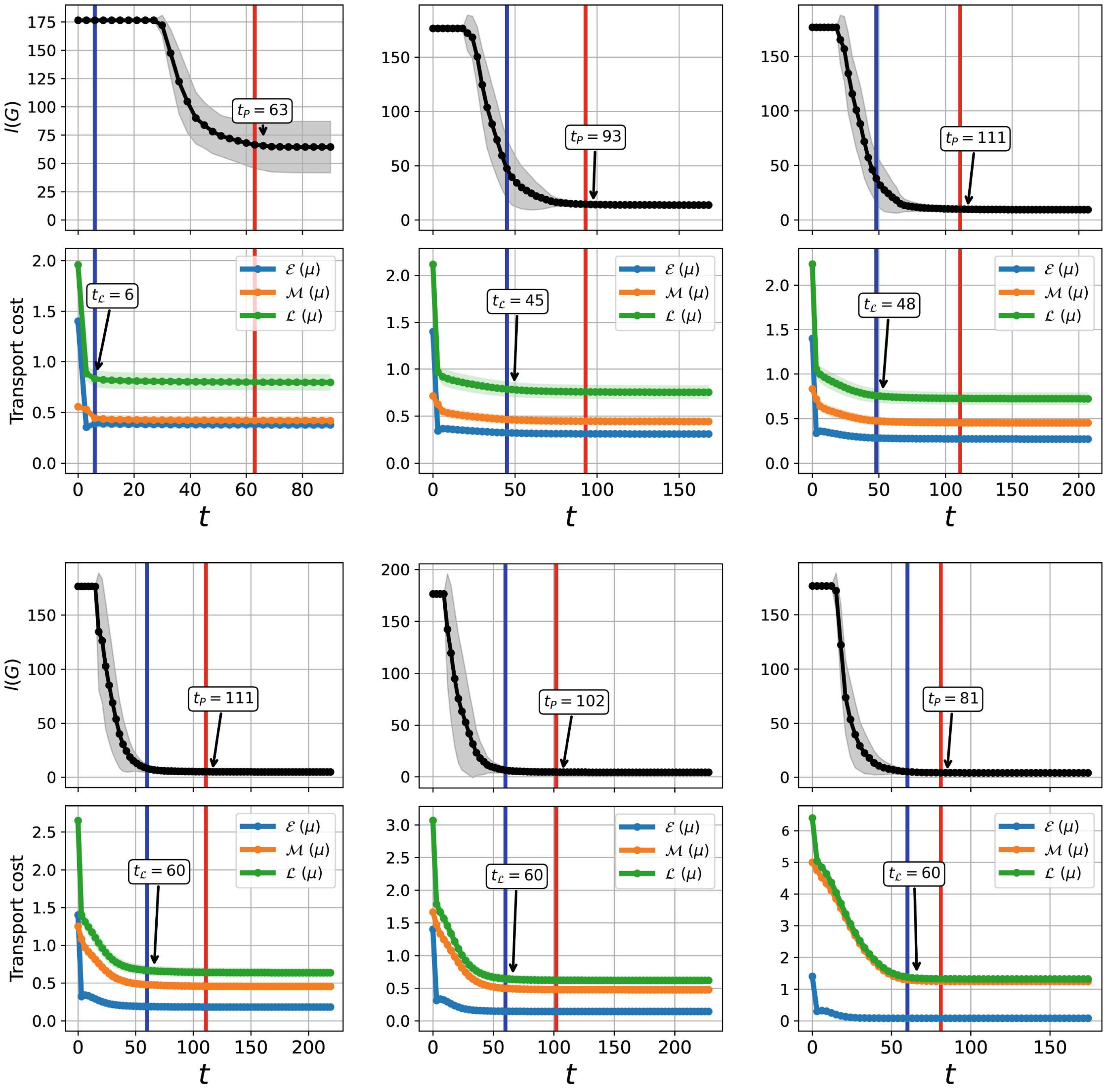


**Fig. 8. Total length and Lyapunov cost.** Top row: from left to right we see $\beta = 1.1, 1.3$ and 1.4. Bottom row: from left to right we see $\beta = 1.6, 1.7$ and 1.9. Mean and standard deviation of the total length $l(G)$ as function of time $t$; Bottom plot: Mean and standard deviation of the Lyapunov cost $\mathcal{L}$, energy dissipation $\mathcal{E}$ and structural cost $\mathcal{M}$ of transport densities. Red and blue lines denote $t_P$ and $t_\mathcal{L}$ for $p = 1.05$.

*Total Length and Lyapunov Cost.* We show in this section a figure like the one presented in the Fig. 2 of the main manuscript, for other values of $\beta$. As mentioned in there, the properties show decreasing behaviors for which is always true that $t_P > t_\mathcal{L}$ (see Fig. 8).

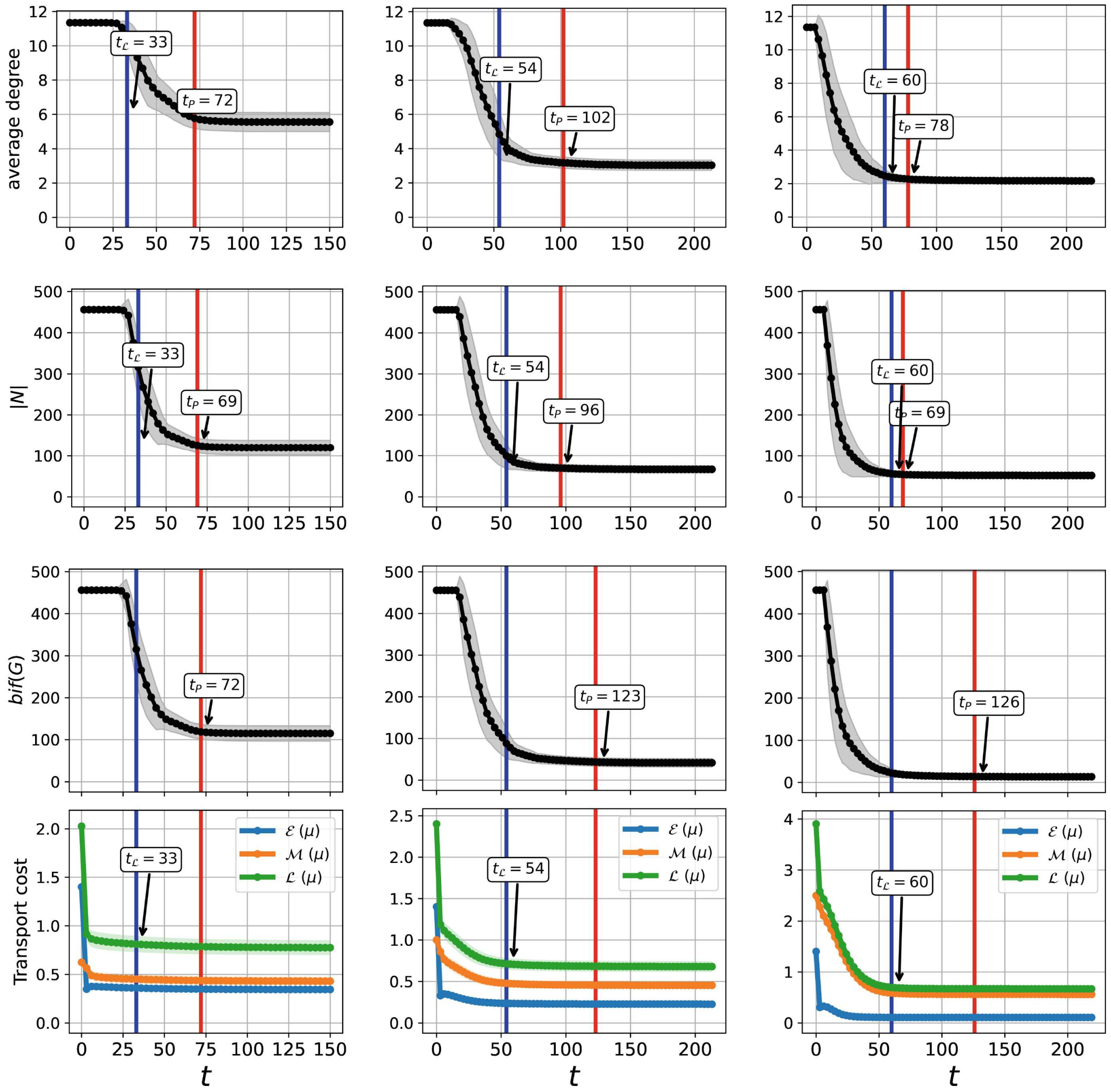


**Fig. 9. Other network properties and Lyapunov cost.** From left to right: $\beta = 1.2, 1.5$ and $1.8$. From top to bottom: Mean and standard deviation of the average degree, number of nodes $|N|$, number of bifurcations $bif(G)$, and the Lyapunov cost $\mathcal{L}$, energy dissipation $\mathcal{E}$ and structural cost $\mathcal{M}$. Red and blue lines denote $t_P$ and $t_{\mathcal{L}}$ for $p = 1.05$.

*Network Properties and Lyapunov Cost.* We show in this section a figure like the one presented in the Fig. 2 of the main manuscript, for the other network properties (see Fig. 9).

## 2 *P. polycephalum* Networks

*Data Information.* In this section, we give further details about the used real data. As mentioned in the main manuscript, the images are taken from the Slime Mold Graph Repository [16]. The number of studied sequences $\{G_i\}_i^T$ equals 4. Every sequence's length $T$ changes depending on the amount of images provided in the repository, since different experiments need more o less shots. An

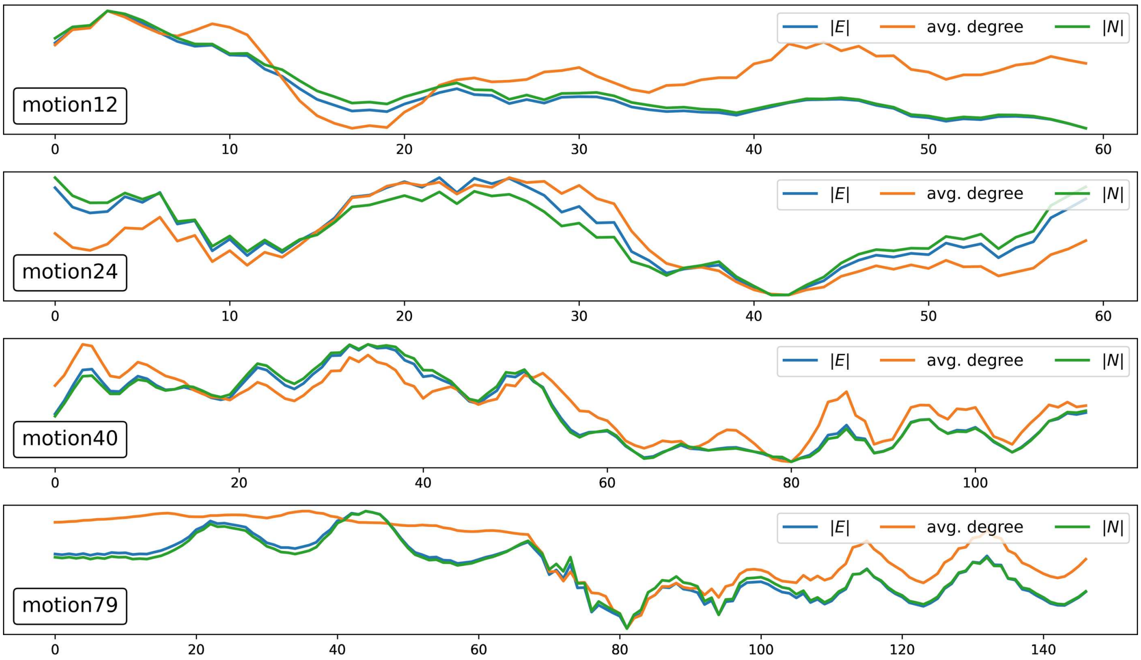


**Fig. 10. Network properties for *P. polycephalum* sequences.** From top to bottom: *motion12, motion24, motion40* and *motion79*. Subfigures show the evolution of the properties $|E|$, average degree and $|N|$ for every sequence as a function of time.

experiment, as explained in the repository's documentation, consists of placing a slime mold inside a Petri dish with a thin sheet of agar and no sources of food. The idea, as explained by the creators, is to let the slime mold fully explore the Petri dish. Since the slime mold is initially lined up along one of the short side of the dish, the authors stop capturing images once the plasmodium is about to reach the other short side.

*Network Extraction.* The studied network sequences are extracted from the image sets *motion12, motion24, motion40* and *motion79*, which are stored in the repository. Each image set contains a number of images ranging from 60 to 150, thus, obtained sequences exhibit diverse lengths. Every network is extracted using the *Img2net* algorithm described in [20]. The main parameters of this algorithms are `N_runs`, `t2`, `t3` and `new_size`. `N_runs` controls how many times the algorithm needs to be run; `t2` (and `t3`) are the minimum value (and maximum) a pixel's grayscale value must be so it is considered as a node; `new_size` is the size to which the input image must be downsampled before extracting the network from it. For all the experiments reported in this manuscript, the previously mentioned parameters are set to be `N_runs` $= 1$, `t2` $= 0.25$, `t3` $= 1$ and `new_size` $= 180$.

*More Network Properties.* Other network properties are computed for the real systems referenced in this manuscript. Similar decreasing behaviors, like the one shown for the total length property in the main manuscript, are found for these properties; see Figs. 10 and 11.

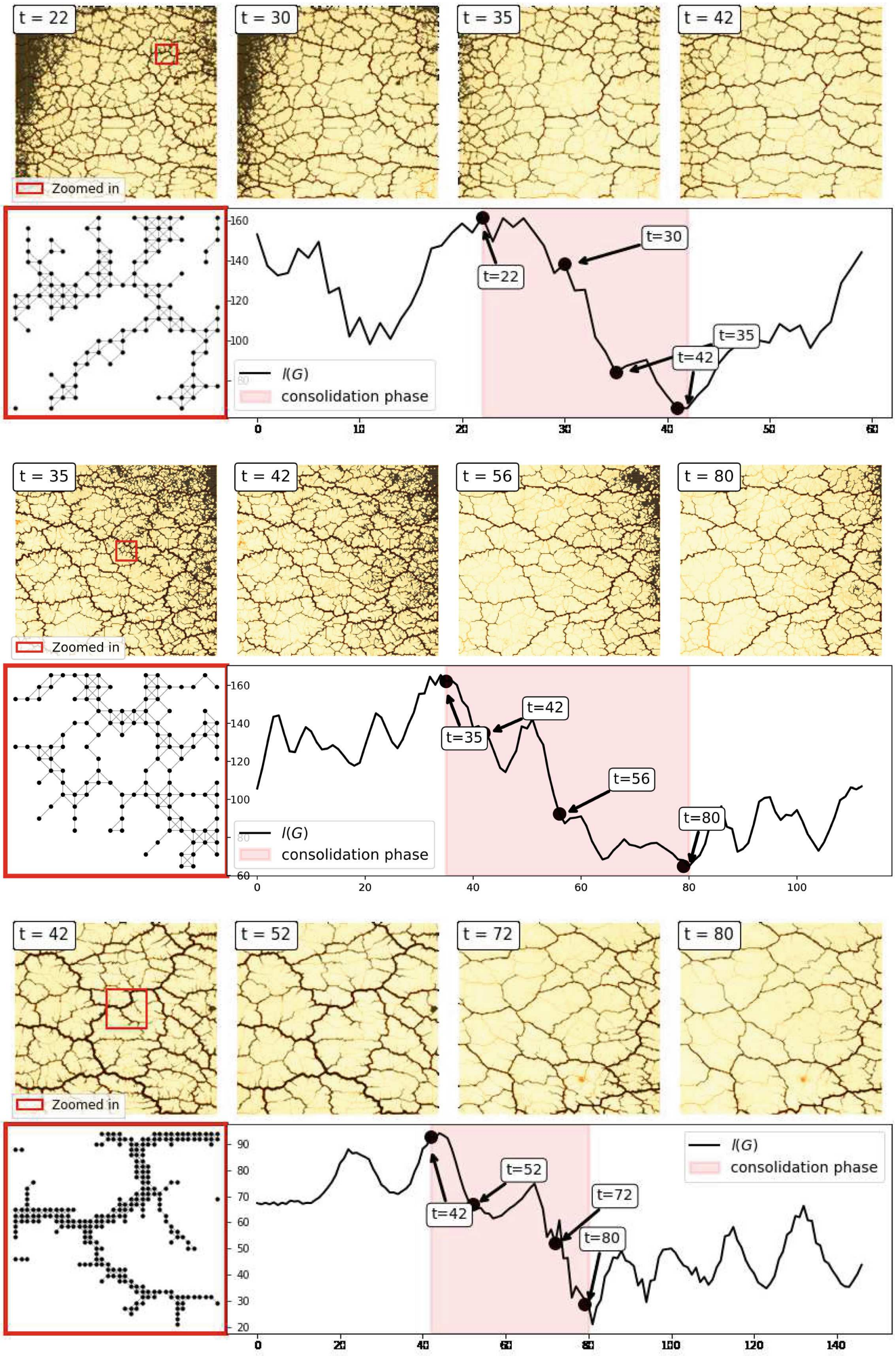


**Fig. 11.** ***P. polycephalum* total length evolution.** From top to bottom: *motion24*, *motion40* and *motion79*. Plots are separated in couples. For every couple, the plots on top show both *P. polycephalum* images and networks extracted from them. The network at the lower leftmost plot is a subsection of the graph shown inside the red rectangle on top. The plot at the bottom shows the total length as a function of time. The red shade in this plot highlights a tentative consolidation phase towards optimality.

## References

1. Villani, C.: Optimal Transport: Old and New, vol. 338. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-71050-9

2. Santambrogio, F.: Optimal transport for applied mathematicians. Birkäuser NY **55**, 58–63 (2015)
3. Peyré, G., Cuturi, M., et al.: Computational optimal transport: with applications to data science. Found. Trends® Mach. Learn. **11**(5–6), 355–607 (2019)
4. Facca, E., Cardin, F., Putti, M.: Towards a stationary Monge-Kantorovich dynamics: the physarum polycephalum experience. SIAM J. Appl. Math. **78**(2), 651–676 (2018)
5. Facca, E., Daneri, S., Cardin, F., Putti, M.: Numerical solution of Monge-Kantorovich equations via a dynamic formulation. J. Sci. Comput. **82**, 1–26 (2020)
6. Facca, E., Cardin, F.: Branching structures emerging from a continuous optimal transport model. J. Comput. Phys. (2020, Submitted)
7. Baptista, D., Leite, D., Facca, E., Putti, M., De Bacco, C.: Network extraction by routing optimization. Sci. Rep. **10**(1), 1–13 (2020)
8. Corson, F.: Fluctuations and redundancy in optimal transport networks. Phys. Rev. Lett. **104**(4), 048703 (2010)
9. Bohn, S., Magnasco, M.O.: Structure, scaling, and phase transition in the optimal transport network. Phys. Rev. Lett. **98**(8), 088702 (2007)
10. Durand, M.: Architecture of optimal transport networks. Phys. Rev. E **73**(1), 016116 (2006)
11. Katifori, E., Szöllősi, G.J., Magnasco, M.O.: Damage and fluctuations induce loops in optimal transport networks. Phys. Rev. Lett. **104**(4), 048704 (2010)
12. Baumgarten, W., Hauser, M.J.: Functional organization of the vascular network of physarum polycephalum. Phys. Biol. **10**(2), 026003 (2013)
13. Baumgarten, W., Ueda, T., Hauser, M.J.: Plasmodial vein networks of the slime mold physarum polycephalum form regular graphs. Phys. Rev. E **82**(4), 046113 (2010)
14. Dirnberger, M., Mehlhorn, K.: Characterizing networks formed by p. polycephalum. J. Phys. D: Appl. Phys. **50**(22), 224002 (2017)
15. Westendorf, C., Gruber, C., Grube, M.: Quantitative comparison of plasmodial networks of different slime molds. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pp. 611–612 (2016)
16. Dirnberger, M., Mehlhorn, K., Mehlhorn, T.: Introducing the slime mold graph repository. J. Phys. D Appl. Phys. **50**(26), 264001 (2017)
17. Nakagaki, T., Yamada, H., Tóth, Á.: Maze-solving by an amoeboid organism. Nature **407**(6803), 470 (2000)
18. Tero, A., Kobayashi, R., Nakagaki, T.: A mathematical model for adaptive transport network in path finding by true slime mold. J. Theor. Biol. **244**(4), 553–564 (2007)
19. Tero, A., et al.: Rules for biologically inspired adaptive network design. Science **327**(5964), 439–442 (2010)
20. Baptista, D., De Bacco, C.: Principled network extraction from images. R. Soc. Open Sci. **8**, 210025 (2021)
21. Lonardi, A., Facca, E., Putti, M., De Bacco, C.: Optimal transport for multi-commodity routing on networks. arXiv preprint arXiv:2010.14377 (2020)
22. Ibrahim, A.A., Lonardi, A., Bacco, C.D.: Optimal transport in multilayer networks for traffic flow optimization. Algorithms **14**(7), 189 (2021)

Baptista and De Bacco
*Applied Network Science* (2023) 8:3
https://doi.org/10.1007/s41109-022-00529-0

Applied Network Science

RESEARCH Open Access

# Convergence properties of optimal transport-based temporal hypergraphs

Diego Baptista* and Caterina De Bacco

*Correspondence:
diego.theuerkauf@tuebingen.mpg.de

Max Planck Institute for Intelligent Systems, Cyber Valley, 72076 Tüebingen, Germany

**Abstract**

We present a method to extract temporal hypergraphs from sequences of 2-dimensional functions obtained as solutions to Optimal Transport problems. We investigate optimality principles exhibited by these solutions from the point of view of hypergraph structures. Discrete properties follow patterns that differ from those characterizing their continuous counterparts. Analyzing these patterns can bring new insights into the studied transportation principles. We also compare these higher-order structures to their network counterparts in terms of standard graph properties. We give evidence that some transportation schemes might benefit from hypernetwork representations. We demonstrate our method on real data by analyzing the properties of hypernetworks extracted from images of real systems.

**Keywords:** Optimal transport theory, Hypergraph theory, Graph theory, Network structure, Network design

## Introduction

Optimal Transport (OT) is a principled theory to compare probability distributions (Kantorovich 1942; Villani 2009; Santambrogio 2015; Peyré et al. 2019). Although this task is usually framed as an optimization problem, recent studies have mapped it within the framework of dynamic partial differential equations (Evans and Gangbo 1999; Facca et al. 2018, 2020, 2021; Tero et al. 2007, 2010). In this context, solutions to a transportation problem are often found as the convergent state of evolving families of functions.

In some scenarios, the steady states of these evolving families are supported in network-shaped structures (Xia 2003, 2014, 2015). Recently, this fact has called the attention of network scientists and graph theorists leading to the development of methods that convert the solutions of OT problems into actual graph structures (Baptista et al. 2020; Leite and De Bacco 2022). This has broadened the available set of tools to understand and solve these transportation problems. Recent studies have shown that common patterns can be unveiled in both the original mathematical setting and in the converted graph structures (Baptista and De Bacco 2021b).

Representations of these functions as sets of dyadic relations have been proven meaningful in various applications (Baptista and De Bacco 2021a; Facca et al. 2021). Nonetheless, traditional dyadic representations may be limited in representing flows of quantities like mass or information as observed in real systems. Various examples of systems where interactions happen between 3 individuals or more are observed in applications as social contagion (de Arruda et al. 2020; Chowdhary et al. 2021), random walks (Carletti et al. 2020; Schaub et al. 2020) or non-linear consensus (Neuhäuser et al. 2020). Understanding the relation between the structure and dynamics taking place on higher-order structures is an active field of research (Taylor et al. 2015; Patania et al. 2017). For instance, key elements controlling dynamics are linked to the heterogeneity of hyperedges' sizes present in their higher-order representations (Patania et al. 2017). These systems are hence best described by hypergraphs, generalizations of networks that encode structured relations among any number of individuals. With this in mind, a natural question to ask is how do OT-based structures perform in terms of higher-order representations?

To help bridge this knowledge gap about higher-order properties of structures derived from OT solutions, we elaborate on the results observed in Baptista and De Bacco (2021b). Specifically, we propose a method to convert the families of 2-dimensional functions into temporal hypernetworks. We enrich the existing network structures associated with these functions by encoding the observed interactions into hyperedges. We study classic hypergraph properties and compare them to the predefined cost functional linked to the transportation problems. Finally, we extend this method and the analysis to study systems coming from real data. We build hypergraph representations of *P. polycephalum* (Westendorf et al. 2016) and analyze their topological features.

## Methods

### The Dynamical Monge-Kantorovich method

#### *The Dynamical Monge-Kantorovich set of equations*

We start by reviewing the basic elements of the mechanism chosen to solve the OT problems. As opposed to other standard optimization methods used to solve this (Cuturi 2013), we use an approach that turns the problem into a dynamical set of partial differential equations. In this way, initial conditions are updated until a convergent state is reached. The dynamical system of equations as proposed by Facca et al. (2018, 2020, 2021), is presented as follows. We assume that the OT problem is set on a continuous 2-dimensional space $\Omega \in \mathbb{R}^2$, and at the beginning, no underlying network structure is observed. This gives us the freedom of exploring the whole space to design an optimal network topology, solution of the transportation problem. The main quantities that need to be specified in input are *source* and *target* distributions. We refer to them as sources and sinks, where a certain mass (e.g. passengers in a transportation network, water in a water distribution network) is injected and then extracted. We denote these with a "forcing" function $f(x) = f^{+}(x) - f^{-}(x) \in \mathbb{R}$, describing the flow-generating sources $f^{+}(x)$ and sinks $f^{-}(x)$. To ensure mass balance it is imposed $\int_{\Omega} f(x)dx = 0$. We assume that the flow is governed by a transient Fick–Poiseuille flux $q = -\mu \nabla u$, where $\mu, u$ and $q$ are called *conductivity* (or *transport density*), *transport potential* and *flux*, respectively. Intuitively, mass is injected through the source, moved based on the conductivity across space, and then extracted through the sink. The way mass moves determines a flux that depends on the pressure exerted on the different points in space; this pressure is described by a potential function.

The set of *Dynamical Monge–Kantorovich* (DMK) equations is given by:

$$-\nabla \cdot (\mu(t,x)\nabla u(t,x)) = f^{+}(x) - f^{-}(x), \tag{1}$$

$$\frac{\partial \mu(t,x)}{\partial t} = [\mu(t,x)\nabla u(t,x)]^{\beta} - \mu(t,x), \tag{2}$$

$$\mu(0,x) = \mu_0(x) > 0, \tag{3}$$

where $\nabla = \nabla_x$. Equation (1) states the spatial balance of the Fick–Poiseuille flux and is complemented by no-flow Neumann boundary conditions. Equation (2) enforces the dynamics of this system, and it is controlled by the so-called *traffic rate* $\beta$. It determines the transportation scheme, and it shapes the topology of the solution: for $\beta < 1$ we have congested transportation where traffic is minimized, whereas $\beta > 1$ induces branched transportation where traffic is consolidated into a smaller amount of space. The case $\beta = 1$ recovers shortest path-like structures. Finally, Eq. (3) constitutes the initialization of the system and can be thought of as an initial guess of the solution.

Solutions $(\mu^{*}, u^{*})$ of Eqs. (1)–(3) minimize the transportation cost function $\mathcal{L}(\mu, u)$ (Facca et al. 2018, 2020, 2021), defined as:

$$\mathcal{L}(\mu, u) := \mathcal{E}(\mu, u) + \mathcal{M}(\mu, u) \tag{4}$$

$$\mathcal{E}(\mu, u) := \frac{1}{2}\int_{\Omega} \mu |\nabla u|^2 dx, \;\; \mathcal{M}(\mu, u) := \frac{1}{2}\int_{\Omega} \frac{\mu^{\frac{(2-\beta)}{\beta}}}{2-\beta} dx. \tag{5}$$

$\mathcal{L}$ can be thought of as a combination of $\mathcal{M}$, the total energy dissipated during transport (or network operating cost) and $\mathcal{E}$, the cost to build the network infrastructure (or infrastructural cost). It is known that this functional's convexity changes as a function of $\beta$. Non-convex cases arise in the branched schemes, inducing fractal-like structures (Facca et al. 2021; Santambrogio 2007). This is the case that we considered in this work, and it is the only one where meaningful network structures, and thus, hypergraphs, can be extracted (Baptista et al. 2020).

## Hypergraph sequences

### *Hypergraph construction*

We define a hypergraph (also, hypernetwork) as follows (Battiston et al. 2020): a *hypergraph* is a tuple $H = (V, E)$, where $V = \{v_1, ..., v_n\}$ is the set of *vertices* and $E = \{e_1, e_2, ..., e_m\}$ is the set of *hyperedges* in which $e_i \subset V, \forall i = 1, ..., m$, and $|e_i| > 1$. If $|e_i| = 2, \forall i$ then $H$ is simply a graph. We call *edges* those hyperedges $e_i$ with $|e_i| = 2$ and *triangles*, those with $|e_i| = 3$. We refer to the *1-skeleton* of $H$ as the *clique expansion* of $H$. This is the graph $G = (V, E_G)$ made of the vertices $V$ of $H$, and of the pairwise edges built considering all the possible combinations of pairs that can be built from each set of nodes defining each hyperedge in $E$.

Let $\mu$ be the conductivity found as a solution of Eqs. (1)–(3). As previously mentioned, $\mu$ at convergence regulates where the mass should travel for optimal transportation. Similar to Baptista and De Bacco (2021b), we turn this 2-dimensional function into a different data structure, namely, a hypergraph. This is done as follows: consider $G(\mu) = (V_G, E_G)$ the network extracted using the method proposed in Baptista et al. (2020). We define $H(\mu)$ as the tuple $(V_H, E_H)$ where $V_H = V_G$ and $E_H = E_G \cup T_G$, s.t., $T_G = \{(u, v, w) : (u, v), (v, w), (w, u) \in E_G, \}$. In words, $H(\mu)$ is the graph $G(\mu)$ together with all of its triangles. This choice is motivated by the fact that the graph-extraction method proposed in Baptista et al. (2020) uses triangles to discretize the continuous space $\Omega$, which can have a relevant impact on the extracted graph or hypergraph structures. Hence, triangles are the natural sub-structure for hypergraph constructions. The method proposed in this work is valid for higher-order structures beyond triangles. Exploring how these additional structures impact the properties of the resulting hypergraphs is left for future work.

Figure 1 shows an example of one of the studied hypergraphs. The red shapes represent the different triangles of $H(\mu)$. Notice that, although we consider here the case where $|e| \leq 3$ for each hyperedge $e$—for the sake of simplicity—higher-order structures are also well represented by the union of these elements, as shown in the right panels of the figure.

Since this hypergraph construction method is valid for any 2-dimensional transport density, we can extract a hypergraph not only from the convergent $\mu$ but also at any time step before convergence. This then allows us to represent optimal transport sequences as hypergraphs evolving in time, i.e. temporal hypernetworks.

### Hypergraph sequences

Formally, let $\mu(x, t)$ be a *transport density* (or *conductivity*) function of both time and space obtained as a solution of the DMK model. We denote it as the sequence $\{\mu_t\}_{t=0}^{T}$, for some index $T$ (usually taken to be that of the convergent state). Each $\mu_t$ is the $t$-th update of our initial guess $\mu_0$, computed by following the rules described in Eqs. (1)–(3). This determines a sequence of hypernetworks $\{H(\mu_t)\}_{t=0}^{T}$ extracted from $\{\mu_t\}_{t=0}^{T}$ with the extraction method proposed in Baptista et al. (2020). Figure 2 shows three hypergraphs built from one of the studied sequences $\{\mu_t\}$ using this method at different time steps. The corresponding OT problem is that defined by the (filled and empty) circles: mass is injected in the bottom left circle and must be extracted at the highlighted destinations. On the top row, different updates (namely, $t = 12, 18, 26$) of the solution are shown. They are defined on a discretization of $[0, 1]^2$. Darkest colors represent their support.

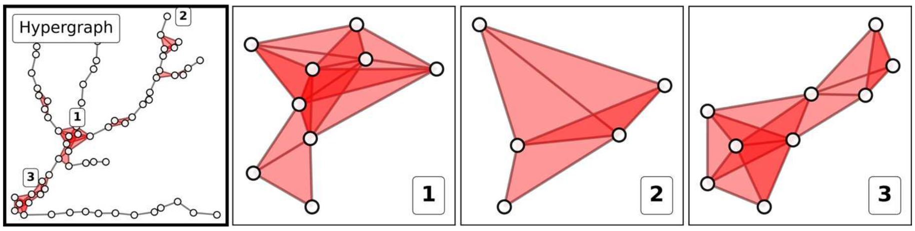


**Fig. 1** Hypernetwork construction. Higher order structures are built using edges and triangles as hyperedges. The leftmost panel shows one of the studied graphs together with the triangles (in red) used. The subsequent panels highlight different clusters of triangles that can be seen in the main hypergraph

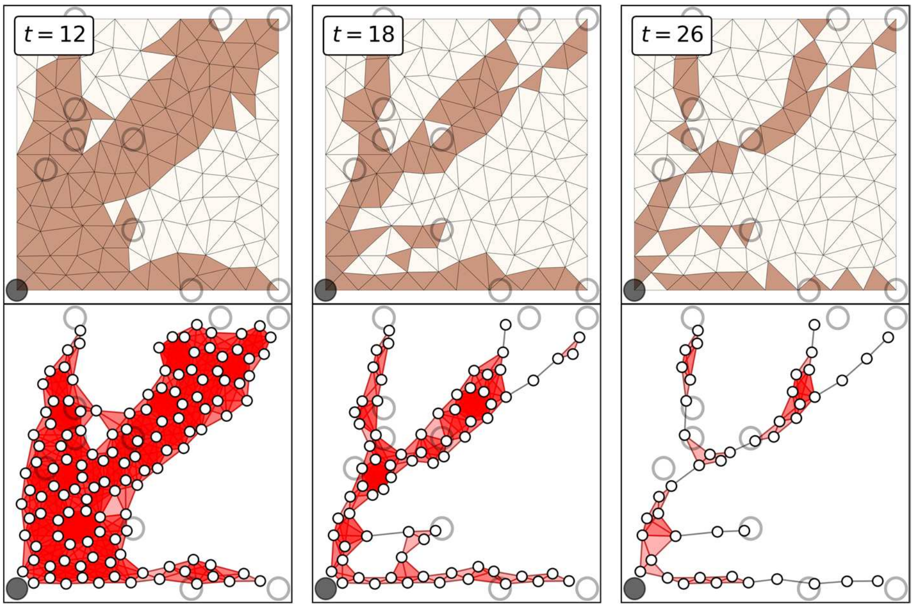


**Fig. 2** Temporal hypergraphs. Top row: different timestamps of the sequence $\{\mu_t\}$; triangles are a discretization of $[0,1]^2$. Bottom row: hypergraphs extracted for $\mu_t$ at the time steps displayed on the top row; triangles are highlighted in red. In both rows, filled and empty circles correspond to the support of $f^+$ and $f^-$, i.e. sources and sinks, respectively. This sequence is obtained for $\beta = 1.5$

Hypergraphs extracted from these functions are displayed at the bottom row. As can be seen, only edges (in gray) and triangles (in red) are considered as part of $H(\mu_t)$. Notice that the larger the $t$ is, the less dense the hypergraphs are, which is expected for a uniform initial distribution $\mu_0$ and branched OT ($\beta > 1$) (Facca et al. 2021).

## Graph and hypergraph properties

We compare hypergraph sequences to their correspoding network counterparts (defined as described in the previous paragraph). We analyze the following main network and hypergraph properties for the different elements in the sequences and for different sequences. Denote with $G = (V_G, E_G)$ and $H = (V_H, E_H)$ one of the studied graphs and hypergraphs belonging to some sequence $\{G(\mu_t)\}_{t=0}^T$ and $\{H(\mu_t)\}_{t=0}^T$, respectively. We consider the following network properties:

1. $|E_G|$, total number of edges;
2. Average degree $d(G)$, the mean number of neighbors per node;
3. Average closeness centrality $c(G)$: let $v \in V_G$, the closeness centrality of $v$ is defined as $\sum_{u \in V_G} 1/d(u,v)$, where $d(u,v)$ is the shortest path distance between $u$ and $v$.

Hypernetwork properties can be easily adapted from the previous definitions with the help of generalized adjacency matrices and line graphs (Aksoy et al. 2020). Let $H$ be a hypergraph with vertex set $V = \{1, .., n\}$ and edge set $E = \{e_1, ..., e_m\}$. We define the generalized *node s-adjacency matrix* $A_s$ of $H$ as the binary matrix of size $n \times n$, s.t., $A_s[i][j] = 1$ if $i$ and $j$ are part of at least $s$ shared hyperedges; $A_s[i][j] = 0$, otherwise.

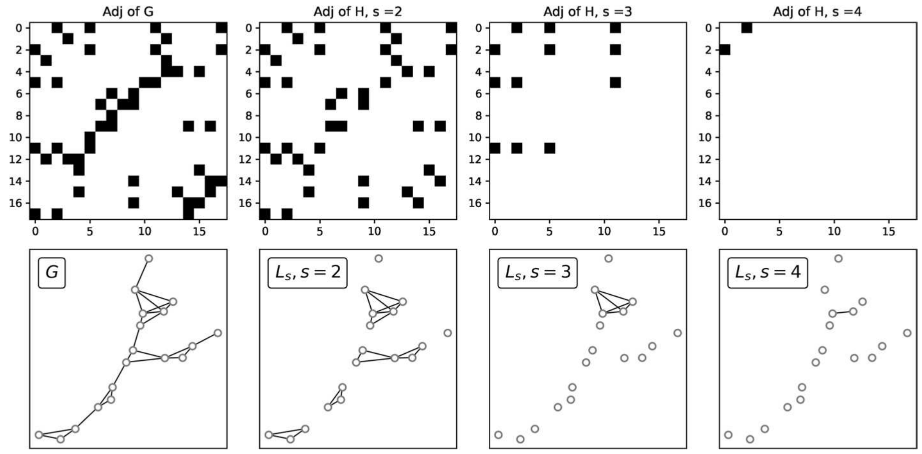


**Fig. 3** Adjacency matrices and line graphs. Top: generalized node $s$-adjacency matrices for different values of $s$ from a given toy graph $G$. Bottom, from left to right: reference network $G$, and $s$-line graphs for $s = 2, 3$, and 4

We define the *s-line graph* $L_s$ as the graph generated by the adjacency matrix $A_s$. Notice that $A_1$ corresponds to the adjacency matrix of $H$'s skeleton (which is $L_1$). Figure 3 shows a family of adjacency matrices together with the line graphs generated using them. We can then define hypergraphs properties in the following way:

1. $|E_H|$, total number of hyperedges;
2. $|T| = |\{e \in E_H : |e| = 3\}|$, total number of triangles;
3. $S = \sum_{t \in T} a(t)$, *covered area*, where $a(t)$ is the area of the triangle $t$;
4. Average degree $d_s(H)$, the mean number of incident hyperedges of size greater or equal than $s$ per node;
5. Average closeness centrality $c_s(H)$: let $v \in V_H$, the closeness centrality of $v$ is defined as its closeness centrality in $L_s$.

$S$ can be defined in terms of any other property of a hyperedge, e.g. a function of its size $|e|$. Here we consider the area covered by a hyperedge to keep a geometrical perspective. On the other hand, this area $S$ can be easily generalized to hyperedges with $|e_i| > 3$ by suitably changing the set $T$ in the summation, e.g. by considering structures containing four nodes. As for the centrality measures, we focus our attention to compare the case $s > 1$ against $s = 1$, as the latter traces back to standard graph properties and we are interested instead to investigate what properties are inherent to hypergraps. Figure 4 shows values of the $d_s(H)$ and $c_s(H)$ for convergent hypergraphs $H$ (obtained from different values of $\beta$) together with the degree and closeness centrality of their correspondent graph versions. The considered hypergraphs are displayed in the top row of the figure. As can be seen in the figure, patterns differ considerably for different values of $\beta$. As $s$ controls the minimum number of shared connections for different nodes in the networks, the higher this number, the more restrictive this condition becomes, thus leading to more disconnected line graphs. In the case of the $s$-degree centrality, we observe decreasing values for increasing $s$, with nodes with the highest centrality having much higher values than nodes less central.

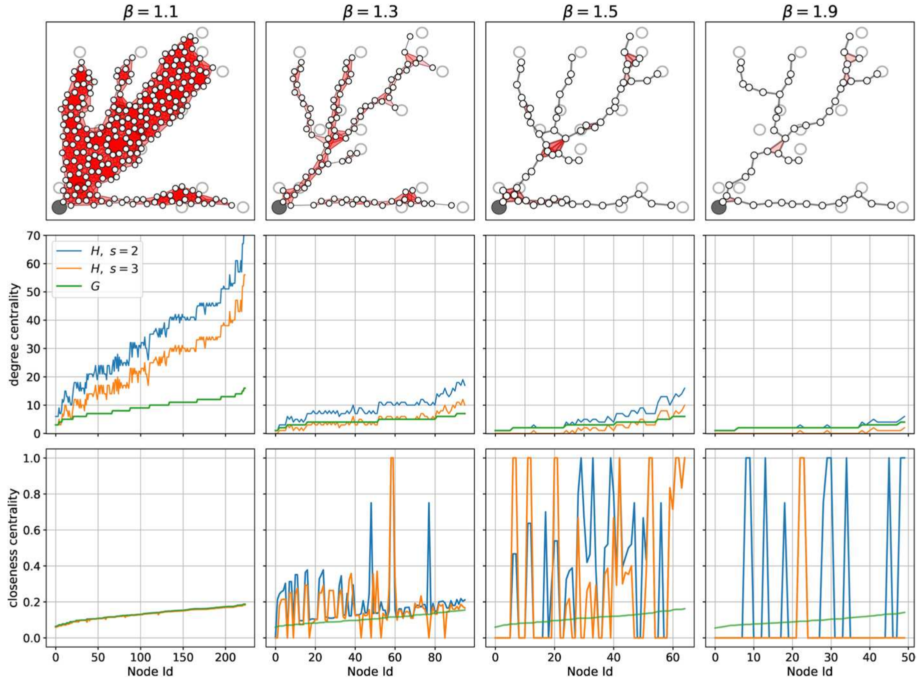


**Fig. 4** Graph and Hypergraph properties. Top row: optimal hypernetworks obtained with different traffic rates. Center and bottom rows: degree distributions and closeness distributions for the hypernetworks shown on the top row, and their 1-skeletons. The node labels in the *x*-axis of the center and bottom rows are sorted by their degree of centrality values

For both $s = 2, 3$ we observe higher values than nodes in $G$. This follows from the fact that once hyperedges are added to $G$, the number of incidences per node can only increase. Centrality distributions strongly depend on $\beta$. For small values—more distributed traffic ($\beta = 1.1$)—the number of hyperedges per node remains larger than the number of regular edges connected to it. But if traffic is consolidated on less space ($\beta = 1.9$), then very few hyperedges are found. This suggests that the information learned from hypergraphs that is distinct to that contained in the graph skeleton is influenced by the chosen traffic regime.

As for the closeness centrality distribution, this resembles that of $G$ for small values of $\beta$, regardless $s$. For higher $\beta$ it switches towards an almost binary signal. Thus, nodes tend to become more central as $\beta$ increases, suggesting that adding hyperedges to networks $G$ leads to shorter distances between nodes. The loss of information seen for the highest values of $s$ is due to the fact that the line graphs $L_s$ become disconnected with many small connected components. In these cases, the closeness centrality of a node is either 0 if it is isolated, or proportional to the diameter of the small connected component where it lives in.

### *Convergence criteria*

Numerical convergence of the DMK Eqs. (1)–(3) is usually defined by fixing a threshold $\tau$. The updates are considered enough once the cost associated to them does not change more ($\leq \tau$) than that of the previous time step. As it is usually the case when this threshold is too small ($\tau = 10^{-12}$ in our experiments), the cost or the network structure may consolidate to a constant value earlier than algorithmic convergence. Similar to Baptista and De Bacco (2021b), to meaningfully establish when is hypergraph optimality reached, we consider as convergence time the first time step when the transport cost, or a given network property, reaches a value that is smaller or equal to a certain fraction $p$ of the value reached by the same quantity at algorithmic convergence (in the experiments here we use $p = 1.05$). We refer to $t_{\mathcal{L}}$ and $t_{P}$ for the convergence in times in terms of cost function or a network property, respectively.

## Results

To test the properties presented in the previous section and understand their connection to transportation optimality, we synthetically generate a set of optimal transport problems, determined by the configuration of sources and sinks. As done in Baptista and De Bacco (2021b), we fix a source's location and sample several points in the set $[0,1]^2$ to be used as sinks' locations. Let $S = \{s_0, s_1, ..., s_M\}$ be the set of locations in the space $[0,1]^2$, and fix a positive number $0 < r$. We define the distributions $f^+$ and $f^-$ as $f^+(x) \propto \mathbb{1}_{R_0}(x)$, and $f^-(x) \propto \sum_{i>0} \mathbb{1}_{R_i}(x)$, where $\mathbb{1}_{R_i}(x) := 1$, if $x \in R_i$, and $\mathbb{1}_{R_i}(x) := 0$, otherwise; $R_i = C(s_i, r)$ is the circle of center $s_i$ and radius $r$. The value of $r$ is chosen based on the used discretization, and as mentioned before, the centers are sampled uniformly at random. The symbol $\propto$ stands for proportionality and is used to ensure that $f^+$ and $f^-$ are both probability distributions. The transportation cost is that of Eq. (4).

### Synthetic OT problems

The set of transportation problems considered in our experiments consists of 100 source-sink configurations. We place the location of the source $s_0 = (0,0)$ (i.e. the support of $f^+$ at (0, 0)), and sample 15 points $s_1, s_2, ..., s_M$ uniformly at random from a regular grid. By sampling them from the nodes of the grid, we ensure that two different locations are at a safe distance so they are considered different once the space is discretized. We initialize $\mu_0(x) = 1, \forall x$ to be a uniform distribution on $[0,1]^2$. This can be interpreted as a non-informative initial guess for the solution. Starting from $\mu_0$, we compute a maximum of 300 updates. Depending on the chosen traffic rate $\beta$ more or fewer iterations can be needed. We claim that the sequence $\{\mu_t\}_{t=0}^{T}$ *converges* to a certain function $\mu^*$ at iteration $T$ if either $|\mu_T - \mu_{T-1}| < \tau$, for a *tolerance* $\tau \in (0,1]$, or $T$ reaches the mentioned maximum. For the experiments reported in this manuscript, the tolerance $\tau$ is set to be $10^{-12}$. Given the dependence of the solution of traffic constraints, a wide range of values of $\beta$ is considered. Namely, we study solutions obtained from low traffic cases ($\beta = 1.1$, and thus, less traffic penalization) to large ones ($\beta = 1.9$), all of them generating branched transportation schemes. Our 100 problems are linked to a total of 900 hypergraph sequences, each of them containing between 50 and 80 higher-order structures.

### Convergence: transport cost vs hypernetwork properties

As presented in Baptista and De Bacco (2021b), we show a comparison between hypernetwork properties and the cost function minimized by the dynamics, where convergence times are highlighted (Fig. 5). We focus on the property $S$, the area of the surface covered by the triangles in $H$. This quantity is influenced by both the amount of triangles

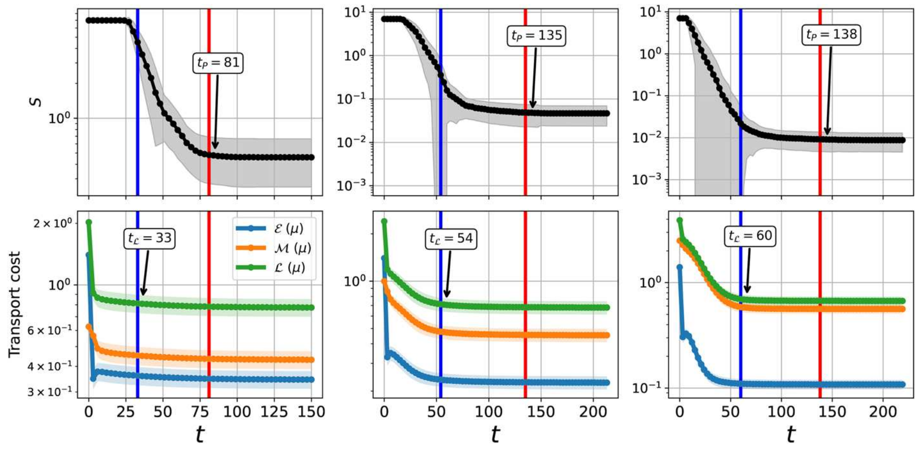


**Fig. 5** Covered area and Lyapunov cost. Mean (markers) and standard deviations (shades around the markers) of the covered area $S$ (top plots) and of the Lyapunov cost, energy dissipation $\mathcal{E}$ and structural cost $\mathcal{M}$ (bottom plots), as functions of time $t$. Means and standard deviations are computed on the set described in Paragraph *Synthetic OT problems*. From left to right: $\beta = 1.2, 1.5$ and 1.8. Red and blue lines denote $t_P$ and $t_{\mathcal{L}}$

(hence of hyperedges) and their distribution in space. Hence, it is a good proxy for how hypergraph properties change both in terms of iteration time and as we tune $\beta$. We observe that $t_P > t_{\mathcal{L}}$ in all the cases, i.e. convergence in terms of transportation cost is reached earlier than the convergence of the topological property. Similar behaviors are seen for other values of $\beta \in [1.1, 1.9]$ and other network properties (see "Appendix"). Similar to DMK-based network properties, the covered area's decay is faster for the smallest values of $\beta$. This is expected, given the convexity properties of $\mathcal{L}$ (Facca et al. 2018, 2020, 2021). However, the transport cost decays even faster, in a way that the value of $S$ is still far away from convergence in the congested transportation case (small $\beta$).

Notice that $S$ remains stable after the first few iterations, and then it starts decreasing at different rates (depending on $\beta$) until reaching the converged value. This suggests that the dynamics tend to develop thick branches—covering a large area— at the beginning of the evolution, and then it slowly compresses them until reaching the optimal topologies.

These different convergence rates for $S$ and $\mathcal{L}$ may prevent construction of converged hypernetwork topologies: if the solver is stopped at $t_{\mathcal{L}} < t_P$, the resulting hypergraphs $H(\mu_t)$, $t = t_{\mathcal{L}}$ would mistakenly cover a surface larger than that covered by the convergent counterpart ($H(\mu_t)$, for $t \geq t_P$). This scenario is less impactful for larger values of $\beta$, although in these scenarios $H$ is much more similar to a regular graph, because of the small number of higher-order structures. Topological differences between converged hypernetworks can be seen in Fig. 4.

Finally, we observe that both $t_{\mathcal{L}}(\beta)$ and $t_P(\beta)$ are increasing functions on $\beta$. This is expected since the larger the traffic rate is, the longer it takes for the sequences to converge. This particular behavior matches what is shown in Baptista and De Bacco (2021b) in the case of $t_{\mathcal{L}}$, but this is not the case for $t_P(\beta)$: it was observed a non-monotonic behavior in the network case.

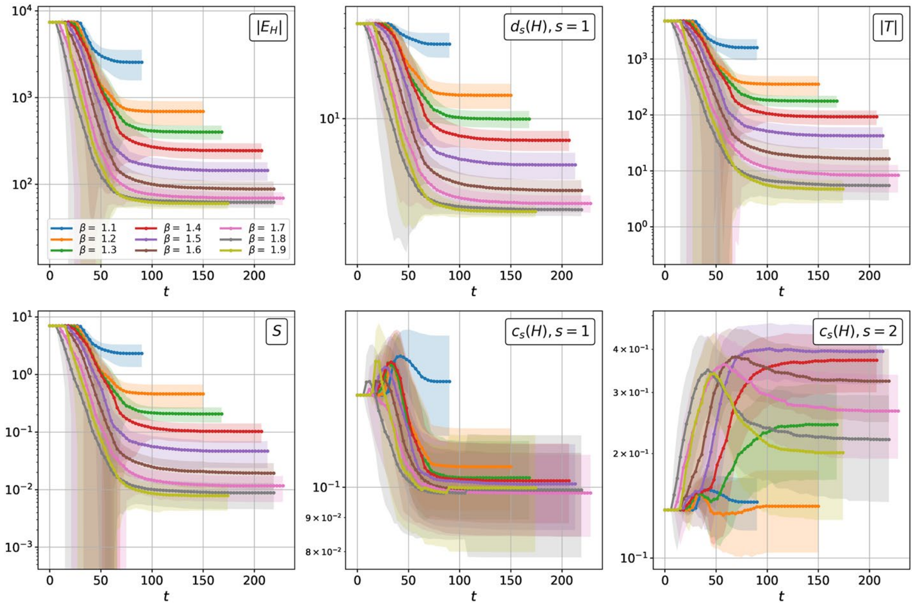


**Fig. 6** Evolution of hypernetwork properties. Mean (markers) and standard deviations (shades around the markers) of number of hyperedges $|E_H|$ (upper left), number of triangles $|T|$ (upper center), covered area $S(H)$ (upper right), average 2-degree $d_2(H)$ (lower left), average 1-closeness centrality $c_1(H)$(lower center) and 2-closeness centrality $c_2(H)$(lower right), computed for different values of $\beta$ as a function of time

### Convergence behavior of hypernetwork properties

Figure 6 shows how the various network properties change depending on the traffic rate. Mean values and standard deviations are computed across times, for a fixed value of $\beta$. As shown, the number of hyperedges, number of triangles, covered area, and average 1-degree exhibit decreasing patterns as functions of $t$. As a consequence, transport optimality can be thought of as reaching minimum states on the mentioned hypernetwork properties. Another clear feature of these functions is related to the actual converged values: the larger the $\beta$ is, the smaller these metrics become. This is explained by a cost function increasingly encouraging consolidations of paths on fewer edges. Notice also that the gap between these converged values signals a non-linear dependence on the outputs of the dynamics; e.g., a converged hypernetwork obtained for $\beta = 1.1$. loses many more hyperedges if the traffic rate is then set to 1.2, whereas this loss would not be that large if $\beta = 1.2$ is increased to 1.3. The nature of these gaps is substantially different depending on the property itself. This also shows that certain properties better reveal the distinction between different optimal traffic regimes.

The behavior of the closeness centralities is distinctly different than that of the other properties. While its initial values are the same for all values of $\beta$ (similar to the previous properties), no clear trend can be found as time increases. For $s = 1$, on average $\beta = 1.1$ generates sequences that tend to recover initial values after increasing and then decreasing behavior. For the other traffic rates, we observe different patterns. Notice that $s-$closeness centrality on the hypergraph for $s = 1$ is the same as the classic closeness centrality on the skeleton of it. Thus, these rather noisy patterns are not due to the addition of hyperedges. On the other hand, for $s = 2$ the average centrality shows increasing curves. This may be due to $L_s$ getting increasingly disconnected with small connected components. Therefore, the larger $s$, the closer the nodes are seen (see Fig. 3). Moreover, in this case small values of $\beta$ lead to more stable closeness centrality values, showing the impact of $\beta$ in building higher-order structures. While different values of $\beta$ lead to different behaviors of the hypergraph properties (e.g. decreasing degrees and amount of hyperedges for increasing $\beta$) we remark that choosing the value of $\beta$ should depend on the application at hand. The analysis performed here showcases how this choice may impact the resulting topologies. This can help practitioners to visualize possible consequences in terms of downstream analysis on the transportation properties of the underlying infrastructure.

## *P. polycephalum* hypernetworks

We now analyze hypernetworks extracted from images of real data. We are interested in the evolution of the area covered by triangles in the sequences $\{H(\mu_t)\}_{t=0}^{T}$ extracted from real images of the slime mold *P. polycephalum*. The behavior of this organism is the inspiration of the modeling ideas of the DMK equations described in Methods. It has been shown that these slime molds follow a similar optimization strategy as that captured by the DMK dynamics while foraging for food in 2D surfaces (Nakagaki et al. 2000; Tero et al. 2007, 2010). We extract hypernetworks from images using the idea described in Methods but instead of applying (Baptista et al. 2020) to obtain the networks, we use the method proposed by Baptista and De Bacco (2021a) which takes images as input. This pipeline uses the color intensities of the different image pixels to build a graph, by connecting adjacent meaningful nodes. We dedicate our attention to 4 image sequences from the Slime Mold Graph Repository (Dirnberger et al. 2017). The sequences are then describing the evolution of a *P. polycephalum* placed in a rectangular Petri dish. Each image, and thus each hypernetwork, is a snapshot of the movement of this organism over periods of 120 seconds.

We study the covered area for every one of the 4 sequences, and plot the results for one of them (namely, image set *motion12*; see "Appendix") in Fig. 7. We highlight 4 times along the property sequence to display the used images together with the corresponding hypernetworks. The lower leftmost plot shows a subsection of one of the studied snapshots. As can be seen there, this subhypernetwork topology exhibits a significant number of hyperedges of dimension 3, mainly around the thickest parts of the slime mold. On the other side, in the lower rightmost plot, the evolution of $S$ is overall decreasing in time (similar results are obtained for other sequences, as shown in the "Appendix"). This suggests that the thicker body parts tend to get thinner as the *P. polycephalum* evolves into a consolidated state. This pattern resembles what is shown above for the synthetic data, i.e. the covered area tends to decrease as time evolves similar to the behavior of the DMK-based hypernetwork sequence. This suggests that the DMK model realistically mirrors a consolidation phase towards optimality of real slime molds (Dirnberger et al. 2017).

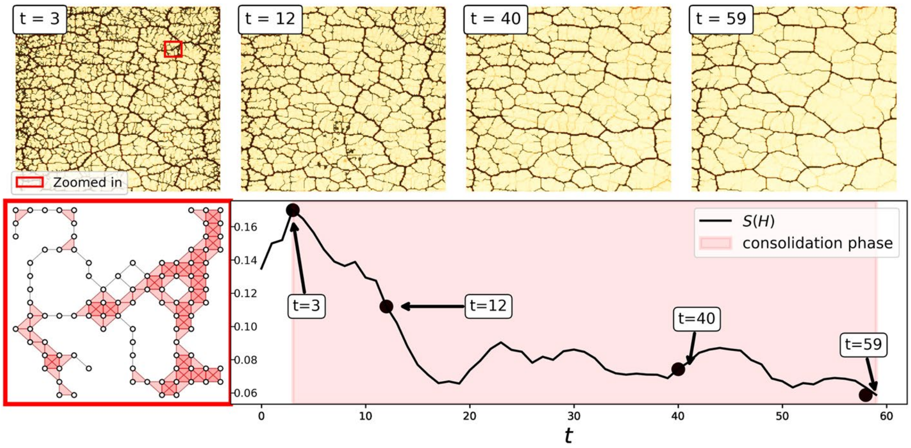


**Fig. 7** *P. polycephalum* hypergraphs. On top: *P. polycephalum* images and hypernetworks extracted from them. Bottom left: a zoomed-in part of the hypergraph shown inside the red rectangle on top. Bottom right: covered area as a function of time. The red shade highlights a tentative consolidation phase towards optimality

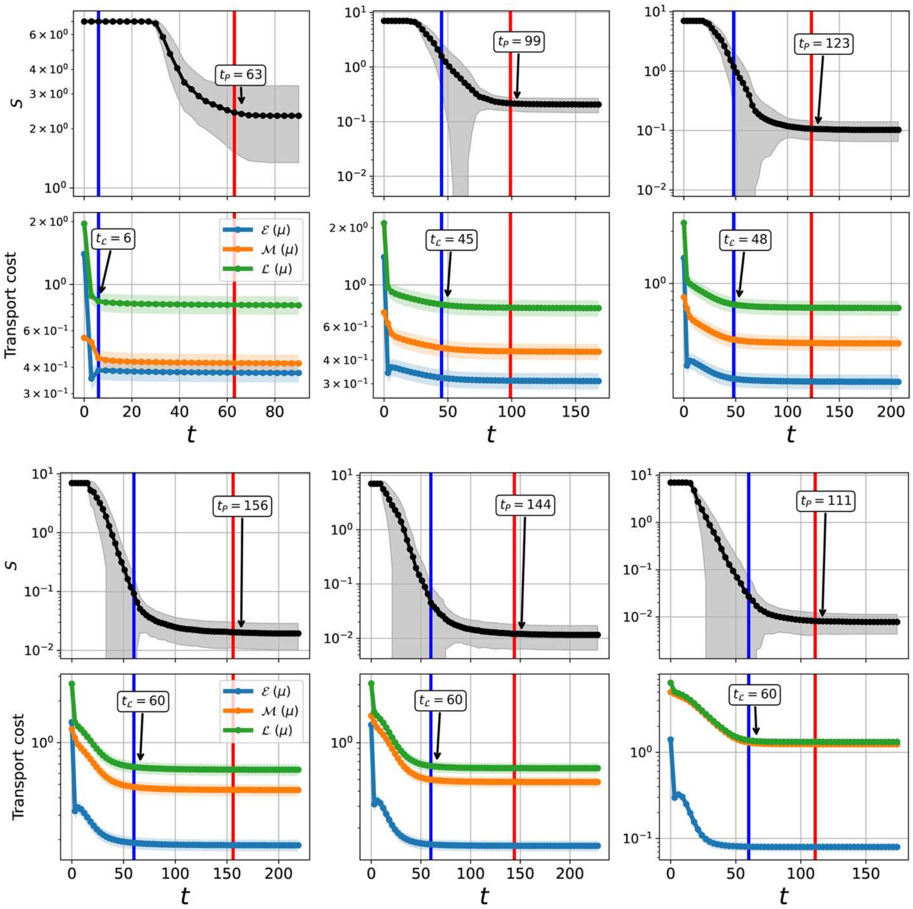


**Fig. 8** *S* and Lyapunov cost. First and second top-down rows: from left to right we see $\beta = 1.1, 1.3$ and 1.4. Third and fourth top-down rows: from left to right we see $\beta = 1.6, 1.7$ and 1.9. First and third top-down rows: mean and standard deviation of *S* as a function of time *t*; Second and fourth top-down rows: Mean and standard deviation of the Lyapunov cost $\mathcal{L}$, energy dissipation $\mathcal{E}$ and structural cost $\mathcal{M}$ of transport densities. Red and blue lines denote $t_P$ and $t_{\mathcal{L}}$ for $p = 1.05$

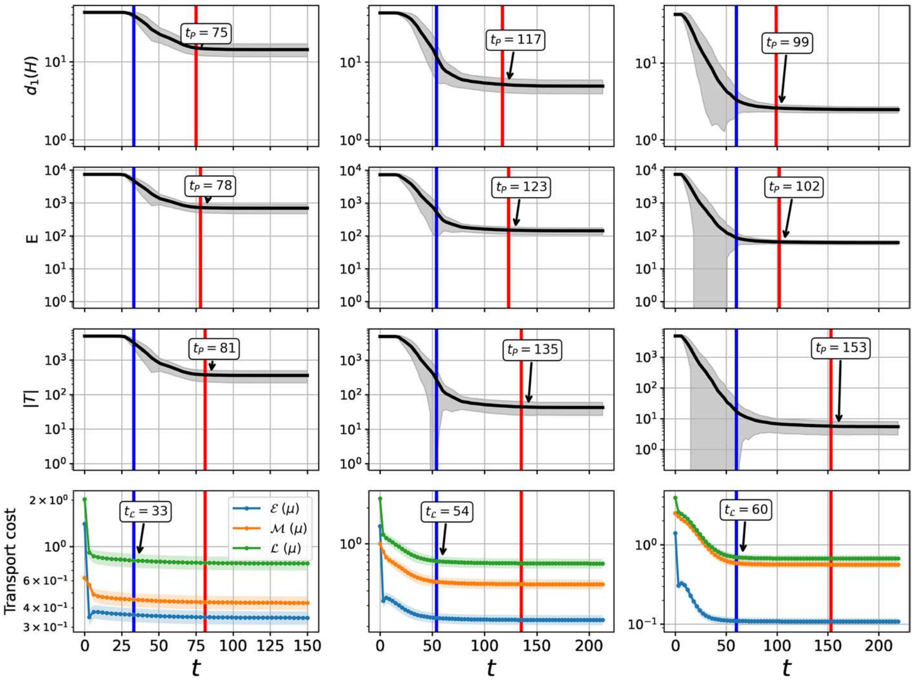


**Fig. 9** Other hypernetwork properties and Lyapunov cost. From left to right: $\beta = 1.2, 1.5$ and 1.8. From top to bottom: Mean and standard deviation of the average degree $d_1(H)$, number of hyperedges $|E_H|$, number of triangles $|T|$, and the Lyapunov cost $\mathcal{L}$, energy dissipation $\mathcal{E}$ and structural cost $\mathcal{M}$. Red and blue lines denote $t_P$ and $t_\mathcal{L}$ for $p = 1.05$

## Conclusions

We proposed a method to build higher-order structures from OT sequences. This method maps every member of the sequence into a hypergraph, outputting a temporal hypernetwork. We analyzed standard hypergraph properties on these temporal families and compared them to their continuous counterparts. We showed that convergence in terms of transportation cost tends to happen faster than that given by the covered area of the hypernetworks. This suggests that the dynamics used to solve the OT problems concentrates the displaced mass into main branches, and once this task is carried out, it slightly reduces the area covered by them. We studied this and other hypergraph properties, and compared them to those of their network versions. In some cases, hypernetworks reveal more information about the topology at convergence. This suggests that hypernetworks could be a better alternative representation to solutions of OT problems for some transportation schemes. The conclusions found in this work may further enhance our comprehension of OT solutions and the links between this field and that of hypergraphs.

## Appendix

### Covered area for other values of $\beta$

We present in this section a similar plot to that of Fig. 5—comparing the covered area and the cost function— for other values of $\beta$. As mentioned there, $S$ shows decreasing behaviors for which $t_P > t_\mathcal{L}$ holds true (see Fig. 8).

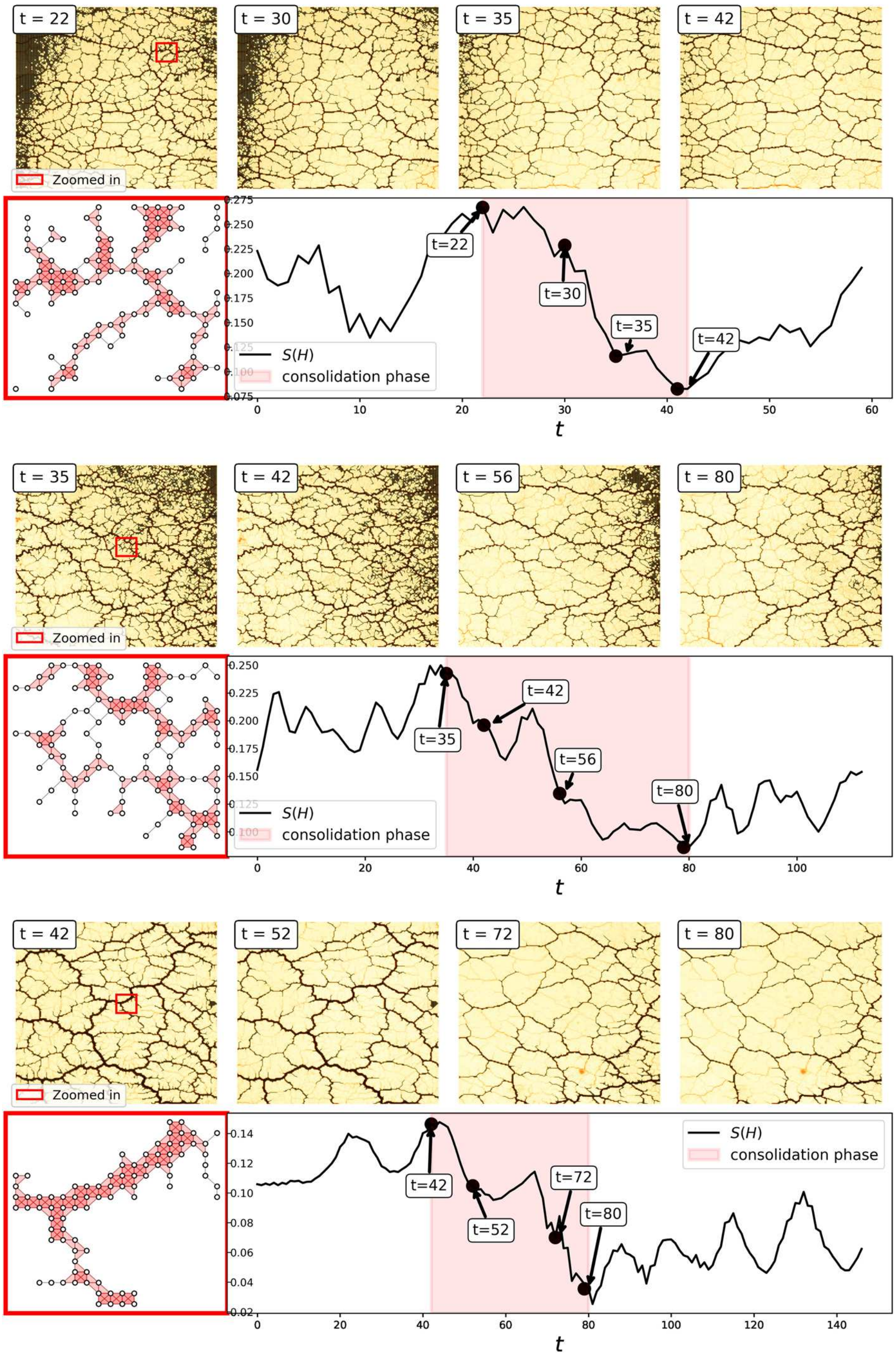


**Fig. 10** *P. polycephalum S* evolution. From top to bottom: *motion24*, *motion40* and *motion79*. Plots are separated in couples. For every couple, the plots on top show both *P. polycephalum* images and hypernetworks extracted from them. The hypernetwork at the lower leftmost plot is a subsection of the hypergraph shown inside the red rectangle on top. The plot at the bottom shows the covered area as a function of time. The red shade in this plot highlights a tentative consolidation phase towards optimality

### Additional hypernetwork properties

In this section we extend the comparison between the cost function—minimized by the dynamics—and hypernetwork properties (see Fig. 9). As mentioned in the main manuscript, similar monotonic behaviors can be observed in these cases.

### *P. polycephalum* hypernetworks

#### *Data information*

We explain in this section further details about the analyzed real data.

The images are taken from the Slime Mold Graph Repository (Dirnberger et al. 2017) as mentioned in the main manuscript. We study 4 $\{H_i\}_i^T$ sequences of different lengths. The length ($T$) varies depending on the number of images included in the sequence. This is because different experiments need more o fewer shots. These experiments, as mentioned in the repository's documentation, consist of placing a slime mold inside a Petri dish with a thin sheet of agar where no food is provided. Slime mold's exploration of the dish, as explained by the creators, is unbiased, due to the lack of food. Given that this organism is initially placed along one of the short edges of the rectangular dish, the experiment is considered to be finished once the plasmodium reaches the other short side. No more pictures are taken after this happens (Fig. 10).

#### *Hypergraph extraction*

We used the image sets *motion12, motion24, motion40* and *motion79*, located in the repository, to build the studied hypernetworks. These sets contain a number of images ranging from 60 to 150. Hypernetworks are then extracted using the *Img2net* algorithm described in Baptista and De Bacco (2021a) as mentioned in the main manuscript, using the same configuration described in Baptista and De Bacco (2021b).

**Acknowledgements**
The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Diego Baptista.

**Author contributions**
DB and CDB derived the model, analyzed results, and wrote the manuscript. DB conducted the experiments. Both authors read and approved the final manuscript.

**Funding**
Open Access funding enabled and organized by Projekt DEAL.

**Availability of data and materials**
The code used for the analysis was taken from https://github.com/Danielaleite/Nextrout/. Data used in the analysis was taken from Dirnberger et al. (2017).

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
The authors declare that they have no competing interests.

Received: 28 March 2022 Accepted: 22 December 2022
Published online: 11 January 2023

**References**

Aksoy SG, Joslyn C, Marrero CO, Praggastis B, Purvine E (2020) Hypernetwork science via high-order hypergraph walks. EPJ Data Sci 9(1):16

Baptista D, De Bacco C (2021a) Principled network extraction from images. R Soc Open Sci 8:210025

Baptista D, De Bacco C (2021b) Convergence properties of optimal transport-based temporal networks. In: International conference on complex networks and their applications. Springer, pp 578–593

Baptista D, Leite D, Facca E, Putti M, De Bacco C (2020) Network extraction by routing optimization. Sci Rep 10(1):1–13

Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young J-G, Petri G (2020) Networks beyond pairwise interactions: structure and dynamics. Phys Rep 874:1–92

Carletti T, Battiston F, Cencetti G, Fanelli D (2020) Random walks on hypergraphs. Phys Rev E 101:022308. https://doi.org/10.1103/PhysRevE.101.022308

Chowdhary S, Kumar A, Cencetti G, Iacopini I, Battiston F (2021) Simplicial contagion in temporal higher-order networks. J Phys Complex 2(3):035019

Cuturi M (2013) Sinkhorn distances: lightspeed computation of optimal transport. In: Advances in neural information processing systems, vol 26

de Arruda GF, Petri G, Moreno Y (2020) Social contagion models on hypergraphs. Phys Rev Res 2:023032. https://doi.org/10.1103/PhysRevResearch.2.023032

Dirnberger M, Mehlhorn K, Mehlhorn T (2017) Introducing the slime mold graph repository. J Phys D Appl Phys 50(26):264001

Evans LC, Gangbo W (1999) Differential equations methods for the Monge–Kantorovich mass transfer problem, vol 653. American Mathematical Society, Providence

Facca E, Cardin F, Putti M (2018) Towards a stationary Monge–Kantorovich dynamics: the *Physarum polycephalum* experience. SIAM J Appl Math 78(2):651–676

Facca E, Daneri S, Cardin F, Putti M (2020) Numerical solution of Monge–Kantorovich equations via a dynamic formulation. J Sci Comput 82(68):1–26

Facca E, Cardin F, Putti M (2021) Branching structures emerging from a continuous optimal transport model. J Comput Phys 447:110700

Kantorovich L (1942) On the transfer of masses. Dokl Akad Nauk 37:227–229 (**in Russian**)

Leite D, De Bacco C (2022) Revealing the similarity between urban transportation networks and optimal transport-based infrastructures. arXiv:2209.06751

Nakagaki T, Yamada H, Tóth Á (2000) Maze-solving by an amoeboid organism. Nature 407(6803):470–470

Neuhäuser L, Mellor A, Lambiotte R (2020) Multibody interactions and nonlinear consensus dynamics on networked systems. Phys Rev E 101:032310. https://doi.org/10.1103/PhysRevE.101.032310

Patania A, Vaccarino F, Petri G (2017) Topological analysis of data. EPJ Data Sci 6(1):1–6

Peyré G, Cuturi M et al (2019) Computational optimal transport: With applications to data science. Found Trends Mach Learn 11(5–6):355–607

Santambrogio F (2007) Optimal channel networks, landscape function and branched transport. Interfaces and Free Boundaries 9(1):149–169

Santambrogio F (2015) Optimal transport for applied mathematicians, vol 55. Birkäuser, New York, pp 58–63

Schaub MT, Benson AR, Horn P, Lippner G, Jadbabaie A (2020) Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. SIAM Rev 62(2):353–391

Taylor D, Klimm F, Harrington HA, Kramár M, Mischaikow K, Porter MA, Mucha PJ (2015) Topological data analysis of contagion maps for examining spreading processes on networks. Nat Commun 6(1):1–11

Tero A, Kobayashi R, Nakagaki T (2007) A mathematical model for adaptive transport network in path finding by true slime mold. J Theor Biol 244(4):553–564

Tero A, Takagi S, Saigusa T, Ito K, Bebber DP, Fricker MD, Yumiki K, Kobayashi R, Nakagaki T (2010) Rules for biologically inspired adaptive network design. Science 327(5964):439–442

Villani C (2009) Optimal transport: old and new, vol 338. Springer, Berlin

Westendorf C, Gruber C, Grube M (2016) Quantitative comparison of plasmodial networks of different slime molds. In: Proceedings of the 9th EAI international conference on bio-inspired information and communications technologies (formerly BIONETICS), pp 611–612

Xia Q (2003) Optimal paths related to transport problems. Commun Contemp Math 5(02):251–279

Xia Q (2014) On landscape functions associated with transport paths. Discrete Contin Dyn Syst 34(4):1683–1700

Xia Q (2015) Motivations, ideas and applications of ramified optimal transportation. ESAIM Math Model Numer Anal 49(6):1791–1832

## Publisher's Note

www.nature.com/scientificreports

scientific reports

OPEN

# Community detection in networks by dynamical optimal transport formulation

Daniela Leite[1,3]✉, Diego Baptista[1,3], Abdullahi A. Ibrahim[1], Enrico Facca[2] & Caterina De Bacco[1]

**Detecting communities in networks is important in various domains of applications. While a variety of methods exist to perform this task, recent efforts propose Optimal Transport (OT) principles combined with the geometric notion of Ollivier–Ricci curvature to classify nodes into groups by rigorously comparing the information encoded into nodes' neighborhoods. We present an OT-based approach that exploits recent advances in OT theory to allow tuning between different transportation regimes. This allows for better control of the information shared between nodes' neighborhoods. As a result, our model can flexibly capture different types of network structures and thus increase performance accuracy in recovering communities, compared to standard OT-based formulations. We test the performance of our algorithm on both synthetic and real networks, achieving a comparable or better performance than other OT-based methods in the former case, while finding communities that better represent node metadata in real data. This pushes further our understanding of geometric approaches in their ability to capture patterns in complex networks.**

Complex networks are ubiquitous, hence modeling interactions between pairs of individuals is a relevant problem in many disciplines[1,2]. Among the variety of analyses that can be performed on them, community detection[3–6] is a popular application that involves finding groups (or communities) of nodes that share similar properties. The detected communities may reveal important structural properties of the underlying system. Community detection has been used in diverse areas including, discovering potential friends on social networks[7], evaluating social networks[8], personalized recommendation of item to user[9], detecting potential terrorist activities on social platforms[10], fraud detection in finance[11], study epidemic spreading process[12] and so on.

Several algorithms have been proposed to tackle this problem which utilize different approaches, such as statistical inference[13,14], graph modularity[15], statistical physics[16], information theory[17] and multifractal topological analysis[18]. Here, instead, we adopt a recent approach connecting community detection with geometry, where communities are detected using geometric methods like the Ollivier–Ricci curvature (ORC) and we exploit a dynamical approach of optimal transport theory to calculate this efficiently and flexibly across various transportation regimes.

In Riemannian geometry, the sign of the curvature quantifies how geodesic paths converge or diverge. In networks, the ORC plays a similar role: edges with negative curvature are traffic bottlenecks, whereas positively curved ones allow mass to flow more easily along the network. Defining communities as structures that allow robust transport of information, we could cluster edges based on their curvature: those with positive curvature can be clustered together, while those with negative curvature may be seen as "bridges" connecting different communities. The idea of using Ricci curvature to find communities on networks was first proposed by Jost and Liu [19] and then further explored in subsequent works[20–23]. Our work follows a similar approach as in[22,23] to calculate the OR curvature, but generalizes it for the cases of branched[24,25] and congested[26] optimal transport problems, building from recent results[27,28]. Specifically, our algorithm allows to efficiently tune the sensitivity to detecting communities in a network, through a parameter that controls the flow of information shared between nodes. We perform a comprehensive comparison between the proposed algorithm and existing ones on synthetic and real data. Our algorithm, named ORC-Nextrout, detects communities in synthetic networks with similar or higher accuracy compared to other OT-based methods in the regime where inference is not trivial, i.e. the inference problem is neither too easy nor too difficult to solve, and thus communities are only partially retrieved. This is also observed in a variety of real networks, where the ability to tune between different transportation regimes

[1]Max Planck Institute for Intelligent Systems, Cyber Valley, 72076 Tübingen, Germany. [2]Univ. Lille, Inria, CNRS, UMR 8524 - Laboratoire Paul Painlevé, 59000 Lille, France. [3]These authors contributed equally: Daniela Leite and Diego Baptista. ✉email: daniela.leite@tuebingen.mpg.de

allows finding at least one result that outperforms other methods, including approaches based on statistical inference and modularity.

**Related work.** The idea of exploring the geometrical properties of a graph, and in particular curvature, has been explored in different branches of network science, ranging from biological[29] to communication[30] networks. Intuitively, the Ricci curvature can be seen as the amount of volume through which a geodesic ball in a curved Riemannian manifold deviates to the standard ball in Euclidean spaces[31]. When defined in graphs, it indicates whether edges (those with positive values for the curvature) connect nodes inside a cluster, or if they rather bond different clusters together (those with negative values for the curvature).

Previous works[32–35] extended the idea of the OR curvature. In[32], the authors introduced the concept of "resistance curvature" for both nodes and edges. Taking inspiration from electrical circuits, this approach assigns a resistance being applied by the whole network from a current that flows between any two edges and correlates this to known concepts of OR discrete curvature. The resistance curvature provides a natural way to define the Ricci flow. In[33] the authors proposed a *dynamical* version of the OR curvature, where a continuous-time diffusion process is defined for every node, at different time scales. In this context, the dynamical perspective is used to frame probability masses at nodes in terms of diffusion processes, e.g. those deployed in random walks. In our work instead, the dynamics enters to solve efficiently the underlying optimization problem required to compute the OR curvature. Regardless of the choice of the distribution that characterizes mass on nodes, this quantity is then used to define the curvature of the edges of the graph. Previous works have typically defined the OR curvature in terms of the 1-Wasserstein distance. In contrast, we take a more general approach and explore the usage of the $\beta$-Wasserstein, where $\beta \in (0, 2]$, to account for a variety of OT problems, ranging from branched to congested transportation.

Other discrete graph curvature approaches include the Ollivier–Ricci (OR) curvature based on the Optimal Transport theory introduced by Ollivier,[36,37] and Forman–Ricci curvature introduced by Forman[38]. While the graph Laplacian-based Forman curvature is computationally fast and less geometrical, we focus on the OT-based approach due to its more geometric nature. Some applications of the Ollivier–Ricci curvature include network alignment[39] and community detection[22,23,40].

On the other hand, community detection in networks is a fundamental area of network science, with a wide range of approaches proposed for this task[3,4,41]. These include methods based on statistical mechanical models[16,17,42], probabilistic generative models[13,43–45], nonnegative matrix factorization[46], spectral methods[47,48], multifractal topological analysis[18] and modularity optimization[15,49,50]. In contrast, our work is inspired by recent OT-based methods[22,23] for community detection. These methods consider the OR curvature to sequentially identify and prune negatively curved edges from a network to identify communities. While our approach also considers OR curvature to prune edges, it controls the flow of information exchanged between nodes by means of a traffic-penalization parameter, making the edge pruning completely dynamic. This is detailed in "β-Wasserstein community detection algorithm" section.

## β-Wasserstein community detection algorithm

In this section, we describe how our approach solves the community detection problem. As previously stated, we rely on optimal transport principles to find the communities. To solve the optimal transport problem applied in our analysis we use the discrete *Dynamic Monge-Kantorovich* model (*DMK*), as proposed by Facca et al.[51,52] to solve transportation problems on networks.

We denote a weighted undirected graph as $G = (V, E, W)$, where $V$, $E$, $W$ are the set of nodes, edges, and weights, respectively. We use the information of the neighborhood of a node $i$, $\mathcal{N}(i) = \{j \in V | (i, j) \in E\}$, to decide whether a node belongs to a given community. We do this by comparing a distribution defined on $\mathcal{N}(i)$ with those defined on other nodes close to $i$. There are several choices that can be made for this. For instance, one could frame this in the context of diffusion processes on networks and relate the distribution to random walkers traveling along the network with a certain jump probability[33]. Here we follow previous work[39] and assign it as $m_i^\alpha$, where $m_i^\alpha(k) := \alpha$ if $k = i$ and $m_i^\alpha(k) := (1 - \alpha)/|\mathcal{N}(i)|$ if $k \in \mathcal{N}(i)$. Intuitively, the distribution $m$ assigns a unit of mass to $i$ and its connections: $\alpha$ controls how much weight node $i$ should have, and once this is assigned, its neighbors receive the remaining mass in an even way. We use $\alpha = 0$ in all the experiments reported in this manuscript, i.e. the mass is equally distributed on the neighbors. This corresponds to a one-step transition probability for a random walker in the context of diffusion processes.

The next step is to compare the distribution $m_i^\alpha$ of node $i$ to that of its neighbors. Consider an edge $(i, j) \in E$ and $m_j$, the distribution defined on node $j$, neighbor of $i$. We assume that if $i$ and $j$ belong to the same community, then both nodes may have several neighbors in common and, therefore, $m_i$ and $m_j$ should be similar. Note that this is valid for both assortative and disassortative community structures. In the former case, nodes are more likely to interact within the same community, while in the latter case we have the opposite, nodes are more likely to interact across communities[2,4]. When there is a consistent community pattern for all groups (e.g. all communities are assortative), this idea of comparing the distributions $m_i^\alpha$ may be appropriate to detect communities. On the contrary, it may be difficult to perform this task in networks with mixed connectivity patterns, where some communities are assortative and others are disassortative. This makes it difficult to detect communities as edges within an assortative community are shortened, likewise edges between a node in a disassortative and a node in an assortative one. This may confuse the algorithm, as both types of edges are shortened. A careful treatment of these cases is an interesting direction for future work.

To estimate the similarity between $m_i$ and $m_j$ we use OT principles. Specifically, we compute the cost of transforming one distribution into the other. This is related to the cost of moving the mass from one neighborhood to the other, and it is assumed to be the weighted shortest-path distance between nodes belonging to $\mathcal{N}(i)$ and

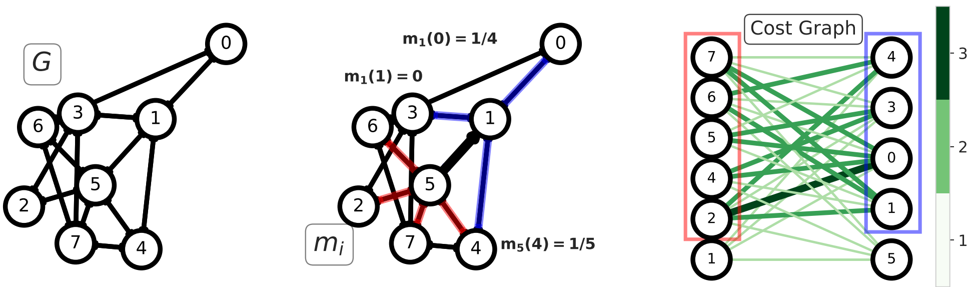


**Figure 1.** Left: an example graph $G$ where edges have unitary weights. Center: the edge (1, 5) (bold black line) is selected to define the OT problem between $m_1, m_5$; neighborhoods of nodes 1 and 5 are highlighted with blue and red edges and are used to build the corresponding distributions $m_1, m_5$. Right: The complete bipartite graph $B_{15}$ where the OT problem is defined. The color intensity of the edges represents the distance between the associated nodes on the graph $G$, as shown by the color bar. $m_1$ and $m_5$ are both defined for $\alpha = 0$, i.e. no mass is left in 1 and 5.

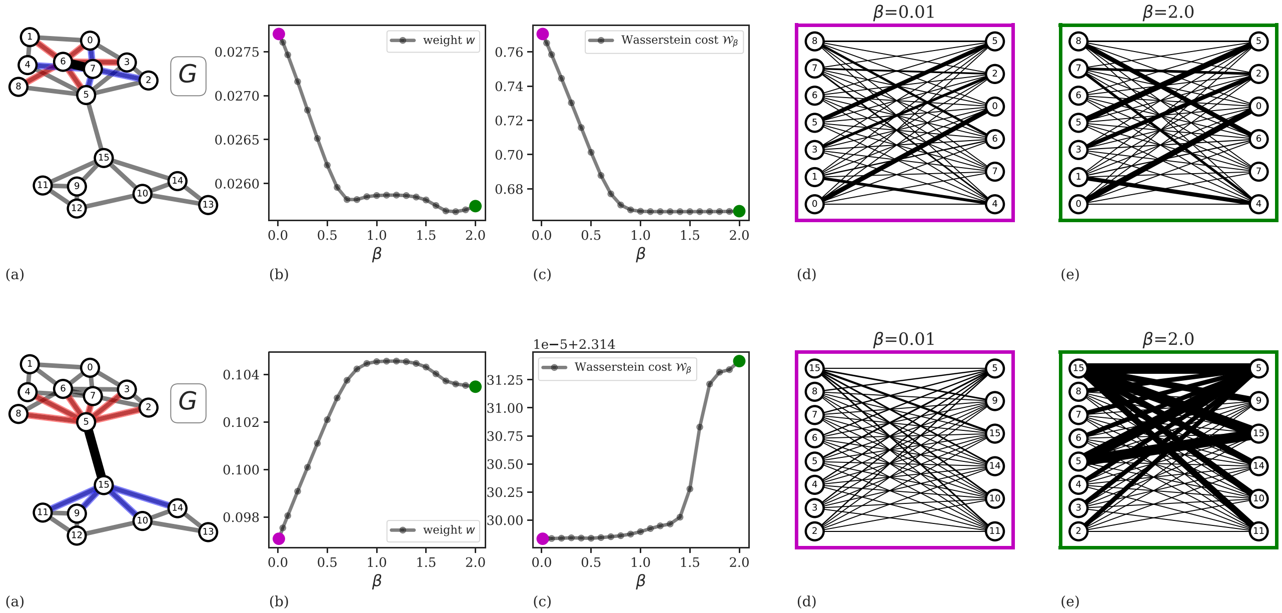


**Figure 2.** Visualization of how $\beta$ impacts intra-community and inter-community edge weights. (**a**) Examples of intra-community (top panel) and inter-community (bottom panel) structures between nodes 6 and 7, and nodes 5 and 15, respectively. (**b**) The weight of edge (6, 7) decreases when $0 < \beta < 0.6$, while for $0.5 < \beta < 2.0$ it reaches a minimum, and then slightly increases again. Similar but opposite pattern is observed for the edge (5, 15). (**c**) The $\beta$-Wasserstein cost: for intra-community edges, $\beta > 1$ consolidates traffic in the network as the Wasserstein cost stabilizes, making it minimum for the extreme value $\beta = 2$, whereas it is maximized in the case of the inter-community edge. (**d**, **e**) Example cost graphs $B_{67}$ (top) and $B_{515}$ (bottom) with fluxes solution of the OT problem (edge thickness is proportional to the amount of flux) in the regimes of small (**d**) and high (**e**) values of $\beta$.

$\mathcal{N}(j)$. A schematic representation of the algorithm can be seen in Fig. 1. The OT problem is solved in an auxiliary graph, the complete bipartite network $B_{ij} = (V_{ij}, E_{ij}, \omega_{ij})$ where $V_{ij} := (V_i, V_j) := (\mathcal{N}(i) \cup \{i\}, \mathcal{N}(j) \cup \{j\})$, $E_{ij}$ is made of all the possible edges between $V_i$ and $V_j$. The weights of the edges are given by the weighted shortest path distance $d$ between two nodes measured on the input network $G$.

The similarity between $m_i$ and $m_j$ is the Wasserstein cost $\mathcal{W}(m_i, m_j, \omega_{ij})$ of the solution of the transportation problem. In its standard version, this number is the inner product between the solution $Q$, a vector of flows defined on edges, and the cost $\omega_{ij}$. In our case, since the DMK model allows to control the flow of information through a hyperparameter $\beta \in (0, 2]$, we define the *$\beta$-Wasserstein cost*, $\mathcal{W}_\beta(m_i, m_j, \omega_{ij})$, as the inner product of the solution $Q = Q(\beta)$ of the DMK model and the cost $\omega_{ij}$. For $\beta = 1$ we compute the 1−Wasserstein distance between $m_i$ and $m_j$, while for $\beta \neq 1$ the influence of $\beta$ in the solution of the transportation problem can be seen in Fig. 2. When $\beta < 1$, more edges of $B$ tend to be used to transport the mass, thus we observe congested transportation[26]. When $\beta > 1$ fewer edges are used, hence we observe branched transportation, and the *β-Wasserstein cost* coincides with a branched transport distance[25,53]. The idea of tuning $\beta$ to interpolate between various transportation regimes has been used in several works and engineering applications[27,54–59].

Calculating the Wasserstein cost is necessary to determine our main quantity of interest, the discrete Ollivier–Ricci curvature, defined as

$$\kappa_\beta(i,j) := 1 - \frac{\mathcal{W}_\beta(m_i, m_j, \omega_{ij})}{d_{ij}} \quad , \tag{1}$$

where $d_{ij}$ is the weighted shortest path distance between $i$ and $j$ as measured in $G$. Intuitively, if $i$ and $j$ are in the same communities, several $k \in V_i$ and $\ell \in V_j$ will be also directly connected. Thus, the Wasserstein distance between $m_i$ and $m_j$ will be shorter than $d_{ij}$, yielding a positive $\kappa_\beta(i,j)$. Instead, when $i$ and $j$ are in different communities, their respective neighbors will be unlikely connected, hence $d_{ij} < \mathcal{W}_\beta(m_i, m_j, \omega_{ij})$, yielding a negative $\kappa_\beta(i,j)$.

The Ricci flow algorithm on a network is defined by iteratively updating the weights of the graph $G$[22,23]. These are updated by combining the curvature and shortest path distance information[36]. We redefine these updates using our proposal for the Ollivier–Ricci curvature:

$$w_{ij}^{(t+1)} := d_{ij}^{(t)} - \kappa_\beta^{(t)}(i,j) \cdot d_{ij}^{(t)}, \tag{2}$$

where $w_{ij}^{(t+1)}$ is the weight of edge $(i,j)$ at time $t$, $w_{ij}^{(0)} = d_{ij}^{(0)}$, and $d_{ij}^{(t)}$ is the shortest path distance between nodes $i$ and $j$ at iteration $t$. At every time step $t$, the weights are normalized by their total sum.

The algorithm ORC-Nextrout dynamically changes the weights of the graph $G$ to isolate communities: intra-community edges will be shortened, while inter-community ones will be enlarged. These changes are reached after a different number of iterations of the whole routine depending on the input data. To find the communities, we apply a *network surgery* criterion on the edges based on the stabilization of the modularity of the network, as proposed by Ni et al.[22]. Notice that our algorithm does not need prior information about the number of communities: edges will be enlarged or shortened depending on the optimal transport principles, agnostic to community labeling. The computational complexity of the algorithm is dominated by that of solving the DMK model, which takes $O(|E|^{2.36})$ (estimated numerically) and by computing weighted shortest path distances $d_{ij}$, which costs $O(|V|^2 \log |V| + |V||E|)$[60]. A pseudo-code of the implementation is shown in Algorithm 1.

**Algorithm 1** ORC-Nextrout

```
Input: $G = (V, E, W)$, traffic rate $\beta$, $MaxIterNum \in \mathbb{N}$
Output: updated $W$
Initialize: edge weights $\mathbf{w}^0 = W$; neighborhood distributions $\mathbf{m}$
for $t \in range(MaxIterNum)$ do
    Compute all-pair-shortest-path matrix $d^t$
    for $e = (i,j) \in E$ do
        Build $B_{ij}$
        Get $Q(\beta) \in \mathbb{R}^{|E_{ij}|}$, $Q(\beta) = DMK(B_{ij}, m_i, m_j, \beta, d^t)$
        Compute $\kappa_\beta(e)$ using $d^t(e)$ and $Q(\beta)$
        Compute $\mathbf{w}_\beta(e)$ using $\kappa_\beta(e)$ and $d^t(e)$
    end for
    Update $\mathbf{w}^t = \mathbf{w}_\beta$
end for
```

## Results on community detection problems

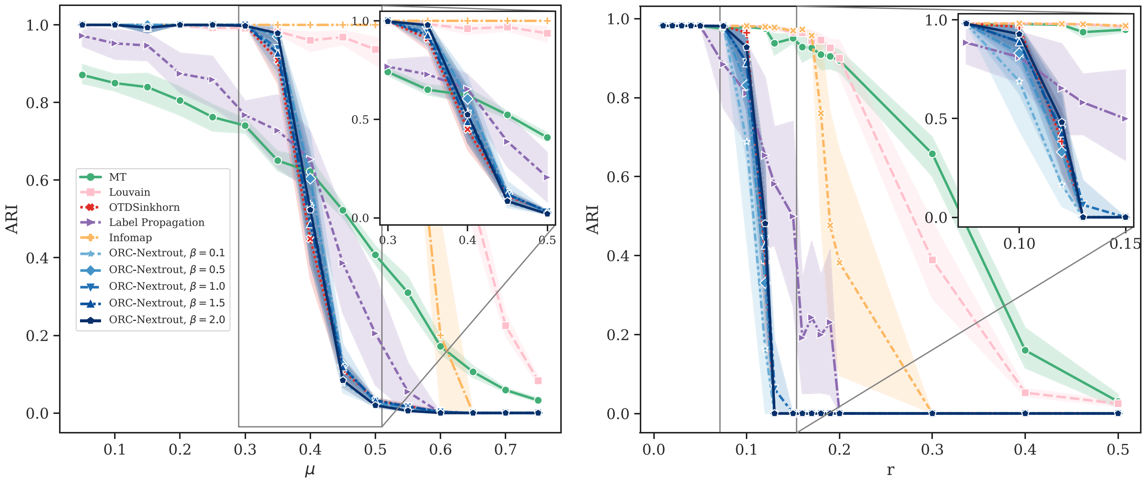
### Synthetic networks.

To investigate the accuracy of our model in detecting communities, we consider synthetic networks generated using the *Lancichinetti–Fortunato–Radicchi* (LFR) benchmark[61] and the *Stochastic Block Model* (SBM)[62]. Both models provide community labels used as *ground-truth* information during the classification tasks.

*Lancichinetti–Fortunato–Radicchi benchmark:* this benchmark generates undirected unweighted networks $G$ with disjoint communities. It samples node degrees and community sizes from power law distributions, see Fig. 4 for an example. One of its advantages is that it generates networks with heterogeneous distributions of degrees and community sizes. The main parameters in input are the number of nodes $N$, two exponents $\tau_1$ and $\tau_2$ for the power law distributions of the node degree and community size respectively, the expected degree $d$ of the nodes, the maximum number of communities on the network $K_{max}$ and a fraction $\mu$ of inter-community edges incident to each node. To test the performance of our algorithm, we use the set of LFR networks used and provided by the authors of[22]. We set $\tau_1 = 2, \tau_2 = 1, d = 20, K_{max} = 50$ and $\mu \in [0.05, 0.75]$.

*Stochastic Block Model:* this model probabilistically generates networks with non-overlapping communities. One specifies the number of nodes $N$ and the number of communities $K$, together with the expected degree $d$ of a node and a ratio $r \in [0,1]$. Networks are generated by connecting nodes with a probability $r * p_{intra}$ if they belong to different communities; $p_{intra}$ if they are part of the same community, where $p_{intra} = d \times K/N$. Notice that the smaller the ratio $r$ is, the fewer inter-community connections would exist, which leads to networks with a more distinct community structure.

We set $N = 500, K = 3, d = 15$ and $r \in [0.01, 0.5]$ and generate 10 random networks per value of $r$.



(**a**) LFR

(**b**) SBM

**Figure 3.** Results on LFR and SBM synthetic data. Performance in detecting ground-truth communities is measured by the ARI score. Markers and shadows are the averages and standard deviations over 10 network realisations with the same value of the parameter used in generation. Markers' shape denote different algorithms. (**a**) LFR graph with $N = 500$ nodes and different values of $K$ ranging from (17, 22). (**b**) SBM with $N = 500$ nodes, $K = 3$ communities and average degree $d = 15$. The parameter $r$ is the ratio of inter-community with intra-community edges. The inset on each plot zooms in the central parts of the plots.

**Results.** To evaluate the performance of our method in recovering the communities, we use the *Adjusted Rand Index* (ARI)[63]. ARI compares the community partition obtained with the *ground truth* clustering. It takes values ranging from 0 to 1, where ARI = 0 is equivalent to random community assignment, and ARI = 1 denotes perfect matching with the ground truth communities, hence the higher this value, the better the recovery of communities. A more detailed presentation of this metric is given in the Supplementary Information.

We test our algorithm for different types of information spreading in our OT-based model, as controlled by the parameter $\beta$, using the software developed in[64] (available at https://gitlab.com/enrico_facca/dmk_solver). We used $\beta = 1$, i.e., standard Wasserstein distance; $\beta \in \{0.1, 0.5\}$ for congested transportation, enforcing broad spreading across neighbors; and $\beta \in \{1.5, 2\}$ to favor branching schemes, where fewer edges are used to decide which community a node should belong to. For OT-based algorithms where we update the weights in Eq. (2) for 15 times (*MaxIterNum* $= 15$ in Algorithm 1). Since in some cases the ARI score does not consistently increase with the number of iterations, we show results only for the iteration that maximizes the score.

The results in Fig. 3 show the performance on both LFR and SBM benchmarks with OT-based methods, our method for various $\beta$ and one based on the Sinkhorn algorithms (OTDSinkhorn)[65,66]. Our main goal is to assess the impact of tuning between different transportation regimes (as done by $\beta$) in terms of community detection via OT principles. Nevertheless, to better contextualize the performance of OT-based algorithms in the wide spectrum of community detection methods, we also include comparisons with algorithms that are not OT-based. Namely, we consider a probabilistic model with latent variables (MT)[13], two modularity-based algorithms, Label Propagation[14] and Louvain[50], and with a flow-based algorithm, Infomap[17]. Our algorithm outperforms OTDSinkhorn for various values of $\beta$ in an intermediate regime where OT-based inference is not trivial, i.e. detecting communities is neither too easy nor too difficult. This occurs in both the LFR and SBM benchmark, as shown in Fig. 3. For lower and higher values of the parameters, performance is similar and close to the two extremes of ARI $= 0$ and 1. OT-based methods have a similar sharp decay in performance from the regime where inference is easy to the more difficult one, as also observed in[22]. The other community detection methods have smoother decay, but with lower performance in the regime where OT-based approaches strive, except for Label Propagation and MT, which are more robust in this sense. In the intermediate regime where inference is not trivial (i.e. along the sharp decay of OT-based methods), we observe that different values of $\beta$ give higher performance than OTDSinkhorn in most cases. For SBM the highest performance is consistently achieved for high $\beta = 2$, while for LFR the best $\beta$ varies with $\mu$. A qualitative example where ORC-Nextrout is performing better than OTDSinkhorn, in an instance of LFR of this intermediate regime, is shown in Fig. 4. Note that in this case, ORC-Nextrout perfectly recovers the 21 communities described by the ground-truth network, whereas OTDSinkhorn merges three of the central communities into one group, therefore recovering only 19 groups.

These results suggest that practitioners may choose the $\beta$ that gives the best performance in detecting communities, e.g. the one that maximizes ARI or other relevant metrics depending on the application at hand. We show examples of this on real data below.

## Analysis of real networks.

Next, we evaluate our model on various real datasets[67] containing node metadata that can be used to assess community recovery. While failing to recover communities that align well with node metadata should not be automatically interpreted as a model's failure[68] (e.g. the inferred communities and the chosen node metadata may capture different aspect of the data), having a reference community structure to

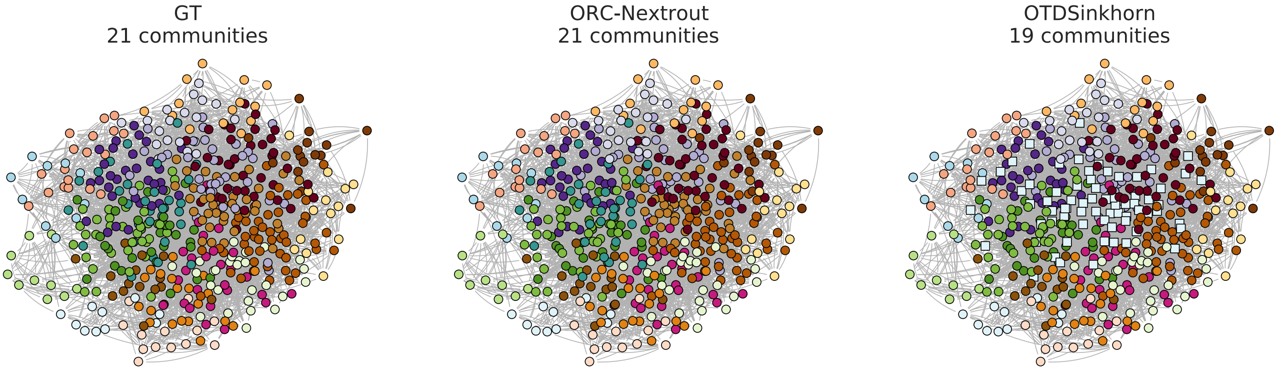


**Figure 4.** Example of community structure on a synthetic LFR network. The rightmost panel shows the ground-truth community structures to be predicted in an LFR network generated using $\mu = 0.35$. This network is one sample of the synthetic data used in Fig. 3. Square-shaped markers denote nodes that are assigned to communities different from those in ground-truth. In the middle and last panels, ORC-Nextrout with $\beta = 2$ perfectly retrieves the 21 communities, while OTDSinkhorn predicts only 19 communities with an ARI score of 0.73, wrongly assigning ground-truth dark green and light brown (square-shaped) nodes to the light blue community.

| Dataset | *N* | *E* | *K* | AvgDeg | AvgBtw | AvgClust |
|---|---|---|---|---|---|---|
| Les Miserables | 77 | 254 | 11 | 6.6 | 0.0219 | 0.5731 |
| Dolphins | 62 | 159 | 4 | 5.1 | 0.0393 | 0.2590 |
| American football | 115 | 613 | 12 | 10.7 | 0.0133 | 0.4032 |
| Political books | 105 | 441 | 3 | 8.4 | 0.0202 | 0.4875 |

**Table 1.** Real networks description. We report statistics for the real networks used in our experiments. *N* and *E* denote the number of nodes and edges, respectively. *K* is the number of communities in the ground truth data. AvgDeg, AvgBtw and AvgClust are the average degree, betweenness centrality and average clustering coefficient, respectively.

compare against allows one to inspect quantitatively difference between models. These real networks differ on structural features like number of nodes, average degree, number of communities, and other standard network properties as detailed in Table 1. Specifically, we consider (i) a network of co-appearances of characters in the novel *Les Misérables*[69] (Les Miserables). Edges are built between characters that encounter each other. (ii) A network of 62 bottlenose dolphins in a community living off Doubtful Sound, in New Zealand[70] (Dolphins). The nodes represent dolphins, and the edges indicate frequent associations between them. This network is clustered into four groups, conjectured as clustered from one population and three sub-populations based on the interactions between dolphins of different sex and ages[71]. The dolphins were observed between 1994 and 2001. (iii) A network of Division I matches of American Football during a regular season in the fall of 2000[49] (American football). Nodes represent teams, and edges are games between teams. Teams can be clustered according to their football college conference memberships. (iv) A network of books on US politics published around the 2004 presidential election and sold by an online bookseller[72] (Political books). Nodes represent the books, and the edges between books are frequent co-purchasing of books by the same buyers. Books are clustered based on their political spectrum as neural, liberal, or conservative.

OT-based algorithms outperform other community detection algorithms in detecting communities aligned with node metadata for two of the four studied datasets, as shown in Fig. 5. In particular, ORC-Nextrout has the highest accuracy performance considering the best performing $\beta$ in these cases. The impact of tuning this parameter is noticeable from these plots, as the best-performing value varies across datasets. In Les Miserables and Dolphins networks, $\beta < 1$ has better performance, while in American Football the best performing value is for $\beta > 1$. Performance is similar across OT-based methods in the Political books network. In Fig. 6 we show the communities detected by the best-performing ORC-Nextrout version together with OTDSinkhorn and Infomap in Les Miserables and Political books (see Supplementary Material for the remaining datasets). Focusing on Les Miserables, we see how ORC-Nextrout successfully detects three characters in the green communities, in particular a highly connected node in the center of the figure (in dark green). Notice that these are placed in the same community (pink or black) by OTDSinkhorn. Thus ORC-Nextrout achieves a higher ARI than OTDSinkhorn. Both OT-based approaches retrieve well communities exhibiting clustering patterns with many connections within the community. Instead, they both divide the communities with a hub and spokes structure due to the lack of common connections within the group.

The communities detected in both datasets highlight the tendency of OT-based methods to extract a larger number of communities than those observed from node metadata. Among these extra communities, some are made of a few nodes (e.g. the light-blue and violet), while others are made of one isolated node each (highlighted

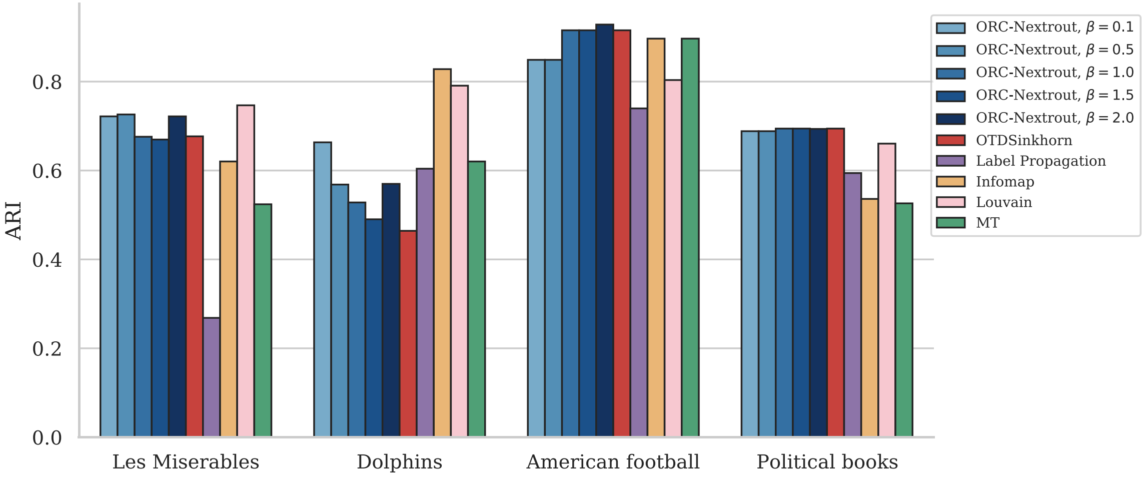


**Figure 5.** Results on real data. Performance in terms of recovering communities using metadata information is calculated in terms of the ARI score. ORC-Nextrout shows competing results against all methods with different optimal $\beta$ across datasets.

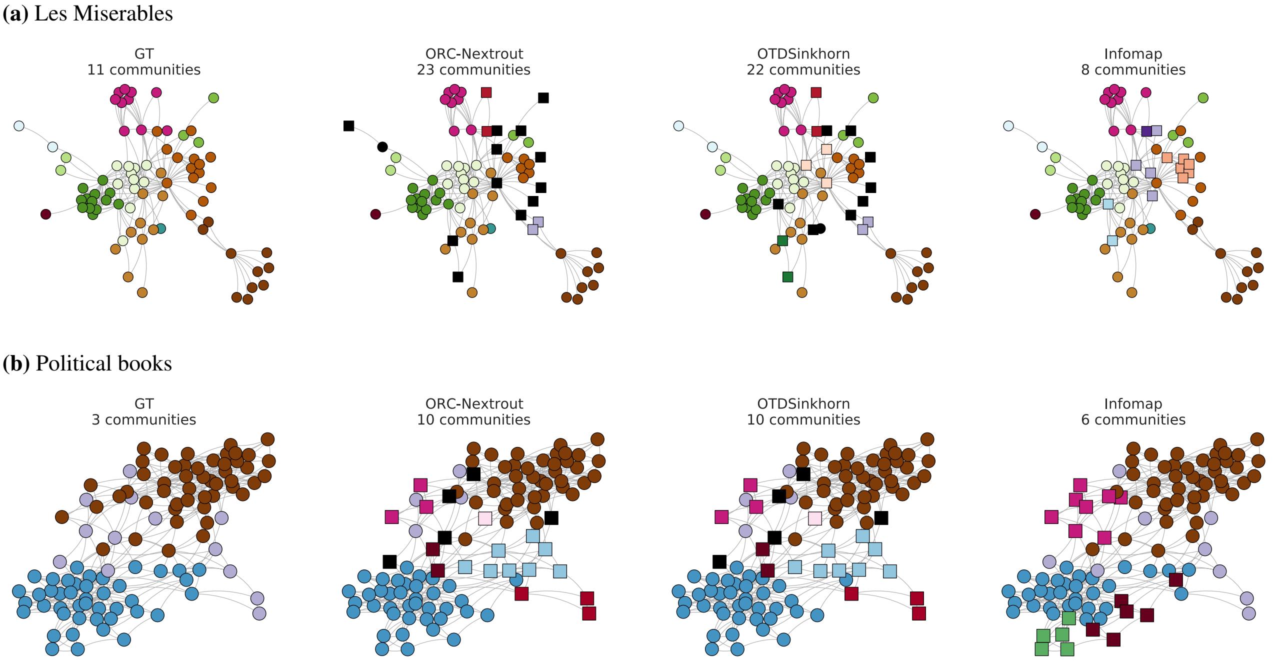


**Figure 6.** Communities in real networks. We show the communities inferred for Les Miserables (**a**) and Political books (**b**) by ORC-Nextrout ($\beta = 0.5, 0.1$ for top and bottom rows respectively), OTDSinkhorn and Infomap and compare against those extracted using node attributes (GT). The visualization layout is given by the *Fruchterman-Reingold force-directed* algorithm[73], therefore, groups of well-connected nodes are located close to each other. Dark nodes represent individual nodes who are assigned to isolated communities by OT-based methods. Square-shaped markers denote nodes assigned to communities different from those obtained from node metadata.

in black). This is related to the fact that OT-based methods perform particularly well for networks with internally densely connected community structures, but may be weaker for community structures that are sparsely connected[23]. One could potentially assign these nodes to larger communities, for instance, by preferential attachment as done in[23], thus in practice reducing the number of communities. Devising a principled method or criterion to do this automatically is an interesting topic for future work. This tendency is further corroborated by the fact that OT-based algorithms recover robustly the two communities that are mostly assortative (blue and brown in the figure) in the Political books network, while they struggle to recover the disassortative community depicted in the center (violet). This community has several connections with nodes in the other two communities and has been separated into smaller groups by OT-based approaches, as described above. This also highlights the

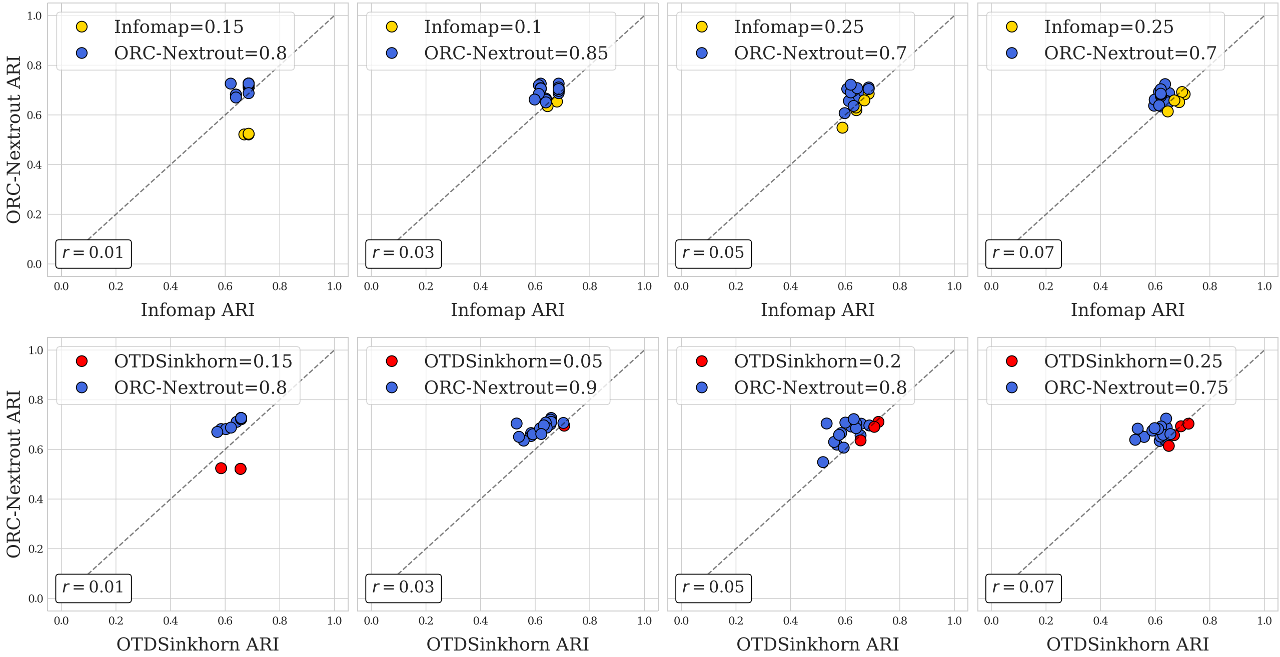


**Figure 7.** Removing intra-community edges test on Les Miserables data. Markers correspond to 20 instances of semi-synthetic networks generated from real data. Their $(x, y)$ coordinates are ARI scores of the indicated method on the axes. Colors are given by the best performing algorithm, e.g. if $x > y$, the color of the associated method to $x$ is chosen. The legend shows the percentage of times that the corresponding method outperforms the other. The parameter $r$ describes the proportion of entries for the adjacency matrix $A$ that have been changed. This increases from left to right.

need for methods that are robust against situations where mixed connectivity patterns arise, i.e. a combination of assortative and disassortative communities coexisting in a network.

**Two tests on semi-synthetic networks.** To further investigate the different performance gaps between the various approaches, we expand the comparison between the OT-based methods and Infomap on two semi-synthetic scenarios generated from Les-Miserables (Fig. 6a) and Dolphins datasets, where the largest ARI differences were observed. Specifically, we add random noise to the existing set of connections to understand if the performance gap can also be observed in more challenging scenarios. We add noise to the real data in two different ways (more details can be found in the Supplementary Material):

1. *Flipping entries:* from a given network, we generate a new one by flipping $R$ entries of its adjacency matrix $A$ uniformly at random. This means that if $A_{ij} = 1$, this is changed to $A_{ij} = 0$, and vice versa. The flipping of an entry $A_{ij}$ occurs with probability $p = 0.1$.
2. *Removing intra-community edges:* from a given network, we build a new one with the same inter-community structure but modified intra-community one by removing $R$ within-community edges uniformly at random, based on the ground truth communities. To avoid generating disconnected networks, we only sample edges that are not connected to any leaves.

Both of these procedures make inference harder, but they act differently. The first process is meant to add random noise independently of the community structure (flips are made uniformly at random), while the second aims at targeting the community structure by weakening the assortative structure. We choose $R$ to be $r \times |V|^2$ for the first test and $r \times |E|$ for the second, where we vary $r \in [0, 1]$ to study the impact of these parameters on inferring the communities as measured by the ARI score. We generate 20 samples for each of these two mechanisms built using the Les-Miserables dataset ($|V| = 77, |E| = 254$) shown in Fig. 6a.

We show the results obtained in the test of removing intra-community edges as scatter plots in Fig. 7. We use these plots to compare the algorithms on trial-by-trial community detection tasks: a point on each plot is an instance of a semi-synthetic network with $(x, y)$ coordinates being the ARI scores of ORC-Nextrout ($y$) and either OTDSinkhorn or Infomap ($x$). If $y > x$, then ORC-Nextrout outperforms the other method in this particular dataset (blue markers), and vice versa if $x < y$. We then compute the percentage of times that ORC-Nextrout outperforms the other (as indicated in the legend). We find that ORC-Nextrout outperforms both OTDSinkhorn and Infomap clearly and consistently across different values of $r$ ranging from $r = 0.01$ ($R \approx 3$) to $r = 0.07$ ($R \approx 20$). In at least 70% of the cases, ORC-Nextrout gives more accurate results than the other two algorithms. This suggests that ORC-Nextrout is more robust against perturbations of the community structure.

Similar patterns are seen in the case where edges are removed at random and in semi-synthetic networks generated from the Dolphins dataset, see Supplementary Material.

## Conclusion

Community detection on networks is a relevant and challenging open area of research. Several methods have been proposed to tackle this issue, with no "best algorithm" that fits well for every type of data. We focused here on a recent line of work that exploits principles from Optimal Transport theory combined with the geometric concept of Ollivier-Ricci curvature applied to discrete graphs. Our method is flexible in that it tunes between different transportation regimes to extract the information necessary to compute the OR curvature on edges. On synthetic data, our model is able to identify communities more robustly than other OT-based methods based on the standard Wasserstein distance in the regime where inference is not trivial. On real data, our model shows better or comparable performance in recovering community structure aligned with node metadata compared to other approaches, thanks to the ability to tune the parameter $\beta$.

A relevant advantage of OT-based methods is that the number of communities is automatically learned from data, contrarily to other approaches that need this as an input parameter. In this respect, our model has the tendency to overestimate this number, similarly to other OT-based methods. Understanding how to properly incorporate small-size communities into larger ones in a principled and automatic way is an interesting topic for future work. Similarly, it would be interesting to quantify the extent to which various $\beta$ capture different network topologies. To address this, one could, for instance, use methods to calculate the structural distance between networks[74] and correlate this against the values of the best performing $\beta$. Similarly, as our approach allows obtaining different sets of weights on edges, depending on $\beta$, it would be interesting to investigate how different values of this parameter impact network properties that are governed by the weight distribution, such as multi-fractality[75].

There are a number of directions in which this model could be extended. Nodes can be connected in more than one way, as in multilayer networks. Our model could be extended by considering a different $\beta$ for each edge type, as done in[59,76]. Similarly, real networks are often rich in additional information, e.g. attributes on nodes. It would be interesting to incorporate a priori additional information to inform community detection[43,77]. This information can potentially be used to mitigate the problem of overestimation of the number of communities, as explained above. Finally, we have focused here on the flexibility of solving various transportation problems to provide different computations of the OR curvature. Different results could also be obtained by choosing different input mass distributions on nodes' neighborhoods, as done in[33]. It would be interesting to combine these two approaches to reveal further insights of the role that the OR curvature plays in detecting communities in networks.

## Methods

### Optimal transport formulation.

Consider the probability distributions $q$ that take pairs of vertices and also satisfy the constraints $\sum_i q_{ij} = m_j, \sum_j q_{ij} = m_i$. In other words, these are the joint distributions whose marginals are $m_i$ and $m_j$. We call these distributions *transport plans* between $m_i$ and $m_j$. The Optimal Transport problem we are interested in is that of finding a transport plan $q^*$ that minimizes the quantity $\sum_{i\sim j} q_{ij} d_{ij}$, where $i \sim j$ means that the nodes $i$ and $j$ are neighbors and $d_{ij}$ is the cost of transporting mass from $i$ to $j$, e.g. the distance between these two nodes. The quantity $\mathscr{W}_\beta(m_i, m_j, d) := \sum_{i,j} q^*_{ij} d_{ij}$, defined for this optimal $q^*$, is the *Wasserstein distance* between $m_i$ and $m_j$.

### The dynamical Monge–Kantorovich model.

It was recently proved[51,52] that solutions of the optimal transport problem previously stated can be found by turning that problem into a system of differential equations. This section is dedicated to describe this dynamical formulation.

Let $G = (V, E, W)$ be a weighted graph, with $N$ the number of nodes and $E$ the number of edges in $G$. Let $\mathbf{B}$ be the *signed incidence* matrix of $G$. Let $f^+$ and $f^-$ be two $N$-dimensional discrete distributions such that $\sum_{i\in V} f_i = 0$ for $f = f^+ - f^-$; let $\mu(t) \in \mathbb{R}^E$ and $u(t) \in \mathbb{R}^N$ be two time-dependent functions defined on edges and nodes, respectively. The discrete *Dynamical Monge-Kantorovich model* can be written as:

$$f_i = \sum_e B_{ie} \frac{\mu_e(t)}{w_e} \sum_j B_{ej}\, u_j(t), \tag{3}$$

$$\mu'_e(t) = \left[ \frac{\mu_e(t)}{w_e} \left| \sum_j B_{ej}\, u_j(t) \right| \right]^\beta - \mu_e(t), \tag{4}$$

$$\mu_e(0) > 0, \tag{5}$$

where $|\cdot|$ is the absolute value element-wise. Equation (3) corresponds to Kirchhoff's law, Eq. (4) is the discrete dynamics with $\beta$ a traffic rate controlling the different routing optimization mechanisms; Eq. (5) is the initial distribution for the edge conductivities.

For $\beta = 1$ the dynamical system described by Eqs. (3)–(5) is known to reach a steady state, i.e., the updates of $\mu_e$ and $u_e$ converge to stationary functions $\mu^*$ and $u^*$ as $t$ increases. The flux function $q$ defined as $q^*_e := \mu^*_e |u^*_i - u^*_j| / w_e$ is the solution of the optimal transport problem presented in the previous section. Notice that $\mu$ and $u$ depend on the chosen traffic rate $\beta$, and thus, so does $q = q(\beta)$. Therefore, we can introduce a generalized version of the distance $\mathscr{W}$:

$$\mathscr{W}_\beta(m_i, m_j, w) := \sum_{i,j} q^*_{ij}(\beta)\ w_{ij}.$$

We then redefine the proposed Ollivier-Ricci curvature as:

$$\kappa_\beta(i,j) := 1 - \frac{\mathscr{W}_\beta(m_i, m_j, w)}{d_{ij}}.$$

**Probability distributions on neighborhoods.** ORC-Nextrout takes in input a graph and a forcing term. While the graph encapsulates the neighborhood information provided by the nodes $i$ and $j$, the forcing function is related to the distributions that one needs to transport. Analogously to what was proposed by[22], we define this graph to be the weighted *complete bipartite* $B_{ij} = (V_{ij}, E_{ij}, \omega_{ij})$. The weights in $\omega_{ij}$ change iteratively based on the curvature. Notice that a bipartite graph must satisfy $\mathscr{N}(i) \cap \mathscr{N}(j) = \varnothing$, which does not hold true if $i$ and $j$ have common neighbors (this is always the case since $i \in \mathscr{N}(j)$). Nonetheless, this condition does not have great repercussions in the solution of the optimal transport problem since the weights corresponding to these edges (of the form $(i, i)$) are equal to 0. As for the forcing function, we define it to be $f := f^+ - f^- = m_i - m_j$.

**Other methods.** To evaluate the performance of ORC-Nextrout, we compare with some of the well-established community detection algorithms including: Infomap[17], MULTITENSOR[13] (MT), discrete Ricci flow[22] (OTDSinkhorn), Label propagation[14] and Louvain[50]. We briefly describe each of these algorithms as follows;

- The *Discrete Ricci flow* (here addressed as OTDSinkhorn)[22] is an iterative node clustering algorithm that deforms edge weights as time progresses, by shrinking sparsely traveled links and stretching heavily traveled edges. These edge weights are iteratively updated based on neighborhood transportation Wasserstein costs, similarly to what is proposed in this manuscript. After a predefined number of iterations, heavily traveled links are removed from the graph. Communities are then obtained as the connected components of this modified network.
- MULTITENSOR (MT)[13] is an algorithm to find communities in multilayer networks. It is a probabilistic model with latent variables regulating community structure and runs with a complexity of $O(EK)$ with assortative structure (as we consider here), where $K$ is the number of communities. This model assumes that the nodes inside the communities can belong to multiple groups (mixed-membership). In this implementation we use their validity for single layer networks (a particular case of a multilayer network).
- *Infomap*[17] employs information theoretic approach for community detection. This method uses the map equation to attend patterns of flow on a network. This flow is simulated using random walkers' traversed paths. Based on the theoretic description of these paths, nodes with quick information flow are then clustered into the same groups. The algorithm runs in $O(E)$. In the experiments, we fix the number of random initialization of the random walkers to be equal to 10. The inferred partition is then the one minimizing the entropy.
- *Label propagation*[14] assigns each node to the same community as the majority of its neighbors. Its working principle start by initializing each node with a distinct label and converges when every node has same label as the majority of its neighboring node. The algorithm has a complexity scaling as $O(E)$.
- *Louvain*[50] is a fast algorithm used to find communities on networks by maximizing the modularity of the associated partitions. It consists of two phases. First, it assigns every node on the network into a different community. Then, it aggregates nodes and neighbors based on gains of modularity. This last step is repeated until no further improvement can be achieved.

## Data availability

The real datasets analyzed during the current study are available at http://www-personal.umich.edu/~mejn/netdata/. The synthetic data generated are available from the corresponding author upon request.

## Code availability

Open source codes and executables are available at https://github.com/Danielaleite/ORC-Nextrout. https://gitlab.com/enrico_facca/dmk_solver.

Received: 17 May 2022; Accepted: 21 September 2022
Published online: 07 October 2022

## References

1. Huang, X., Chen, D., Ren, T. & Wang, D. A survey of community detection methods in multilayer networks. *Data Min. Knowl. Disc.* **35**, 1–45 (2021).
2. Newman, M. *Networks* (Oxford University Press, Oxford, 2018).
3. Fortunato, S. Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010).
4. Fortunato, S. & Hric, D. Community detection in networks: A user guide. *Phys. Rep.* **659**, 1–44 (2016).
5. Weber, M., Jost, J. & Saucan, E. Detecting the coarse geometry of networks. in *NeurIPS 2018 workshop* (2018).
6. Samal, A. *et al.* Comparative analysis of two discretizations of Ricci curvature for complex networks. *Sci. Rep.* **8**, 1–16 (2018).

7. Zhu, J., Wang, B., Wu, B. & Zhang, W. Emotional community detection in social network. *IEICE Trans. Inf. Syst.* **100**, 2515–2525 (2017).
8. Wang, D., Li, J., Xu, K. & Wu, Y. Sentiment community detection: exploring sentiments and relationships in social networks. *Electron. Commer. Res.* **17**, 103–132 (2017).
9. Li, C. & Zhang, Y. A personalized recommendation algorithm based on large-scale real micro-blog data. *Neural Comput. Appl.* **32**, 11245–11252 (2020).
10. Waskiewicz, T. Friend of a friend influence in terrorist social networks. In *Proceedings on the international conference on artificial intelligence (ICAI)*, 1 (The Steering Committee of The World Congress in Computer Science, Computer..., 2012).
11. Pinheiro, C.A.R. Community detection to identify fraud events in telecommunications networks. in *SAS SUGI proceedings: Customer intelligence* (2012).
12. Chen, J., Zhang, H., Guan, Z.-H. & Li, T. Epidemic spreading on networks with overlapping community structure. *Physica A* **391**, 1848–1854 (2012).
13. De Bacco, C., Power, E. A., Larremore, D. B. & Moore, C. Community detection, link prediction, and layer interdependence in multilayer networks. *Phys. Rev. E* **95**, 042317 (2017).
14. Raghavan, U. N., Albert, R. & Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**, 036106 (2007).
15. Clauset, A., Newman, M. E. & Moore, C. Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111 (2004).
16. Reichardt, J. & Bornholdt, S. Statistical mechanics of community detection. *Phys. Rev. E* **74**, 016110 (2006).
17. Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**, 1118–1123 (2008).
18. Yang, R., Sala, F. & Bogdan, P. Hidden network generating rules from partially observed complex networks. *Commun. Phys.* **4**, 1–12 (2021).
19. Jost, J. & Liu, S. Ollivier's Ricci curvature, local clustering and curvature-dimension inequalities on graphs. *Discrete Comput. Geom.* **51**, 300–322 (2014).
20. Weber, M., Jost, J. & Saucan, E. Forman–Ricci flow for change detection in large dynamic data sets. *Axioms* **5**, 26 (2016).
21. Weber, M., Saucan, E. & Jost, J. Characterizing complex networks with Forman–Ricci curvature and associated geometric flows. *J. Complex Netw.* **5**, 527–550 (2017).
22. Ni, C.-C., Lin, Y.-Y., Luo, F. & Gao, J. Community detection on networks with Ricci flow. *Sci. Rep.* **9**, 1–12 (2019).
23. Sia, J., Jonckheere, E. & Bogdan, P. Ollivier–Ricci curvature-based method to community detection in complex networks. *Sci. Rep.* **9**, 1–12 (2019).
24. Santambrogio, F. Optimal channel networks, landscape function and branched transport. *Interfaces Free Bound.* **9**, 149–169 (2007).
25. Facca, E., Cardin, F. & Putti, M. Branching structures emerging from a continuous optimal transport model. *J. Comput. Phys.* **447**, 110700 (2021).
26. Brasco, L., Carlier, G. & Santambrogio, F. Congested traffic dynamics, weak flows and very degenerate elliptic equations. *J. Math. Pures Appl.* **93**, 652–671 (2010).
27. Baptista, D., Leite, D., Facca, E., Putti, M. & De Bacco, C. Network extraction by routing optimization. *Sci. Rep.* **10**, 1–13 (2020).
28. Facca, E., Karrenbauer, A., Kolev, P. & Mehlhorn, K. Convergence of the non-uniform directed Physarum model. *Theoret. Comput. Sci.* **816**, 184–194 (2020).
29. Sandhu, R. *et al.* Graph curvature for differentiating cancer networks. *Sci. Rep.* **5**, 1–13 (2015).
30. Wang, C., Jonckheere, E. & Banirazi, R. Interference constrained network control based on curvature. In *2016 American control conference (ACC)* 6036–6041 (IEEE, 2016).
31. Ni, C.-C., Lin, Y.-Y., Gao, J., Gu, X. D. & Saucan, E. Ricci curvature of the internet topology. In *2015 IEEE conference on computer communications (INFOCOM)* 2758–2766 (IEEE, 2015).
32. Devriendt, K. & Lambiotte, R. Discrete curvature on graphs from the effective resistance. *J. Phys. Complex.* (2022).
33. Gosztolai, A. & Arnaudon, A. Unfolding the multiscale structure of networks with dynamical Ollivier–Ricci curvature. *Nat. Commun.* **12**, 1–11 (2021).
34. Paulin, D. Mixing and concentration by Ricci curvature. *J. Funct. Anal.* **270**, 1623–1662 (2016).
35. Veysseire, L. Coarse Ricci curvature for continuous-time Markov processes. arXiv:1202.0420 (2012)
36. Ollivier, Y. Ricci curvature of Markov chains on metric spaces. *J. Funct. Anal.* **256**, 810–864 (2009).
37. Ollivier, Y. A survey of Ricci curvature for metric spaces and Markov chains. In *Probabilistic approach to geometry* 343–381 (Mathematical Society of Japan, 2010)
38. Forman, R. Bochner's method for cell complexes and combinatorial Ricci curvature. *Discret. Comput. Geom.* **29**, 323–374 (2003).
39. Ni, C.-C., Lin, Y.-Y., Gao, J. & Gu, X. Network alignment by discrete Ollivier–Ricci flow. In *International symposium on graph drawing and network visualization*,447–462 (Springer, 2018).
40. Ye, Z., Liu, K. S., Ma, T., Gao, J. & Chen, C. Curvature graph network. In *International conference on learning representations* (2019)
41. Lancichinetti, A. & Fortunato, S. Community detection algorithms: A comparative analysis. *Phys. Rev. E* **80**, 056117 (2009).
42. Decelle, A., Krzakala, F., Moore, C. & Zdeborová, L. Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.* **107**, 065701 (2011).
43. Contisciani, M., Power, E. A. & De Bacco, C. Community detection with node attributes in multilayer networks. *Sci. Rep.* **10**, 1–16 (2020).
44. Ball, B., Karrer, B. & Newman, M. E. Efficient and principled method for detecting communities in networks. *Phys. Rev. E* **84**, 036103 (2011).
45. Peixoto, T.P. Bayesian stochastic blockmodeling. *Adv. Netw. Clust. Blockmodeling* 289–332 (2019)
46. Yang, J. & Leskovec, J. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining* 587–596 (2013).
47. Newman, M. E. Spectral methods for community detection and graph partitioning. *Phys. Rev. E* **88**, 042822 (2013).
48. Sussman, D. L., Tang, M., Fishkind, D. E. & Priebe, C. E. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *J. Am. Stat. Assoc.* **107**, 1119–1128 (2012).
49. Girvan, M. & Newman, M. E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**, 7821–7826 (2002).
50. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).
51. Facca, E., Cardin, F. & Putti, M. Towards a stationary Monge–Kantorovich dynamics: The physarum polycephalum experience. *SIAM J. Appl. Math.* **78**, 651–676 (2018).
52. Facca, E., Daneri, S., Cardin, F. & Putti, M. Numerical solution of Monge–Kantorovich equations via a dynamic formulation. *J. Sci. Comput.* **82**, 1–26 (2020).
53. Xia, Q. Optimal paths related to transport problems. *Commun. Contemp. Math.* **5**, 251–279 (2003).
54. Baptista, D. & De Bacco, C. Principled network extraction from images. *R. Soc. Open Sci.* **8**, 210025 (2021).
55. Baptista, D. & De Bacco, C. Convergence properties of optimal transport-based temporal networks. In *International conference on complex networks and their applications* 578–592 (Springer, 2021)
56. Lonardi, A., Facca, E., Putti, M. & De Bacco, C. Designing optimal networks for multicommodity transport problem. *Phys. Rev. Res.* **3**, 043010 (2021).

57. Lonardi, A., Putti, M. & De Bacco, C. Multicommodity routing optimization for engineering networks. *Sci. Rep.* **12**, 1–11 (2022).
58. Lonardi, A., Facca, E., Putti, M. & De Bacco, C. Infrastructure adaptation and emergence of loops in network routing with time-dependent loads. arXiv:2112.10620 (2021)
59. Ibrahim, A. A., Lonardi, A. & Bacco, C. D. Optimal transport in multilayer networks for traffic flow optimization. *Algorithms* **14**, 189 (2021).
60. Fredman, M. L. & Tarjan, R. E. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM (JACM)* **34**, 596–615 (1987).
61. Lancichinetti, A., Fortunato, S. & Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**, 046110 (2008).
62. Holland, P. W., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Netw.* **5**, 109–137 (1983).
63. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193–218 (1985).
64. Facca, E. & Benzi, M. Fast iterative solution of the optimal transport problem on graphs. *SIAM J. Sci. Comput.* **43**, A2295–A2319 (2021).
65. Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. *Adv. Neural Inf. Process. Syst.* **26** (2013)
66. Flamary, R. *et al.* Pot: Python optimal transport. *J. Mach. Learn. Res.* **22**, 1–8 (2021).
67. Network data. http://www-personal.umich.edu/~mejn/netdata/
68. Peel, L., Larremore, D. B. & Clauset, A. The ground truth about metadata and community detection in networks. *Sci. Adv.* **3**, e1602548 (2017).
69. Knuth, D. E. *The Stanford GraphBase: A platform for combinatorial computing* Vol. 1 (AcM Press, New York, 1993).
70. Lusseau, D. *et al.* The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **54**, 396–405 (2003).
71. Lusseau, D. & Newman, M. E. Identifying the role that animals play in their social networks. *Proc R Soc Lond Ser B Biol Sci* **271**, S477–S481 (2004).
72. Books about us politics dataset. http://www.orgnet.com/.
73. Fruchterman, T. M. & Reingold, E. M. Graph drawing by force-directed placement. *Softw. Pract. Exp.* **21**, 1129–1164 (1991).
74. Xiao, X., Chen, H. & Bogdan, P. Deciphering the generating rules and functionalities of complex networks. *Sci. Rep.* **11**, 1–15 (2021).
75. Xue, Y. & Bogdan, P. Reliable multi-fractal characterization of weighted complex networks: Algorithms and implications. *Sci. Rep.* **7**, 1–22 (2017).
76. Ibrahim, A. A., Leite, D. & De Bacco, C. Sustainable optimal transport in multilayer networks. *Phys. Rev. E* **105**, 064302 (2022).
77. Newman, M. E. & Clauset, A. Structure and inference in annotated networks. *Nat. Commun.* **7**, 11863 (2016).

## Acknowledgements

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Daniela Leite, Diego Baptista and Abdullahi A. Ibrahim.

## Author contributions

All authors contributed to developing the models, conceived the experiments, analyzing the results and reviewing the manuscript. D.L., D.B. and A.A.I. conducted the experiments.

## Funding

Open Access funding enabled and organized by Projekt DEAL.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-20986-y.

**Correspondence** and requests for materials should be addressed to D.L.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

ROYAL SOCIETY
OPEN SCIENCE

royalsocietypublishing.org/journal/rsos

# Principled network extraction from images

Diego Baptista and Caterina De Bacco

Max Planck Institute for Intelligent Systems, Cyber Valley, Tuebingen 72076, Germany

CDB, 0000-0002-8634-0211

Research

**Cite this article:** Baptista D, De Bacco C. 2021 Principled network extraction from images. *R. Soc. Open Sci.* **8**: 210025.
https://doi.org/10.1098/rsos.210025

Received: 11 January 2021
Accepted: 6 July 2021

**Subject Category:**
Computer science and artificial intelligence

**Subject Areas:**
image processing/pattern recognition/biomathematics

**Keywords:**
networks, routing optimization, optimal transport

**Author for correspondence:**
Caterina De Bacco
e-mail: caterina.debacco@tuebingen.mpg.de

Electronic supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.5514773.

THE ROYAL SOCIETY
PUBLISHING

Images of natural systems may represent patterns of network-like structure, which could reveal important information about the topological properties of the underlying subject. However, the image itself does not automatically provide a formal definition of a network in terms of sets of nodes and edges. Instead, this information should be suitably extracted from the raw image data. Motivated by this, we present a principled model to extract network topologies from images that is scalable and efficient. We map this goal into solving a routing optimization problem where the solution is a network that minimizes an energy function which can be interpreted in terms of an operational and infrastructural cost. Our method relies on recent results from optimal transport theory and is a principled alternative to standard image-processing techniques that are based on heuristics. We test our model on real images of the retinal vascular system, slime mould and river networks and compare with routines combining image-processing techniques. Results are tested in terms of a similarity measure related to the amount of information preserved in the extraction. We find that our model finds networks from retina vascular network images that are more similar to hand-labelled ones, while also giving high performance in extracting networks from images of rivers and slime mould for which there is no ground truth available. While there is no unique method that fits all the images the best, our approach performs consistently across datasets, its algorithmic implementation is efficient and can be fully automatized to be run on several datasets with little supervision.

## 1. Introduction

Extracting network topologies from images is a relevant problem in applications where the subject of the image has a network-like structure. For example, satellite images of rivers [1], neuronal networks [2,3], blood or vein networks [4–6], mitochondrial networks [7,8] or road networks [9–12]. Assuming this could be done automatically and quantitatively, practitioners would then be able to apply the mathematical study of networks to make quantitative analyses about the topological properties of the system at study. In practice, given a raw image, for instance, a satellite image of a river embedded in a landscape, extracting a network requires identifying a set of nodes and a set of edges connecting them. While it might be relatively easy to perform this identification qualitatively, the challenge here is performing this extraction automatically, thus avoiding tedious manual extraction or specific domain knowledge and *ad hoc* tools. At the same time, this task should be scalable with system size and number of images as high-quality images are increasingly available and for larger systems. In addition, a qualitative intuition of the possible existence of a network behind an image is not enough to ensure that no degree of subjectivity is introduced owing to the observer's eye. For instance, two different observers might both perceive the presence of a network-like structure but distinguish two different sets of nodes, and thus two different networks behind the same image. Another challenge is indeed that of performing this extraction in a principled way so that the number of arbitrary choices in defining what the network is should be limited, if not completely absent.

Here, we present a method that addresses these issues by considering the framework of optimal transport. Specifically, inspired by a recently developed model to extract network topologies from solutions of routing optimization problems [13], we adapt this formalism to our specific and different setting. We start from a raw image as input and propose a model that outputs a network representing the topological structure contained in the image. The novelty of this method is that its theoretical underpinning relies on a principled optimization framework. In fact, a proper energy function is efficiently minimized using numerical methods, which results in an output network topology. This implies, in particular, that network extraction may not depend on the observer's eye, but rather can be automatically done by solving this optimization problem.

We study our model on real images from different fields, we focus in particular on ecology and biology and compare results with an algorithm that relies on standard image processing techniques, highlighting the main differences resulting from these two different approaches. In particular, our model allows for an automatic and principled performance of two tasks: filtering network redundancies and selection of edge weights. These are usually challenging tasks for image processing schemes, as they rely on some pre-defined parameter setting in input, while we obtain both directly in output with our model.

Many solutions for the problem of automatic network extraction from images have been proposed in computer vision, mainly relying on image-processing techniques [6,10–12,14–20], for instance segmentation [21–23], or junction-point processes [24]. The idea is to measure variation of intensity contrast in the image's pixels to highlight curve-like structures. Within this context, NEFI [25] is a flexible toolbox based on a combination of standard image-processing routines. A different approach, closer to the one considered in this work, is that of adopting some sort of optimization framework. For instance, Breuer & Nikoloski [26] considered an optimization problem where the goal is to minimize the total roughness of a path (a measure depending on the difference of weights in adjacent edges), in order to decompose a filamentous network into individual filaments. However, these usually rely on domain-specific optimization set-ups that cannot be easily transferred across domains. Another example is the ant-colony optimization scheme used to extract blood vessels from images of retinas [27]. They all suffer from the nondeterministic polynomial time-hardness of the problem, typical of routing optimization settings. Thus this type of approach relies on approximation techniques. Finally, another approach is that of biologically inspired mathematical models like the one of Tero *et al.* [28,29] that consider dynamical systems of equations that emulate network adaptability, like that observed for the *Physarum polycephalum* slime mould. Our model is also inspired by the feedback mechanism of a slime mould, which adapts the conductivity of the network edges to respond to differences in fluxes.

Our method relies on the formalism of optimal transport theory used to extract networks from solutions of routing optimization problems proposed in [13] and referred to as NextRout. This is made of three subsequent steps, but here we need only the last two, namely the pre-extraction and the filtering steps. While we refer to that work for all the mathematical details, here we describe the main principles behind this method and adapt it to images. The idea is inspired by the behaviour of the *P. polycephalum* slime mould. The body of this organism forms a network structure that flexibly adapts to the surrounding environment and the distribution of food sources displaced in it. This network grows with a feedback mechanism between two physical quantities: the conductivities of network edges and the flow passing through them, through dynamics that is described by a set of equations (sometimes referred to as 'adaptation equations'). In practice, the problem starts by assigning food sources in space and spreading the slime mould uniformly to cover the whole space. The dynamics regulates how the slime mould changes its body shape in time to reach the

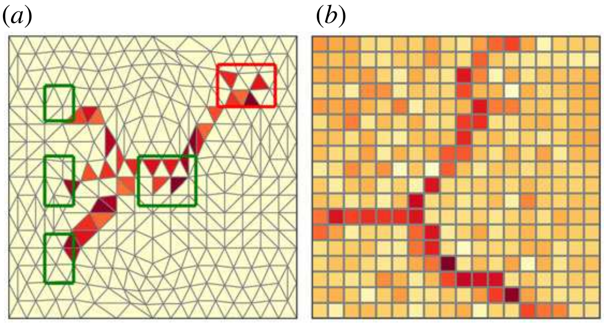


**Figure 1.** Analogy between optimizing trajectories and images. (*a*) The grid structure covering a continuous two-dimensional space and the optimal flux obtained by NextRout for a specific routing problem where sources and sinks are inside the green and red rectangles, respectively. Red tones of increasing darkness denote higher fluxes on the corresponding grid triangles. (*b*) The pixel grid and colours of a reference network-like image.

food in an efficient way. The stationary solution of this dynamical system is a set of conductivities and flows on edges that describe the optimal network topology covered by the mould. When the underlying space is continuous, like a squared patch in two dimensions, these solutions are functions defined on $(x, y)$ coordinates on this space. These are not immediately associated with a network meant as a set of nodes and a set of edges connecting them. However, Baptista *et al.* [13] propose principled rules to automatically extract network topologies from these solutions in continuous space. While the main focus of that work was to extract network topologies from this particular type of input (functions defined in a continuous domain, e.g. the space where food sources are located), they hinted at the possibility of adapting this formalism to discrete spaces, like images made of pixels. Here, we expand on this insight, and adapt this principled network extraction to inputs that are images. In particular, we propose an algorithm to effectively tackle two problems that are relevant for images and that were only briefly discussed for general applications in [13]: how to select source and sinks and how to obtain loopy structures.

## 2. Image2net: the method

The key idea is to treat the images as a particular discretization of a two-dimensional space by means of the pixels and treat the red-green-blue (RGB) colour values on them as conductivities. With this set-up, we can frame the problem as if there was an imaginary flow of colours. This starts by covering the whole discrete domain of the image uniformly and then flows through the pixels until it consolidates to a certain subset of them. The observed image corresponds to a network-like shape. Figure 1 illustrates the analogy between the solutions of NextRout in continuous space and an image of a network-like structure in discrete space.

As we mentioned before, this is not yet formally a network, as we do not have a rigorous definition of what constitutes a node and how nodes are connected. However, thanks to the analogy proposed here, we can use the rules introduced in [13] for the continuous case and adapt them to images. Specifically, we consider the pixels' centre of mass as nodes and draw edges between them depending on their pixels' locations and values, so that two nearby pixels are connected whenever the colour has a high enough intensity and their pixels are neighbours. We say that two pixels are nearby if they have a vertex or an edge in common (this corresponds to the pre-extraction scheme I, as explained in [13]). The result is a *pre-extracted network* [13] that we denote with $G^{\mathrm{pe}}$. We denote with $V$ and $E$ the set of nodes and edges, respectively. The network is mathematically encoded by a signed incidence matrix which has entries $B_{ie} = \pm 1$ if the edge $e$ has node $i$ as start/endpoint, 0 otherwise. The sign is important to define the orientation of the flow passing through an edge.

This temporary network might contain redundancies like dangling nodes or redundant edges, see figure 2 for an example. Standard image processing techniques address this problem with pruning routines, e.g. by pruning away edges or branches shorter than a certain length. However, pruning has to be handled with great care, as small redundancies could be a major source of information or they could be completely irrelevant, depending on the network at hand. Usually, pruning is tuned by the user, thus creating potential for subjective bias in extracting the network. Instead, our model relies on a principled method for filtering such redundancies, which exploits a dynamics similar to that of the original problem but adapted to a discrete space like that of the network $G^{\mathrm{pe}}$. However, to apply the

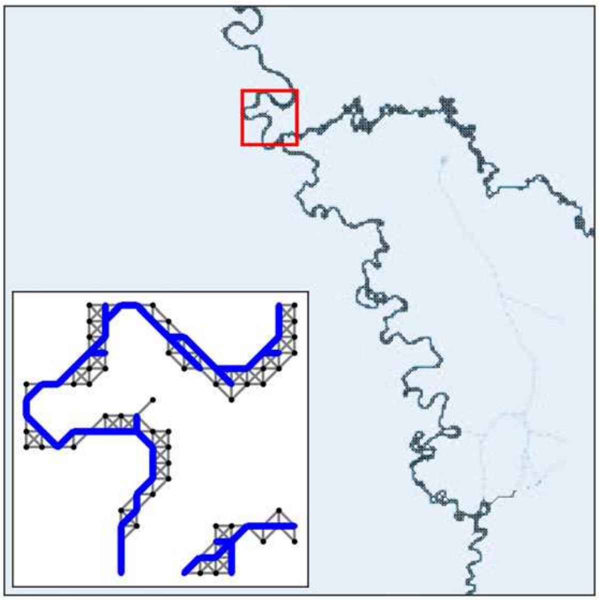


**Figure 2.** $G^{\text{pe}}$ taken from a river image. The subplot on the bottom left corner shows a section of $G^{\text{pe}}$ (in black), highlighted in red in the main plot, together with filtered graph $G^{f}$ (in blue).

filter, one must specify a set of terminals, sources and sinks, as input to the discrete dynamics. Continuing with our analogy, we need to locate the pixels where we imagine that colour mass is being injected and extracted. These are the sources and sinks that drive the dynamics to consolidate the flux of colours on the network-like structure observed in the input image.

## 2.1. Dynamics

Assume for a moment that we knew this set of terminal pixels and denote with $f_i$ the amount of colour mass that enters or exits the image in node $i$ (the centre of mass of pixel $i$). Note that to preserve the mass, we have $\sum_i f_i = 0$. Here we describe in more detail the dynamical rules that regulate how colours spread along the pixels in an optimal way. To describe the flow of colours, we consider the conductivity $\mu_e$ on an edge $e \in E$ and the potential $u_i$ on a node $i \in V$. The conductivities can be interpreted as proportional to the size of the diameter of an edge, and the potentials as pressures on nodes. Together, these two quantities determine the flow $F_e$ of colour passing through an edge $e = (v_1, v_2)$ in the network:

$$F_e = \frac{\mu_e}{\ell_e}(u_{v_1} - u_{v_2}) = \frac{\mu_e}{\ell_e}\left|\sum_{j\in V} B_{ej}\, u_j\right|. \tag{2.1}$$

The quantity $\ell_e$ denotes the Euclidean length of an edge using the centre of mass' coordinates. Although $\ell_e$ is either 1 or $\sqrt{2}$ for this particular choice of nodes, in the equations we keep $\ell_e$ for more general cases where the topology may not be as regular.

In turns, the flow influences the conductivities and potentials, through a feedback mechanism described by the following set of equations:

$$f_i = \sum_{e\in E} B_{ie} F_e, \tag{2.2}$$

$$\mu_e'(t) = \left[\frac{\mu_e(t)}{\ell_e}\left|\sum_{j\in V} B_{ej}\, u_j(t)\right|\right]^{\beta} - \mu_e(t) \tag{2.3}$$

and

$$\mu_e(0) > 0, \tag{2.4}$$

where $|\cdot|$ is the absolute value and $\beta$ is a parameter that determines the optimization mechanism. Equation (2.2) is Kirchhoff's law; equation (2.3) is the discrete dynamics describing the feedback mechanism between conductivity and flow: when the flow of colours is high on an edge $e$, the conductivity increases, and vice versa when the flow is low the conductivity decreases; equation (2.4) is the initial condition. The stationary solution of this dynamical system can be mapped to the solutions of an optimization problem where the cost function can be interpreted as a network transportation cost [13]:

$$\mathcal{L}_\beta(\mu(t)) = \frac{1}{2}\sum_e \mu_e(t)\left(\frac{1}{\ell_e}\sum_j B_{ej}u_j(\mu(t))\right)^2 \ell_e + \frac{\beta}{2}\sum_e \frac{\mu_e(t)^{(2-\beta)/\beta}}{2-\beta}\,\ell_e, \tag{2.5}$$

where $\mu(t) = \{\mu_e(t)\}_e$, and the first term is the network operational cost and the second is the cost to build the network. The values of $\mu_e$ at convergence can be used not only to determine the set of edges in the extracted network but also its weights, which can be interpreted as proportional to the diameter of the edge on the image. This is one of the advantages of our model, as estimating the diameter of edges extracted from an image is an open problem when using image-processing techniques. We get this automatically with the optimal conductivities.

The dynamics works as a filter, i.e. removes redundancies, for $\beta \geq 1$. In this work, we fix $\beta = 1.5$ as it gives good performance consistently across the datasets studied here. The output result is a tree, i.e. it does not contain loops and is the optimal one in terms of minimizing the transportation cost of equation (2.5). In our experiments, we use the numerical solver proposed in [13] to extract the stationary solutions of the system of equations (2.2)–(2.4).

## 2.2. Selecting terminal pixels

Having introduced how the dynamics of the colours works, we now tackle the problem of selecting the terminal pixels where to inject or extract imaginary colour mass. This choice is crucial as it determines the final extracted networks. In the original problem of [13], this was not an issue because the set of terminals could be selected from that of the original problem in continuous space. In other words, this was an input of the problem. Here, we do not start from that input, but rather have access to only a raw image, without any notion of pre-specified terminals attached to it. In practice, we need to find the pixel nodes corresponding to the rectangles inside figure 1*a*. Here, we propose a method to make this selection effectively. Specifically, we select as a set of eligible terminals $\mathcal{T}(G^{\text{pe}}_{\text{tree}})$ all the leaves of the tree $G^{\text{pe}}_{\text{tree}}$ obtained from running our dynamics on $G^{\text{pe}}$ when we pass in input all pixel nodes in $G^{\text{pe}}$ as terminals, and selecting one of these at random as a source, all the rest as sinks. This choice is motivated by the fact that the tree resulting from the filtering is a good approximation of the pre-extracted $G^{\text{pe}}$, as it follows a principled optimization framework. The obtained leaves determine the coverage of this network, as they are usually located in distant parts of the network. Potentially, one could select terminal pixels ‘manually’, by using domain-knowledge to determine what pixels are the most important. However, this strategy is not scalable to a large number of images. Instead, our proposed procedure does not suffer from this problem as it can be automatically implemented, while also being flexible to receive ‘hand-picked’ terminals if available. Alternatively, a practitioner could make this selection based on some notion of network centrality, for instance selecting as terminal the most ‘central’ nodes. However, this again assumes having extra information to decide what definition of centrality is appropriate based on the application. We do not explore this here.

## 2.3. Obtaining loops

Running the dynamics of §2.1 outputs trees, while network-like structures in images might have loops. The question is thus how to recover networks that are not limited to trees. We tackle this problem by re-running the dynamics multiple times, each time selecting a particular node as source from the eligible ones (and sinks all the others). Specifically, we randomly select an individual source pixel $i \in \mathcal{T}(G^{\text{pe}}_{\text{tree}})$ and assign all the others $j \neq i \in \mathcal{T}(G^{\text{pe}}_{\text{tree}})$ as sinks. Applying the dynamics to $G^{\text{pe}}$ with this choice of one source and multiple sinks outputs a filtered network $G^f_r$, indexed by the iteration run $r$. By repeating for $N_{\text{runs}}$ this filtering step, each time selecting a different source from $\mathcal{T}(G^{\text{pe}}_{\text{tree}})$ (and all the remaining node pixels as sinks), results in a set of filtered networks $\{G^f_1, \ldots, G^f_{N_{\text{runs}}}\}$, all of them trees. We combine them by superposition, so that we obtain a final network $G(V, E, W)$ where the set of nodes and edges are the unions $V = \bigcup_{r=1}^{N_{\text{runs}}} V(G^f_r)$, and $E = \bigcup_{r=1}^{N_{\text{runs}}} E(G^f_r)$. The weights on the edges of the final network are given by the sum of the weights on each run:

$$w_{jk} = \sum_{r=1}^{N_{\text{runs}}} w^r_{jk}, \quad \forall\, j, k \in V, \tag{2.6}$$

where $w^r_{jk}$ is the weight of edge $(j, k)$ in network $G^f_r$ and corresponds to the optimal edge conductivities as obtained from the dynamics at convergence. We assume $w^r_{jk} = 0$, if $(j, k) \notin G^f_r$. The value of $N_{\text{runs}} \le |\mathcal{T}(G^{\text{pe}}_{\text{tree}})|$ is a parameter that has to be tuned based on the input image. Note that a high value of $N_{\text{runs}}$ might not necessarily result in a network more similar to the one depicted in the input image. For instance, in the extreme scenario where the original network-like structure is a tree, then $N_{\text{runs}} = 1$. Empirically, we find that a value of $N_{\text{runs}} = 5$ gives good results in all the experiments reported here, see the electronic supplementary material for more details.

Combining these steps we obtain the whole algorithmic pipeline of our method, which we refer to as Image2net. It takes in input an image and it gives in output a network. We provide an algorithmic pseudo-code in algorithm 1 and an open-source implementation at https://github.com/diegoabt/Img2net.

Our algorithm can handle input images that display distinct objects, like parts of different rivers, resulting in separate network connected components. In fact, depending on how sources and sinks are selected, our algorithm can give in output naturally more than one connected component. Alternatively, one can apply the algorithm separately on each of the different connected components and then combine the results.

**Algorithm 1.** Image2net.

```
Input: Image $\mathcal{I}$, threshold $\delta$, $\beta \ge 1$
Output: $G(V, E, W)$ final network
function Image2net $\mathcal{I}$,$\delta$,$\beta \ge 1$
    $G^{\text{pe}} \leftarrow$ NextRout pre-extraction($\mathcal{I}$,$\delta$)
    $G^{\text{pe}}_{\text{tree}} \leftarrow$ run Dynamics of §2.1 on $G^{\text{pe}}$    ▷ for $\beta$ and using in input as starting sources and sinks all the nodes in $G^{\text{pe}}$
    $\mathcal{T}(G^{\text{pe}}_{\text{tree}}) \leftarrow \{v \in V(G^{\text{pe}}_{\text{tree}}) | d_v = 1\}$    ▷ $d_v$ is the degree of node $v$
    for $r = 1, \dots, N_{\text{runs}}$ do
        Select $i \in \mathcal{T}(G^{\text{pe}}_{\text{tree}})$    ▷ uniformly at random
        $G^f_r \leftarrow$ run Dynamics of §2.1 on $G^{\text{pe}}$    ▷ for $\beta$ and using as starting source $i$ and sinks $\{j \ne i \in \mathcal{T}(G^{\text{pe}}_{\text{tree}})\}$
    end for
    $G(V, E, W) \leftarrow$ Superimpose $\left\{G^f_1, \dots, G^f_{N_{\text{runs}}}\right\}$
end function
```

# 3. Experiments on images

We run our model on three datasets of images covering various types of network-like topologies observed in biology and ecology. The images represent: (i) the slime mould *Physarum polycephalum* (*Physarum polycephalum*) [30], which is also the inspiration of our dynamics; (ii) the retinal vascular system (*retina*) [31]; (iii) river networks (*rivers*) obtained by extracting images from [32]. The number of images taken from the *Physarum polycephalum*, *retina* and *rivers* sources is 25, 20 and 10, respectively, see table 1. Pre-processing was applied, see the electronic supplementary material for details.

For model comparison, we consider NEFI, a routine that combines various image processing techniques, and a variant of our routine based on a combination of minimum spanning tree and Steiner tree optimization (Image2net-MST). The idea behind this last routine is to run our procedure but replacing the optimization steps based on the dynamics of §2.1 with standard routing optimization algorithms, namely a combination of minimum spanning and Steiner trees [34]. The goal is to see how the underlying optimization set-up impacts the final network topology. In fact, while the core idea of treating the problem of network extraction from images within the framework of routing optimization is the same for Image2net and Image2net-MST, the details of their corresponding optimization differ. Specifically, for Image2net-MST, we first run a standard MST optimization algorithm to extract $G^{\text{MST}}_{\text{tree}}$ from $G^{\text{pe}}$ (this is the same input given to Image2net). From the corresponding set of leaves $\mathcal{T}(G^{\text{MST}}_{\text{tree}})$, this time one should extract a subset of terminals $T \subseteq \mathcal{T}(G^{\text{MST}}_{\text{tree}})$

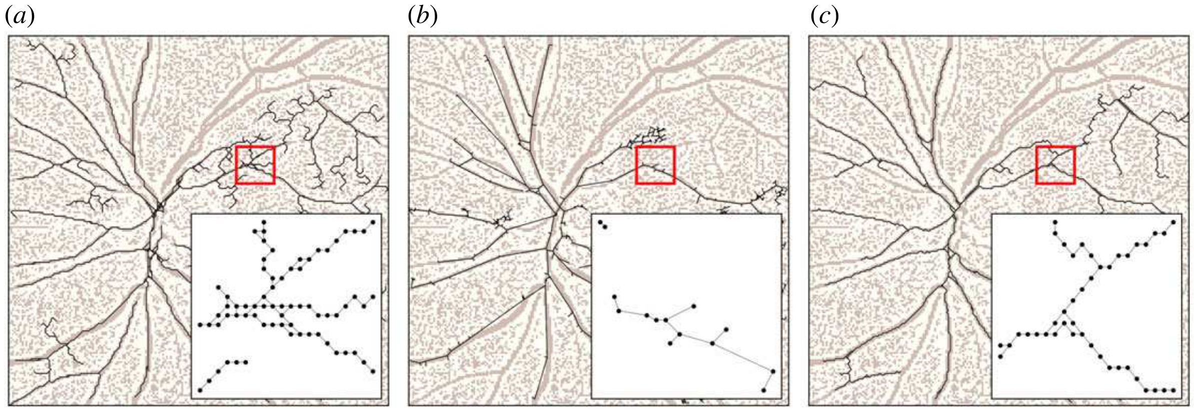


**Figure 3.** Example of network extraction. (*a*) Image2net; (*b*) NEFI; (*c*) Image2net-MST. The extracted network is coloured in black, the original image is in light brown underneath. The inset is a zoom inside the image section highlighted in red.

**Table 1.** Datasets description. (NI is the number of images used; AW is the average width of the images in the dataset; MinW and MaxW denote, respectively, the minimum and maximum width of the images in the dataset.)

| dataset | description | NI | AW | (MinW,MaxW) | ref. |
|---|---|---|---|---|---|
| *retina* | retinal blood vessels | 20 | 1791 | (998,2302) | [31,33] |
| *Physarum polycephalum* | slime mould | 25 | 400 | (400,400) | [30] |
| *rivers* | riverbed | 10 | 924 | (718,958) | [32] |

of predefined size (no distinction between source and sinks is necessary to solve a minimum Steiner tree problem). From $G^{\text{pe}}$ and $T$, extract a Steiner tree $G^{\text{St}}_r$, repeat this $N_{\text{runs}}$ times and obtain the set $\{G^{\text{St}}_1, \ldots, G^{\text{St}}_{N_{\text{runs}}}\}$. Finally, superimpose them as done for our method to obtain $G_{\text{MST}}$, see the electronic supplementary material for more details. Note that Steiner tree optimization has a complexity that scales with the number of terminals, a problem not present in our dynamics. As a result, running Image2net-MST is noticeably computationally more expensive than Image2net.

Finally, edge weights were assigned with rules specific to each method, as there is no common definition that applies to all of them. In fact, the ability to extract edge weights is rare among image processing techniques, and usually relies on image preprocessing and segmentation of the input image. Instead, Image2net extracts edge weights in a principled way based on the results at optimality in terms of conductivity, hence it has a nice direct interpretation as the diameter of the edges in the image. For Image2net, we use the rule effective reweighing (ER) on the resulting conductivities, see [13]; for Image2net-MST, we use the weights given in input to solve the Steiner tree problem, i.e. the weights given by ER rule in $G^{\text{pe}}$; for NEFI, we use as weight the *width*, this is an output of the algorithm; for the original image, we assign the RGB values of the pixels mapped into an integer number increasing with the colour intensity (see the electronic supplementary material for details). All of these definitions of weight agree on the higher the weight the thicker the edge is, and thus the conductivity. Figure 3 illustrates an example of the networks extracted using the various algorithms for an image in *retina*.

## 3.1. Performance metrics

We measure performance in terms of the ability of an algorithm to recover the network-like subject depicted on the underlying image. We consider a measure of similarity adapted from the quality measure defined in [13]. This relies on partitioning the image in a grid of $P$ non-intersecting subsets $C_\alpha$ inside the pixels' domain and then compare the edges of $G$ within $C_\alpha$ assigned by the algorithm and those observed in the original image $I$ (RGB values):

$$\hat{w}_b(G, I) = \frac{1}{P}\left[\sum_{\alpha=1}^{P}\left(\left|\sum_{e\in E}\mathbb{I}_\alpha(e) - \sum_{i\in I}\mathbb{I}_\alpha(\delta, i)\right|\right)^2\right]^{1/2}, \tag{3.1}$$

where $\mathbb{I}_\alpha(\delta, i) = 1, 0$ for $i \in I$, if the pixel $i$ is in $C_\alpha$ or not, respectively; $\delta$ is a threshold used to decide whether that pixel contributes to the network-like image. In words, if the pixel colour intensity is high enough, then we label it as an edge. For $\delta$, we use the same value as used in input to Image2net. This is a coarse-grained measure of similarity that tells how many edges in the extracted graph correspond to high-intensity pairs of pixels. In order to account for edge weights and pixel intensities, we also consider a weighted version of this:

$$\hat{w}(G, I) = \frac{1}{P}\left[\sum_{\alpha=1}^{P}\left(\left|\sum_{e\in E}\mathbb{I}_{\alpha}(e)w_e - \sum_{i\in I}\mathbb{I}_{\alpha}(i)p_i\right|\right)^2\right]^{1/2}, \tag{3.2}$$

where $p_i$ is the intensity of the pixel $i$, and $\mathbb{I}_{\alpha}(i)$ is 1 if $i \in C_{\alpha}$, and 0 otherwise. Note that in this case $\delta$ is not needed because pixels with low intensity are penalized by lower weight in their contributions to $\hat{w}(G, I)$. In both cases, small values of these measures mean higher similarity values between the extracted network and the underlying network-like structure in the image.

While ground-truth for this network-like structure is normally absent, the *retina* dataset contains ground-truth networks which were hand-labelled by individuals [31]. In this case, we calculate the binary similarity using the hand-labelled images instead of the one given in input. There are two sets of labelled images, each corresponding to a different person doing this manual identification. While similar, the resulting two sets of networks are different. In the absence of ground truth, we compare against the input image.

### 3.2. Implementation details

We apply image pre-processing to the input image to improve image quality and distinguish the main subject from the background, see the electronic supplementary material for details. We rescale NEFI's pixels' location to have them in the same scale as that of the other methods, i.e. the set $[0, 1] \times [0, 1]$ (for simplicity, we consider only square-shaped images). All the edge lengths $\ell_e$ have been assigned using the Euclidean distance between the corresponding endpoints. For NEFI, we used the two best performing pipelines of image-processing techniques *polycephalum_high* (`NEFI-high`) and *crack_patterns* (`NEFI-crack`) among the available predefined pipelines. In the figures, we show the best results only, these vary based on the image given in input.

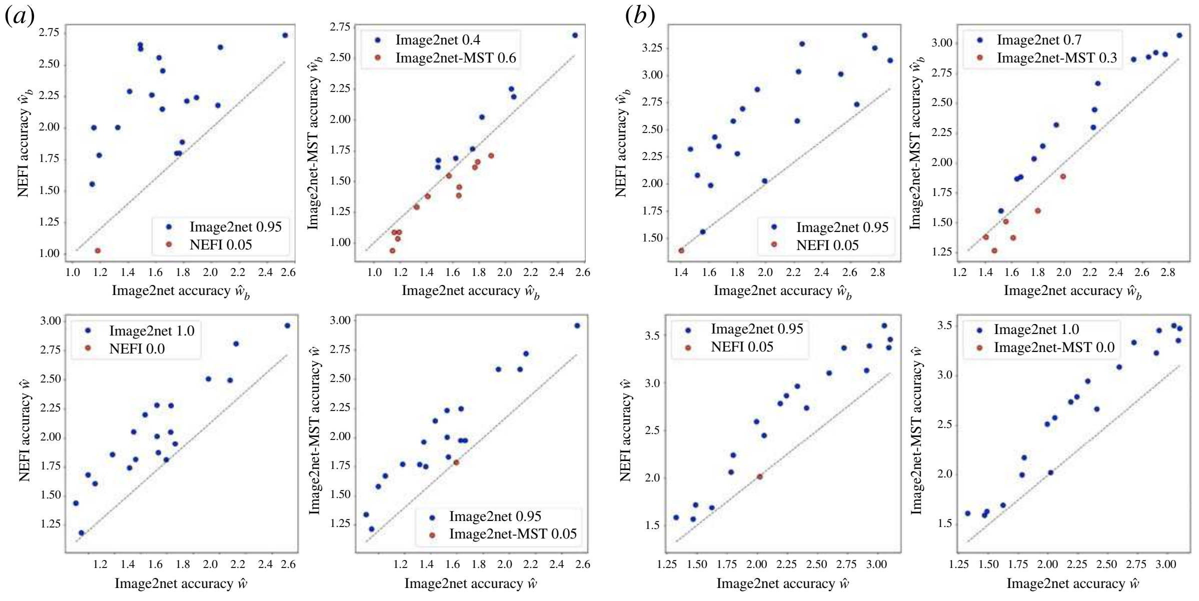
## 4. Results

### 4.1. Retinal vessel image validation

We use the similarity measure defined in the previous section to compare every graph-based approximation of the image with the provided hand-labelled ones, assuming these last ones to be the ground truth $I_{gt}$. We compute $\hat{w}_b(G, I_{gt})$ for each retinal image and the corresponding extracted network, to measure how close a particular network is from the human-labelled one. Figure 4 shows that Image2net consistently outperforms NEFI over all images and the two hand-labelled datasets. Image2net and Image2net-MST perform similarly according to the binary similarity. However, if we account for weight, we obtain the Image2net outperforms Image2net-MST in the majority of the images. Note that Image2net-MST does not assign new weights while selecting the edges, as in a Steiner tree problem, instead, it uses the weights of the input network, in this case, $G^{\mathrm{pe}}$. Instead, Image2net selects edges and weights at the same time, within the same optimization set-up. The fact that the weighted similarity gives better results, signals that the values of the optimal conductivities (the weights assigned to Image2net extracted networks) have a meaningful interpretation, as they better match the pixel's intensities than the weights given by the other algorithms.

### 4.2. *Physarum polycephalum* and *rivers* networks

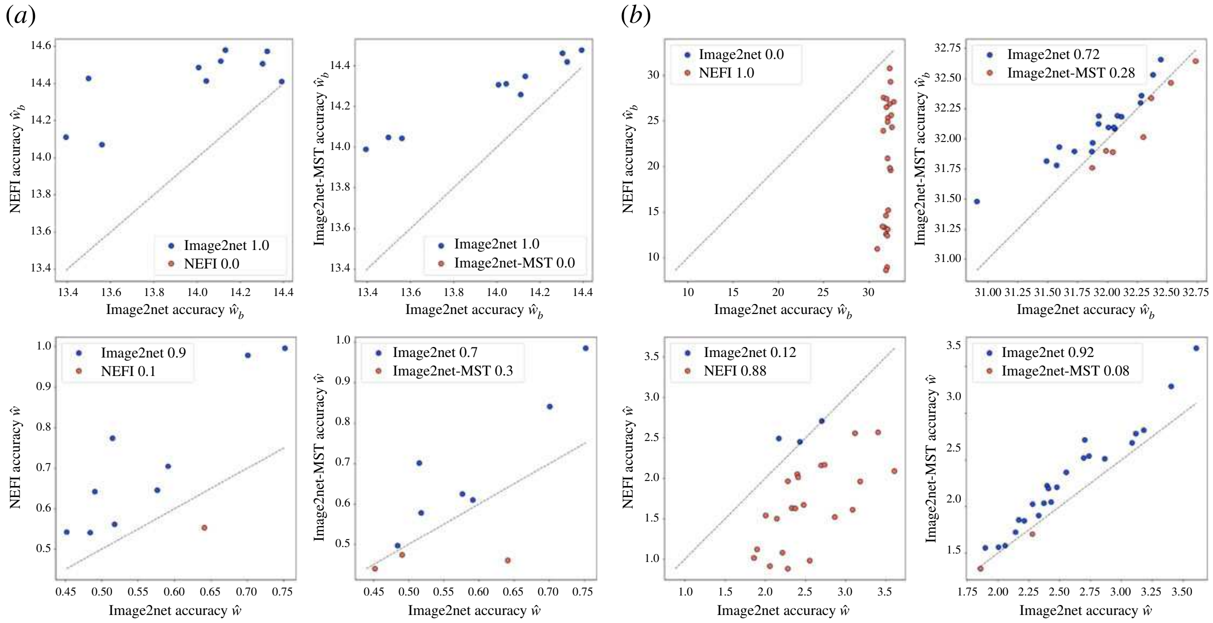
We measure the performance in the two datasets where there is no ground truth, which is often the case in real images. We find that Image2net recovers better the *rivers* networks, for both performance metrics as we show in figure 5. In fact, our model is able to capture the detailed geometry of the network when there are curves, while NEFI has limitations in that edges with curves or kinks are contracted to straight lines. This is one of the main advantages of our model based on an underlying optimization framework, the geometry of the network is automatically selected based on optimality, rather than a predefined setting manually tuned. As a result, Image2net is flexible in detecting different network geometries, as



hand-labelled by A. H.

hand-labelled by V. K.

**Figure 4.** Recovering hand-labelled networks. Performance in terms of similarity $\hat{w}_b(G, I)$ and $\hat{w}(G, I)$ on hand-labelled *retina* networks, (*a*) and (*b*) are networks labelled by two different people. First row shows $\hat{w}_b$ values; second row shows $\hat{w}$ values. Smaller values mean higher similarity and thus better performance. Hence, points above the grey line (blue) means Image2net performs better, whereas points below (red) means worse performance. (*a*) Hand-labelled by A. Hoover (*b*) Hand-labelled by V. Kouznetsova.



*rivers*

*physarum polycephalum*

**Figure 5.** Recovering *river* and *Physarum polycephalum* networks. Performance in terms of similarity $\hat{w}_b(G, I)$ and $\hat{w}(G, I)$ on *rivers* (2 leftmost columns) and *Physarum polycephalum* (two rightmost columns) networks. Smaller values mean higher similarity and thus better performance. Hence, points above the grey line (blue) means Image2net performs better, whereas points below (red) means worse performance. (*a*) *rivers* and (*b*) *Physarum polycephalum*.

can be seen in figure 6 (top). The situation for *Physarum polycephalum* is more nuanced as Image2net is better than Image2net-MST, in particular when considering the weights, but NEFI outperforms all the others. However, this is true if we use the `NEFI-high` routine, which is the one built on purpose to detect *Physarum polycephalum* networks, it is not surprising that this has stronger results on these datasets. In figure 6 (bottom), we note how these networks contain many small details that are better captured by NEFI. Indeed, using other NEFI routines, performance aligns more to Image2net and Image2net-MST. This also shows that if a practitioner aims at extracting networks from a particular image, all the approaches allow for few degrees of freedom to be tuned in order to increase performance. NEFI allows the specification of individual routines to design a custom pipeline,

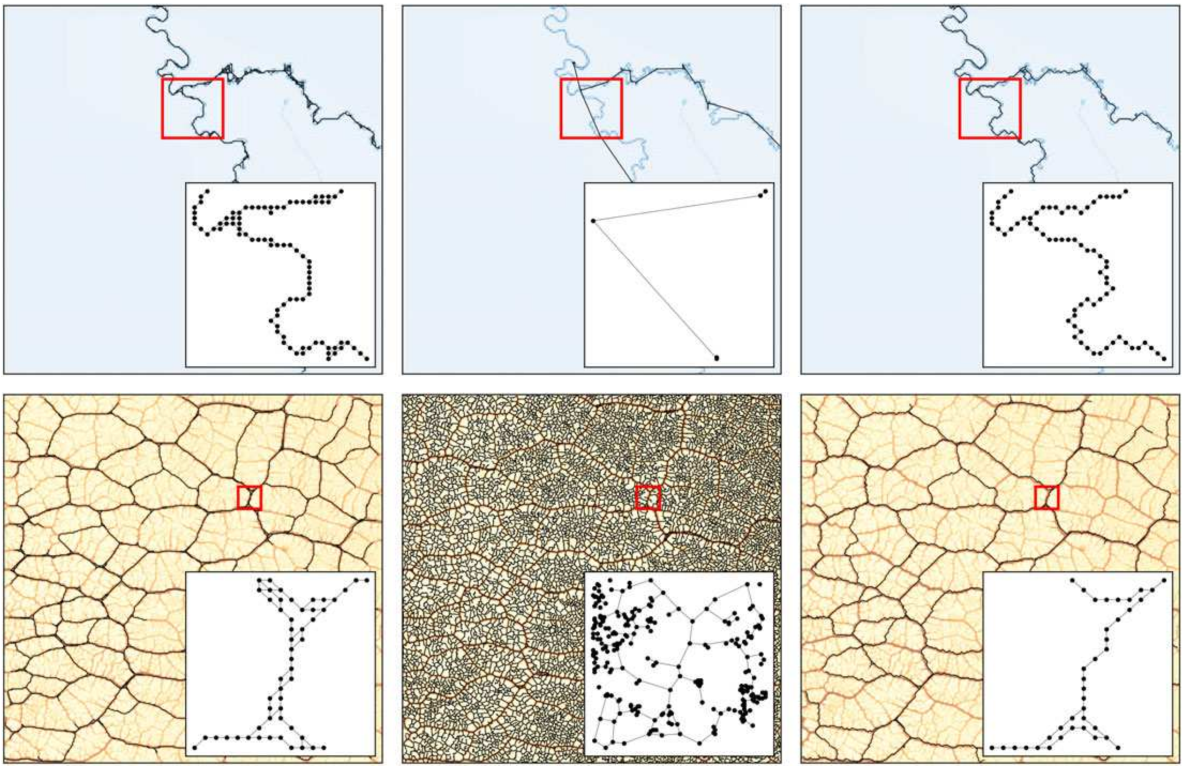


**Figure 6.** Results on *rivers* and *Physarum polycephalum* networks. We show the networks extracted on *rivers* (top) and *Physarum polycephalum* (bottom) using Image2net (left), NEFI (centre) and Image2net-MST (right). Inset is the zoom over the area under the red surface. The input image is depicted underneath the networks.

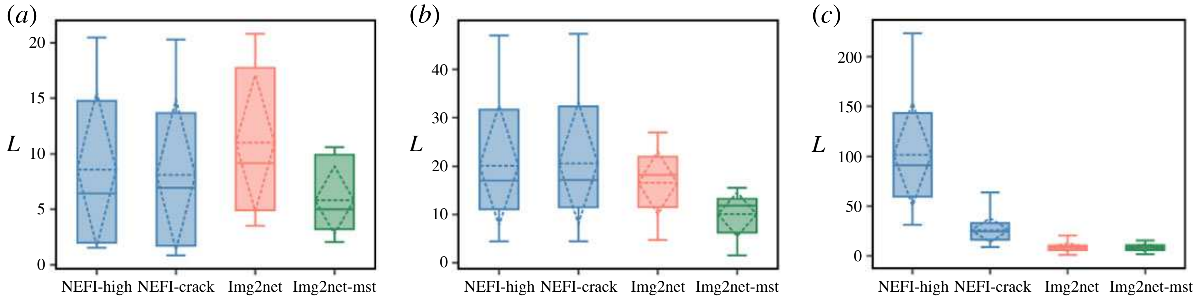


**Figure 7.** Total network length. Boxes show the distribution over the images inside each dataset of the total network length $L = \sum_e \ell_e$ calculated on the networks extracted by each method. Solid lines are the median, dashed lines are the average. (*a*) *rivers*, (*b*) *retina* and (*c*) *Physarum polycephalum*.

Image2net and Image2net-MST have various parameters that could be tuned, the most important being $\delta$ and $\beta$. For instance, decreasing $\delta$ will allow for more fine details on *Physarum polycephalum* networks, see the electronic supplementary material. However, tuning each routine on each input image goes beyond the scope of this work, as we aim at describing how different approaches perform on a corpus of images, potentially quite different, and thus automatize network extraction in a scalable way.

## 5. Qualitative results

Beyond validating the model on recovering network structure that resembles well what is pictured in an image, we illustrate the differences in topological properties of the extracted networks. This also showcases possible applications for our model, where a practitioner extracts a network and can then perform further analysis on it, for instance using the detected network properties.

We calculated the total network length as $L = \sum_e \ell_e$ where $\ell_e$ is the Euclidean distance between the nodes defining the edge $e$, see figure 7. We find that Image2net extracts on average longer *rivers* networks, and similar to NEFI for the *retina*, but with lower variance in this case. Instead, NEFI extracts much longer *Physarum polycephalum* networks, mainly owing to many small minor paths permeating the whole image (this was signalled above by wider result difference in terms of similarity). Instead,

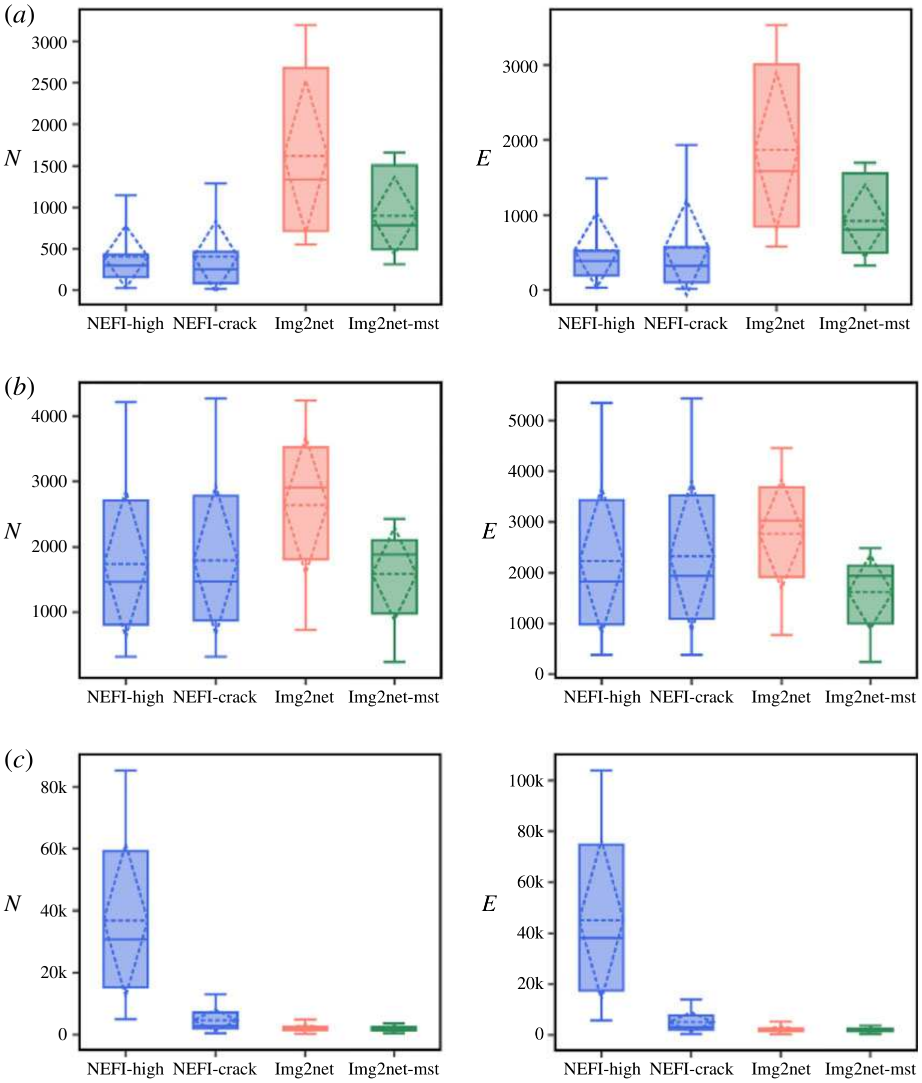


**Figure 8.** Number of nodes and edges. Boxes show the distribution over the images inside each dataset of $N$, the total number of nodes (*a*) and $E$, the total number of edges (*b*), calculated on the networks extracted by each method. Solid lines are the median, dashed lines are the average. (*a*) *rivers*, (*b*) *retina* and (*c*) *Physarum polycephalum*.

Image2net-MST finds smaller values in all the datasets. This highlights one important difference owing to the underlying optimization set-up that distinguished these two approaches.

While a longer network total length might be owing to a higher number of edges, this is not always the case. This can be seen from results on *rivers* in figure 8, where we plot the distribution of the number of nodes and edges, other important topological properties. In fact, for these images, NEFI finds much smaller network sizes than Image2net, while the distribution of $L$ in figure 7 is similar for the two routines. This is again owing to NEFI representing curved parts of the network with fewer but longer straight edges, see figure 6 for an example. In these river networks, Image2net has a higher resolution, where NEFI fails to find enough details. The opposite extreme is seen for the *Physarum polycephalum* images where NEFI has many more nodes and edges when using the routine `NEFI-high`, and also much higher $L$ as we saw before. For the *retina* vessel networks, Image2net extracts on average networks with higher number of nodes and edges than the other two methods, while $L$ is similar to NEFI, hence both Image2net and Image2net-MST have on average shorter edges than NEFI, with the difference that Image2net extract networks with bigger sizes.

## 6. Conclusion

We propose Image2net, a model for extracting networks from images. It takes as input an image and returns a network structure as a set of nodes, edges and the corresponding weights. Standard approaches for addressing these problems rely on image processing techniques. Instead, our model is based on a principled formalism adapted from recent results of optimal transport theory. We build an analogy with fluid dynamics by treating colours on pixels as fluids flowing through an image and considering a set of dynamical equations for their conductivities and flows. At convergence, these correspond to stationary solutions of a cost function that has a nice interpretation in terms of a transportation cost.

The advantage of our approach with respect to more conventional methods is that our model naturally incorporates a principle definition of edge weights as the optimal conductivities and that can be interpreted as proportional to the diameters of network edges. In addition, it allows for a principled and automatic filtering of possible redundancies by means of solving a routing optimization problem, instead of using pre-defined pruning routines.

We test our model on various datasets, and calculate performance measures in terms of recovering the network-like shape in the input image. Image2net performs well compared to other network extraction tools and yields networks that closely approximate the networks depicted in the images. In particular, it is flexible in finding various network shapes, as it can find curved geometries as those observed in river networks.

We expect our method to be appropriate for images that represent phenomena with well-defined and physically meaningful flows or fluxes, as in the datasets considered here. In particular, for underlying conserved incompressible flows as water or blood flows. However, not all networks shown in images represent flow, for instance, fracture networks, foams, or grain boundaries in materials. While our method is still applicable to these scenarios, as it is agnostic to what image is given in input, the resulting networks may not be meaningful in these cases. In addition, there may be also flow-based systems displaying many loops or dynamical behaviours changing frequently in time, e.g. tidal marshes, which may not be captured well by our model. A possible solution for obtaining many loops could be to adapt Image2net to a multi-commodity optimal transport approach as in [35], which can naturally lead to loopy structures. We thus encourage practitioners to consciously select the images given in input to our algorithm based on the expected behaviour of the underlying physical phenomena being displayed.

In addition to being efficient, automated network extraction also has the advantage of yielding reproducible results and reducing human biases. Indeed, given an input image, Image2net will always yield the same networks, whereas manual extraction depends on the perception of the individual performing the measurement. Our model also enables practitioners to measure network-related quantities like centrality measures, branching points or curvature and angles. More importantly, given the computational efficiency of the underlying solver, it also works for large networks where manually measuring metrics across the whole network is not feasible.

In this work, we mostly show example applications from biology and ecology, but the usage of our model is not limited to this kind of networks. It can be used in a broad array of datasets to detect and measure network-like shapes, in particular, those displaying systems with incompressible flows. We foresee that our model will be useful for practitioners willing to perform automatic and scalable network analysis of large datasets of images.

Data accessibility. All data needed to evaluate the conclusions in the paper are present in the paper and/or the electronic supplementary material [36]. An open-source implementation of the code is available online at https://github.com/diegoabt/Img2net.

Authors' contributions. D.B. and C.D.B. derived the model, analysed results and wrote the manuscript. D.B. conducted the experiments.

Competing interests. The authors declare that they have no competing interests.

Funding. No funding has been received for the article.

Acknowledgements. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Diego Baptista.

# References

1. Balister P, Balogh J, Bertuzzo E, Bollobás B, Caldarelli G, Maritan A, Mastrandrea R, Morris R, Rinaldo A. 2018 River landscapes and optimal channel networks. *Proc. Natl Acad. Sci. USA* **115**, 6548–6553. (doi:10.1073/pnas.1804484115)

2. Tsai PS, Kaufhold JP, Blinder P, Friedman B, Drew PJ, Karten HJ, Lyden PD, Kleinfeld D. 2009 Correlations of neuronal and microvascular densities in murine cortex revealed by direct counting and colocalization of nuclei and vessels. *J. Neurosci.* **29**, 14 553–14 570. (doi:10.1523/JNEUROSCI.3287-09.2009)

3. Yin C, Xiao X, Balaban V, Kandel ME, Lee YJ, Popescu G, Bogdan P. 2020 Network science characteristics of brain-derived neuronal cultures deciphered from quantitative phase imaging data. *Sci. Rep.* **10**, 1–13. (doi:10.1038/s41598-019-56847-4)

4. Gazit Y, Berk DA, Leunig M, Baxter LT, Jain RK. 1995 Scale-invariant behavior and vascular network formation in normal and tumor tissue. *Phys. Rev. Lett.* **75**, 2428. (doi:10.1103/PhysRevLett.75.2428)

5. Boddy L, Wood J, Redman E, Hynes J, Fricker MD. 2010 Fungal network responses to grazing. *Fungal Genet. Biol.* **47**, 522–530. (doi:10.1016/j.fgb.2010.01.006)

6. Montoya-Zegarra JA, Russo E, Runge P, Jadhav M, Willrodt A-H, Stoma S, Nørrelykke SF, Detmar M, Halin C. 2019 Autotube: a novel software for the automated morphometric analysis of vascular networks in tissues. *Angiogenesis* **22**, 223–236. (doi:10.1007/s10456-018-9652-3)

7. Nikolaisen J, Nilsson LI, Pettersen IK, Willems PH, Lorens JB, Koopman WJ, Tronstad KJ. 2014 Automated quantification and integrative analysis of 2D and 3D mitochondrial shape and network properties. *PLoS ONE* **9**, e101365. (doi:10.1371/journal.pone.0101365)

8. Ouellet M, Guillebaud G, Gervais V, St-Pierre DL, Germain M. 2017 A novel algorithm identifies stress-induced alterations in mitochondrial connectivity and inner membrane structure from confocal images. *PLoS Comput. Biol.* **13**, e1005612. (doi:10.1371/journal.pcbi.1005612)

9. Banavar JR, Colaiori F, Flammini A, Maritan A, Rinaldo A. 2000 Topology of the fittest transportation network. *Phys. Rev. Lett.* **84**, 4745. (doi:10.1103/PhysRevLett.84.4745)

10. Hu J, Razdan A, Femiani JC, Cui M, Wonka P. 2007 Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Trans. Geosci. Remote Sens.* **45**, 4144–4157. (doi:10.1109/TGRS.2007.906107)

11. Tupin F, Maitre H, Mangin J-F, Nicolas J-M, Pechersky E. 1998 Detection of linear features in SAR images: application to road network extraction. *IEEE Trans. Geosci. Remote Sens.* **36**, 434–453. (doi:10.1109/36.662728)

12. Bastani F, He S, Abbar S, Alizadeh M, Balakrishnan H, Chawla S, Madden S, DeWitt D. 2018 RoadTracer: automatic extraction of road networks from aerial images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4720–4728. New York, NY: IEEE.

13. Baptista D, Leite D, Facca E, Putti M, De Bacco C. 2020 Network extraction by routing optimization. *Sci. Rep.* **10**, 1–3. (doi:10.1038/s41598-019-56847-4)

14. Dehkordi MT, Sadri S, Doosthoseini A. 2011 A review of coronary vessel segmentation algorithms. *J. Med. Signals Sens.* **1**, 49. (doi:10.4103/2228-7477.83519)

15. Fricker MD, Akita D, Heaton LL, Jones N, Obara B, Nakagaki T. 2017 Automated analysis of *Physarum* network structure and dynamics. *J. Phys. D: Appl. Phys.* **50**, 254005. (doi:10.1088/1361-6463/aa72b9)

16. Lasser J, Katifori E. 2017 Net: a new framework for the vectorization and examination of network data. *Source Code Biol. Med.* **12**, 4. (doi:10.1186/s13029-017-0064-3)

17. Bühler J, Rishmawi L, Pflugfelder D, Huber G, Scharr H, Hülskamp M, Koornneef M, Schurr U, Jahnke S. 2015 Phenovein–a tool for leaf vein segmentation and analysis. *Plant Physiol.* **169**, 2359–2370. (doi:10.1104/pp.15.00974)

18. Wang W, Yang N, Zhang Y, Wang F, Cao T, Eklund P. 2016 A review of road extraction from remote sensing images. *J. Traffic Transp. Eng. (English edition)* **3**, 271–282. (doi:10.1016/j.jtte.2016.05.005)

19. Rapacz M, Łazarz R. 2020 Automatic extraction of leaf venation complex networks. *ECAI 2020*, pp. 1914–1921. Amsterdam, The Netherlands: IOS Press.

20. Price CA. 2012 Leaf gui: analyzing the geometry of veins and areoles using image segmentation algorithms. In *High-throughput phenotyping in plants*, pp. 41–49. New York, NY: Springer.

21. Obara B, Grau V, Fricker MD. 2012 A bioimage informatics approach to automatically extract complex fungal networks. *Bioinformatics* **28**, 2374–2381. (doi:10.1093/bioinformatics/bts364)

22. Baumgarten W, Hauser M. 2010 Detection, extraction, and analysis of the vein network. *J. Comput. Interdiscipl. Sci.* **1**, 241–249.

23. Fraz MM, Remagnino P, Hoppe A, Uyyanonvara B, Rudnicka AR, Owen CG, Barman SA. 2012 Blood vessel segmentation methodologies in retinal images–a survey. *Comput. Methods Programs Biomed.* **108**, 407–433. (doi:10.1016/j.cmpb.2012.03.009)

24. Chai D, Forstner W, Lafarge F. 2013 Recovering line networks in images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1894–1901. New York, NY: IEEE.

25. Dirnberger M, Kehl T, Neumann A. 2015 Nefi: Network extraction from images. *Sci. Rep.* **5**, 15669. (doi:10.1038/srep15669)

26. Breuer D, Nikoloski Z. 2015 Define: an optimisation-based method for robust disentangling of filamentous networks. *Sci. Rep.* **5**, 1–14. (doi:10.1038/srep18267)

27. Cinsdikici MG, Aydın D. 2009 Detection of blood vessels in ophthalmoscope images using MF/ant (matched filter/ant colony) algorithm. *Comput. Methods Programs Biomed.* **96**, 85–95. (doi:10.1016/j.cmpb.2009.04.005)

28. Tero A, Kobayashi R, Nakagaki T. 2007 A mathematical model for adaptive transport network in path finding by true slime mold. *J. Theor. Biol.* **244**, 553–564. (doi:10.1016/j.jtbi.2006.07.015)

29. Tero A, Takagi S, Saigusa T, Ito K, Bebber DP, Fricker MD, Yumiki K, Kobayashi R, Nakagaki T. 2010 Rules for biologically inspired adaptive network design. *Science* **327**, 439–442. (doi:10.1126/science.1177894)

30. Dirnberger M, Mehlhorn K, Mehlhorn T. 2017 Introducing the slime mold graph repository. *J. Phys. D: Appl. Phys.* **50**, 264001. (doi:10.1088/1361-6463/aa7326)

31. Hoover A, Kouznetsova V, Goldbaum M. 2000 Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Trans. Med. Imaging* **19**, 203–210. (doi:10.1109/42.845178)

32. Openseamap, see https://map.openseamap.org/. Accessed: from 7–28 September 2020.

33. Hoover A, Kouznetsova V, Goldbaum M. 2000 Hand-labeled dataset. See https://cecas.clemson.edu/˜ahoover/stare/probing/index.html.

34. Hwang FK, Richards DS. 1992 Steiner tree problems. *Networks* **22**, 55–89. (doi:10.1002/net.3230220105)

35. Lonardi A, Facca E, Putti M, De Bacco C. 2020 Optimal transport for multi-commodity routing on networks. *arXiv*, 2010.14377.

36. Baptista D, De Bacco C. 2021 Supplementary material from 'Principled network extraction from images'. The Royal Society. Collection. https://doi.org/10.6084/m9.figshare.c.5514773.

frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 27 February 2023
DOI 10.3389/fphy.2023.1089114

# Immiscible color flows in optimal transport networks for image classification

Alessandro Lonardi*[†], Diego Baptista*[†] and Caterina De Bacco

Physics for Inference and Optimization Group, Max Planck Institute for Intelligent Systems, Cyber Valley, Tübingen, Germany

In classification tasks, it is crucial to meaningfully exploit the information contained in the data. While much of the work in addressing these tasks is focused on building complex algorithmic infrastructures to process inputs in a black-box fashion, little is known about how to exploit the various facets of the data before inputting this into an algorithm. Here, we focus on this latter perspective by proposing a physics-inspired dynamical system that adapts optimal transport principles to effectively leverage color distributions of images. Our dynamics regulates immiscible fluxes of colors traveling on a network built from images. Instead of aggregating colors together, it treats them as different commodities that interact with a shared capacity on the edges. The resulting optimal flows can then be fed into standard classifiers to distinguish images in different classes. We show how our method can outperform competing approaches on image classification tasks in datasets where color information matters.

KEYWORDS

**network flow optimization, image classification, network optimization, optimal transport, self-adapting dynamical systems**

## 1 Introduction

Optimal transport (OT) is a powerful method for computing the distance between two data distributions. This problem has a cross-disciplinary domain of applications, ranging from logistics and route optimization [1–3] to biology [4, 5] and computer vision [6–10], among others. Within this broad variety of problems, OT is largely utilized in machine learning [11] and deployed for solving classification tasks, where the goal is to optimally match discrete distributions that are typically learned from data. Relevant usage examples are also found in multiple fields of physics, as in protein fold recognition [12], stochastic thermodynamics [13], designing transportation networks [14, 15], routing in multilayer networks [16], or general relativity [17]. A prominent application is image classification [18–23], where the goal is to measure the similarity between two images. OT solves this problem by interpreting image pairs as two discrete distributions and then assessing their similarity *via* the Wasserstein ($W_1$) distance ([24], Definition 6.1), a measure obtained by minimizing the cost needed to transform one distribution into the other. Using $W_1$ for image classification carries many advantages over other similarity measures between histograms. For example, $W_1$ preserves all properties of a metric [9, 24], it is robust over domain shift for train and test data [22], and it provides meaningful gradients to learn data distributions on non-overlapping domains [25]. Because of these and several other desirable properties, much research effort has been put into speeding up algorithms to calculate $W_1$ [12, 19, 20, 26, 27]. However, all these methods overlook the potential of effectively using image colors directly in the OT formulation. As a result, practitioners have access to increasingly efficient algorithms, but those do not necessarily

OPEN ACCESS

EDITED BY
Adriano Tiribocchi,
National Research Council (CNR), Italy

REVIEWED BY
Giovanni Franzina,
Istituto per le Applicazioni del Calcolo (IAC), Italy
Pablo Villegas,
Enrico Fermi Center for Study and Research, Italy

*CORRESPONDENCE
Alessandro Lonardi,
alessandro.lonardi@tuebingen.mpg.de
Diego Baptista,
diego.theuerkauf@tuebingen.mpg.de

[†]These authors have contributed equally to this work and share first authorship

SPECIALTY SECTION
This article was submitted to Complex Systems, a section of the journal Frontiers in Physics

RECEIVED 03 November 2022
ACCEPTED 20 January 2023
PUBLISHED 27 February 2023

CITATION
Lonardi A, Baptista D and De Bacco C (2023), Immiscible color flows in optimal transport networks for image classification. *Front. Phys.* 11:1089114. doi: 10.3389/fphy.2023.1089114

improve accuracy in predictions, as we lack a framework that fully exploits the richness of the input information.

Colored images originally encoded as three-dimensional histograms—with one dimension per color channel—are often compressed into lower dimensional data using feature extraction algorithms [9, 23]. Here, we propose a different approach that maps the three distinct color histograms to multicommodity flows transported in a network built using images' pixels. We combine recent developments in OT with the physics insights of capacitated network models [1, 5, 28–31] to treat colors as masses of different types that flow through the edges of a network. Different flows are coupled together with a shared conductivity to minimize a unique cost function. This setup is reminiscent of the distinction between modeling the flow of one substance, e.g., water, and modeling the flows of multiple substances that do not mix, e.g., immiscible fluids, which share the same network infrastructure. By virtue of this multicommodity treatment, we achieve stronger classification performance than state-of-the-art OT-based algorithms in real datasets where color information matters.

# 2 Problem formulation

## 2.1 Unicommodity optimal transport

Given two $m$- and $n$-dimensional probability vectors $g$ and $h$ and a positive-valued ground cost matrix $C$, the goal of a standard—unicommodity—OT problem is to find an optimal transport path $P^\star$ satisfying the conservation constraints $\sum_j P_{ij} = g_i \forall i$ and $\sum_i P_{ij} = h_j \forall j$, while minimizing $J(g, h) = \sum_{ij} P_{ij} C_{ij}$.

Entries $P^\star_{ij}$ can be interpreted as the mass transported from $g_i$ to $h_j$ when paying a cost $C_{ij}$, while $J^\star$, i.e., $J$ evaluated at $P^\star$, encodes the minimum effort needed to transport $g$ to $h$. Notably, if all entries $C_{ij}$ are distances between $i$ and $j$, then $J^\star$ is the $W_1$ distance between $g$ and $h$ (see [24] for a standard proof and [9] for derivations focusing on the discrete case).

## 2.2 Physics-inspired multicommodity optimal transport

Interpreting colors as masses traveling along a network built from images' pixels (as we define in detail below), unicommodity OT could be used to capture the similarity between grayscale images. However, it may not be ideal for colored images, when color information matters. The limitation of unicommodity OT in Section 2.1 is that it does not fully capture the variety of information contained in different color channels as it is not able to distinguish them. Motivated by this, we tackle this challenge and move beyond this standard setting by incorporating insights from the dynamics of immiscible flows into physics. Specifically, we treat the different pixels' color channels as masses of different types that do not mix but rather travel and interact on the same network infrastructure, while optimizing a unique cost function. By assuming capacitated edges with conductivities that are proportional to the amount of mass traveling through an edge, we can define a set of ODEs that regulate fluxes and conductivities. These are optimally distributed along a network to better account for color information while satisfying physical conservation laws. Similar ideas have been successfully used to route different types of passengers in transportation networks [2, 16, 32].

Formally, we couple together the histograms of $M = 3$ color channels, the *commodities*, indexed with $a = 1, \ldots, M$. We define $g^a$ and $h^a$ as $m$- and $n$-dimensional probability vectors of mass of type $a$. More compactly, we define the matrix $G$ with entries $G_{ia} = g^a_i$ (respectively, $H$ for $h$), each containing the intensity of color channel $a$ in pixel $i$ of the first (respectively, second) image. These regulate the sources and sinks of mass in our setting. We then enforce the conservation of mass for each commodity index $a$ $\sum_i g^a_i = \sum_j h^a_j$. This ensures that all the color mass in the first image is accounted for in the second image, and vice versa. This should be valid for each mass type.

Moreover, we define the set $\Pi(G, H)$ containing $(m \times n \times M)$-dimensional tensors $P$ with entries $P^a_{ij}$ being transport paths between $g^a$ and $h^a$. These regulate how fluxes of colors of different types travel along a network. We enforce the interaction between transport paths for different commodities by introducing a *shared cost*.

$$J_\Gamma(G,H) = \sum_{ij} \|P_{ij}\|_2^\Gamma C_{ij}, \tag{1}$$

where $\|P_{ij}\|_2 = (\sum_a P^a_{ij} 2)^{1/2}$ is the 2-norm of the vector $P_{ij} = (P^1_{ij}, \ldots, P^M_{ij})$ and $0 < \Gamma < 4/3$ is a regularization parameter. We take $\Gamma > 0$ since a negative exponent would favor the proliferation of loops with infinite mass [28]. Instead, we conventionally consider $\Gamma < 4/3$ (see Section 3.2) since the cost $J_\Gamma$ exhibits the same convexity properties for any $\Gamma > 1$, i.e., it is strictly convex, and OT paths do not change substantially with $\Gamma$ in this regime [2]. We can thus formulate its corresponding multicommodity OT problem as that of finding a tensor $P^\star$ solution of

$$J^\star_\Gamma(G,H) = \min_{P \in \Pi(G,H)} J_\Gamma(G,H). \tag{2}$$

It should be noted that for $M = 1$ and $\Gamma = 1$, we recover the standard unicommodity OT setup.

The problem in Eq. 2 admits a precise physical interpretation. In fact, it can be recast as a constrained minimization problem with the objective function being the energy dissipated by the multicommodity flows (Joule's law) and a constant total conductivity. Furthermore, transport paths follow Kirchhoff's law enforcing conservation of mass [2, 32, 33] (see Supplementary Material for a detailed discussion).

Noticeably, $J_\Gamma$ is a quantity that takes into account all the different mass types, and the OT paths $P^\star$ are found through a unique optimization problem. We emphasize that this is fundamentally different from solving $M$-independent unicommodity problems, where different types of mass are not coupled together as in our setting, and then combining their optimal costs to estimate images' similarity. Estimating $J^\star_\Gamma(G,H)$ directly gives a quantitative and principled measure of the similarity between two images $G$ and $H$. The lower this cost, the higher the similarity of the two images. While this is valid also for the unicommodity cost in Section 2.1, the difference here is that we account differently for the color information as we distinguish different colors *via* the $M$-dimensional vector $P_{ij}$. The cost in Eq. 2 then properly couples colors by following physical laws regulating immiscible flows. The idea is that if this information matters for the given classification task, incorporating it into the minimization problem would output a cost that helps to distinguish images better, e.g., with higher accuracy.

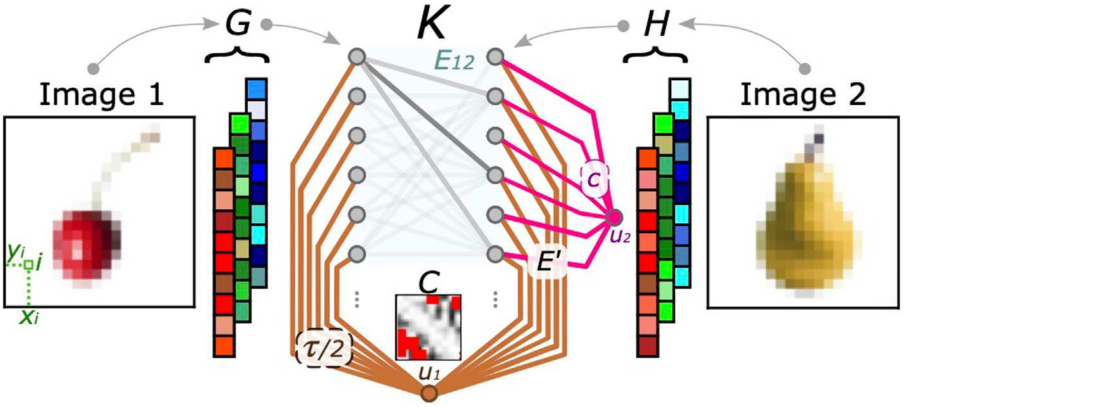


FIGURE 1
Bipartite network representation for multicommodity OT. The two images (shown on the leftmost and rightmost sides of the panel) are encoded in the RGB matrices $G$ and $H$, which regulate the flow traveling on the network $K$. The graph is made of $m + n + 2$ nodes, i.e., the total number of pixels plus the two auxiliary vertices introduced in Section 3.1. Gray edges (belonging to the set $E_{12}$) connect nodes in image 1 to nodes in image 2; these edges are trimmed according to a threshold $\tau$. We highlight the entries of the matrix $C$ in red if these are larger than $\tau$. Transshipment and auxiliary edges used to relax mass conservation (which belong to $E'$) are colored in brown and magenta.

# 3 Materials and methods

## 3.1 Optimal transport network on images

Having introduced the main ideas and intuitions, we now explain in detail how to adapt the OT formalism to images. Specifically, we introduce an auxiliary bipartite network $K_{m,n}(V_1, V_2, E_{12})$, which is the first building block of the network where the OT problem is solved. A visual representation of this is shown in Figure 1. The images 1 and 2 are represented as matrices ($G$ and $H$) of sizes $m \times M$ and $n \times M$, respectively, where $M$ is the number of color channels of the images ($M = 3$ in our examples). The sets of nodes $V_1$ and $V_2$ of the network $K_{m,n}$ are the pixels of images 1 and 2, respectively. The set of edges $E_{12}$ contains a subset of all pixel pairs between the two images, as detailed further. We consider the cost of an edge $(i, j)$ as

$$C_{ij}(\theta,\tau) = \min\left\{(1-\theta)\|\mathrm{v}_i - \mathrm{v}_j\|_2 + \theta\|G_i - H_j\|_1, \tau\right\}, \tag{3}$$

where the vector $\mathrm{v}_i = (x_i, y_i)$ contains the horizontal and vertical coordinates of pixel $i$ of image 1 (similarly $\mathrm{v}_j$ for image 2). The quantity $\theta \in [0, 1]$ is a hyperparameter that is given in input and can be chosen with cross-validation. It acts as a weight for a convex combination between the Euclidean distance between pixels and the difference in their color intensities, following the intuition in [9, 23]. When $\theta = 0$, the OT path $P^\star$ is the one that minimizes only the geometrical distance between pixels. Instead, when $\theta = 1$, pixels' locations are no longer considered, and transport paths are only weighted by color distributions. The parameter $\tau$ is introduced following [22, 23] with the scope of removing all edges with cost $C_{ij}(\theta, \tau) = \tau$, i.e., those for which $(1-\theta)\|\mathrm{v}_i - \mathrm{v}_j\|_2 + \theta\|G_i - H_j\|_1 > \tau$. These are substituted by $m + n$ transshipment edges $e \in E'$, each of which has a cost of $\tau/2$ and is connected to one unique auxiliary vertex $u_1$. Thresholding the cost decreases significantly the computational complexity of OT, making it linear with the number of nodes $|V_1| + |V_2| + 2 = m + n + 2$ (see Supplementary Material).

Furthermore, we relax the conservation of mass by allowing $\sum_i G_{ia} \neq \sum_j H_{ja}$. The excess mass $m^a = \sum_j H_{ja} - \sum_i G_{ia}$ is assigned to a second auxiliary node, $u_2$. We connect it to the network with $n$ additional transshipment edges, $e \in E'$, each penalizing the total cost by $c = \max_{ij} C_{ij}/2$. This construction improves classification when the histograms' total masses largely differ [22]. Intuitively, this can happen when comparing "darker" images against "brighter" images more precisely, when entries of $g^a$ and $h^a$ are further apart in the RGB color space.

Overall, we obtain a network $K$ with nodes $V = V_1 \cup V_2 \cup \{u_1, u_2\}$ and edges $E = E_{12} \cup E'$, i.e., the original bipartite graph $K_{m,n}$, together with the auxiliary transshipment links and nodes. It should be noted that in its entirety, the system is isolated, i.e., the total mass is conserved. See Supplementary Material for a detailed description of the OT setup.

Given this auxiliary graph, the OT problem is then solved by injecting the color mass contained in image 1 in nodes $i \in V_1$, as specified by $G$, and extracting it from nodes $j \in V_2$ of image 2, as specified by $H$. This is carried out by transporting mass using either i) an edge in $E_{12}$ or ii) a transshipment one in $E'$. In the following section, we describe how this problem is solved mathematically.

## 3.2 Optimizing immiscible color flows: The dynamics

We solve the OT problem by proposing the following ODEs for controlling mass transportation:

$$\sum_{j \in \partial i} L_{ij}[x]\phi_j^a = S_i^a \quad \forall i \in V, \quad a = 1, \ldots, M, \tag{4}$$

$$\frac{dx_e}{dt} = x_e^\beta \frac{\|\phi_i - \phi_j\|_2^2}{C_e^2} - x_e, \quad \forall e = (i, j) \in E, \tag{5}$$

which constitute the pivotal equations of our model. Here, we introduce the *shared conductivities* $x_e \geq 0$ and define $S_i^a = G_{ia} - H_{ia}$, taking values $S_{u_1}^a = 0$ and $S_{u_2}^a = m^a$ on the auxiliary nodes. With $L_{ij}[x] = \sum_e (x_e/C_e) B_{ie} B_{je}$, we denote the weighted Laplacian of $K$, where $B$ is its signed incidence matrix and $\partial i$ is the neighborhood of node $i$. Lastly, $\phi_i^a$ is the scalar potential acting on nodes for a given commodity $a$. The least-square solutions of Eq. 4 are $\phi_i^a[x] = \sum_j L^{\dagger}_{ij}[x] S_i^a$, where † denotes the Moore–Penrose inverse. The critical exponent $0 < \beta < 2$ [$\Gamma = 2(2-\beta)/(3-\beta)$] is a hyperparameter that needs to be chosen before solving Eqs 4 and 5. Depending on the modeling task, its value can be fixed *a priori* (e.g., $\beta = 1$ for the shortest path problem [34], $\beta \simeq 5/3$ for river networks [35], and $\beta \to 2^-$ for the Steiner tree problem [36]) or cross-validated as we do here for image classification. The exponent aggregates paths using the principle of economy of scale if $1 < \beta < 2$. It dilutes them along the network otherwise, with the goal of reducing traffic congestion. This behavior is a direct consequence of the subadditivity of $J_\Gamma$ in Eq. 2 for $\beta > 1$ ($\Gamma < 1$), and, respectively, superadditivity for $\beta < 1$ ($\Gamma > 1$). It has been theoretically discussed and empirically observed, for example, in [32, 37, 38].

The feedback mechanism of Eq. 5 defines multicommodity fluxes ($P_e^a$) that are admissible for the minimization problem introduced in Eq. 2. Particularly, for color of type $a$ on edges $e = (i, j)$, we couple potentials ($\phi_i^a$) that are the solutions of Eq. 4 and shared conductivities ($x_e$) to define

$$P_e^a(t) = x_e(t)\,\frac{\phi_i^a[x(t)] - \phi_j^a[x(t)]}{C_e}, \quad \forall e \in E, \quad a = 1, \ldots, M. \qquad (6)$$

This also highlights another physical interpretation; i.e., by interpreting the $\phi_i^a$ as pressure potentials, the fluxes are seen to arise from a difference in pressure between two nodes as in hydraulic or electrical networks. Crucially, this allocation is governed by *one unique conductivity* for all commodities, whose dynamics depends on the 2-norm over $a$ of differences in potentials, as in Eq. 5. In analogy with immiscible flows, this ensures that flows of different types share the same infrastructure, and in practice, it couples them into a unique optimization problem.

In the case of only one commodity ($M = 1$), variants of this dynamics have been used to model transport optimization in various physical systems [1, 5, 29–31].

The salient result of our construction is that the asymptotic trajectories of Eqs 4 and 5 are equivalent to the minimizers of Eq. 2, i.e., $\lim_{t\to+\infty} P(t) = P^{\star}$ (see Supplementary Material for derivations following [32, 33]). Therefore, numerically integrating our dynamics solves the multicommodity OT problem. In other words, this allows us to estimate the optimal cost in Eq. 2 and use that to compute similarities between images. A pseudo-code of the algorithmic implementation is shown in Algorithm 1.

## 3.3 Computational complexity

In principle, our multicommodity method has a computational complexity of order $O(M|V|^2)$ for complete transport network topologies, i.e., when edges in the transport network $K$ are assigned to all pixel pairs. Nonetheless, we substantially reduce this complexity to $O(M|V|)$ by sparsifying the graph with the trimming procedure of [22, 23]. More details are given in Supplementary Material. Empirically, we observe that by running Eqs 4 and 5, most of the entries of $x$ decay to zero after a few steps, producing a progressively sparser weighted Laplacian $L[x]$. This allows for faster computation of the Moore–Penrose inverse $L^{\dagger}[x]$ and least-square potentials $\phi_i^a = \sum_j L^{\dagger}_{ij}[x] S_j^a$. A thorough experimental analysis of the convergence properties of the OT dynamics has been carried out in [39].

# 4 Results and discussion

## 4.1 Classification task

We provide empirical evidence that our multicommodity dynamics outperforms competing OT algorithms on classification tasks. As anticipated previously, we use the OT optimal cost $J_\Gamma^{\star}$ as a measure of similarity between two images and perform supervised classification with a $k$-nearest neighbor ($k$-NN) classifier as described in [20]. Alternative methods (e.g., SVM as in [19]) could also be used for this task. However, these may require the cost $J_\Gamma^{\star}$ to satisfy the distance axioms to properly induce a kernel. While it is not straightforward to verify these conditions for the OT cost in Eq. 2, this is not necessary for the $k$-NN classifier, which requires looser conditions on $J_\Gamma^{\star}$.

We compare the classification accuracy of our model against i) the Sinkhorn algorithm [19, 40] (utilizing the more stable Sinkhorn scheme proposed in [41]); ii) a unicommodity dynamics executed on grayscale images, i.e., with color information compressed into one single commodity ($M = 1$); and iii) the Sinkhorn algorithm on grayscale images. All methods are tested on the following two datasets: the Jena Flowers 30 Dataset (JF30) [42] and the Fruit Dataset (FD) [43]. The first consists of 1,479 images of 30 wild-flowering angiosperms (flowers). Flowers are labeled with their species, and inferring them is the goal of the classification task. The second dataset contains 15 fruit types and 163 images. Here, we want to classify fruit types. The parameters of

```
Input: Image 1 ($G \in \mathbb{R}^{m\times M}$), Image 2 ($H \in \mathbb{R}^{n\times M}$), $0 \le \theta \le 1$, $\tau \ge 0$, $0 \le \beta \le 2$
Initialize: $x(0) = \overline{x} > 0$                                    ▷ e.g. $\overline{x}_e \sim U(0,1)$
Construct a bipartite network $K_{m,n}$ between $G$ and $H$              ▷ complexity $O(m \cdot n)$
Assign $C_{ij}(\theta,\tau) = \min\{(1-\theta)\,||v_i - v_j||_2 + \theta\,||G_i - H_j||_1, \tau\}$ to every edge $(i,j)$ in $K_{m,n}$, as in
    Eq. (3)
Remove from $K_{m,n}$ all edges s.t. $C_{ij} > \tau$                     ▷ complexity $O(m+n)$
Add $u_1$ to $K_{m,n}$ and it $m+n$ auxiliary links, each costing $\tau/2$
Balance mass: add $u_2$, with inflowing mass $m^a = \sum_i H_{ia} - \sum_j G_{ja}$
while convergence is False do
    Solve Kirchhoff's law, Eq. (4) $\to \phi \in \mathbb{R}^{|V|\times M}$
    Update $x$ with discretization of Eq. (5)
end while
Compute $P$ as in Eq. (6)
Return: $J_\Gamma^{\star}(G,H)$ as in Eq. (2)
```

Algorithm 1. Multicommodity dynamics.

**TABLE 1 Classification task results. With multicommodity, Sinkhorn RGB, unicommodity, and Sinkhorn GS, we label methods on colored images (the first two) and grayscale images (the second two). The optimal parameters in the central columns are selected with a 4-fold cross-validation; $k$ is the number of nearest neighbors used in the classifier. The rightmost column shows the fraction (in percentage) of correctly classified images. Results are ordered by performance, and we highlight the best ones in bold.**

| | Algorithm | Hyperparameters | | | | | Class accuracy |
|---|---|---|---|---|---|---|---|
| | | $\theta$ | $\tau$ | $\beta$ | $\varepsilon$ | $k$ | [%] (↑) |
| JF30 | Multicommodity | 0.25 | 0.125 | 1 | — | 1 | 62.2 |
| | Sinkhorn RGB | 0.25 | 0.05 | — | 100 | 1 | 58.4 |
| | Sinkhorn GS | 0.25 | 0.05 | — | 500 | 1 | 54.3 |
| | Unicommodity | 0.25 | 0.125 | 1.25 | — | 1 | 53.6 |
| FD | Multicommodity | 0 | 0.04 | 1.5 | — | 2 | 75.0 |
| | Sinkhorn RGB | 0.5 | 0.06 | — | 750 | 1 | 69.6 |
| | Unicommodity | 0 | 0.06 | 1.5 | — | 5 | 64.3 |
| | Sinkhorn GS | 0.25 | 0.06 | — | 500 | 4 | 60.7 |

the OT problem setup ($\theta$ and $\tau$) and regularization parameters ($\beta$ and $\varepsilon$, which enforce the entropic barrier in the Sinkhorn algorithm [19]), have been cross-validated for both datasets (see Section 3 and Section 4 in Supplementary Material). All methods are then tested in their optimal configurations (see Supplementary Material for implementation details).

Classification results are shown in Table 1. In all cases, leveraging colors leads to higher accuracy (about an 8% increase) with respect to classification performed using grayscale images. This signals that in the datasets under consideration, color information is a relevant feature for differentiating image samples. Remarkably, we get a similar increase in performance (about 7%–8%) on both colored datasets when comparing our multicommodity dynamics against the Sinkhorn algorithm. As the two algorithms use the same (colored) input, we can attribute this increment to the effective usage of color that our approach is capable of.

In addition, by analyzing results in more detail, we first observe that on JF30, all methods perform best when $\theta = 0.25$, i.e., 25% of the information used to build $C$ comes from colors. This trend does not recur on the FD, where both dynamics favor $\theta = 0$ (Euclidean $C$). Hence, our model is able to leverage color information *via* the multicommodity OT dynamical formulation.

Second, on JF30, both dynamics perform best with $\tau = 0.125$, contrary to Sinkhorn-based methods that prefer $\tau = 0.05$. Thus, Sinkhorn's classification accuracy is negatively affected both by low $\tau$—many edges of the transport network are cut—and by large $\tau$—noisy color information is used to build $C$. We do not observe this behavior in our model, where trimming fewer edges is advantageous. All optimal values of $\tau$ are lower on the FD since the color distributions in this dataset are naturally light-tailed (see Supplementary Material).

Lastly, we investigate the interplay between $\theta$ and $\beta$. We notice that $\theta = 0$ (FD) corresponds to higher $\beta = 1.5$. Instead, for larger $\theta = 0.25$ (JF30), the model prefers lower $\beta$ ($\beta = 1$ and 1.25 for the multicommodity and unicommodity dynamics, respectively). In the former case ($\theta = 0$, $C_{ij}$ is the Euclidean distance), the cost is equal to zero for pixels with the same locations. Thus, consolidation of transport paths—large $\beta$—is favored on cheap links. Instead, increasing $\theta$ leads to more edges with comparable costs as colors distribute smoothly over images. In this second scenario, better performance is achieved with distributed transport paths, i.e., lower $\beta$ (see Supplementary Material).

## 4.2 Performance in terms of sensitivity

We assess the effectiveness of our method against benchmarks by comparing the sensitivity of our multicommodity dynamics and that of the Sinkhorn algorithm on the colored JF30 dataset. Specifically, we set all algorithm parameters to their best configurations, as shown in Table 1. Then, for each of the 30 classes in JF30, we compute its one-to-all sensitivity, i.e., the true positive rate. This is defined for any class $c$ as

$$S(c) = \frac{\mathrm{TP}(c)}{\mathrm{TP}(c) + \mathrm{FN}(c)}, \tag{7}$$

where TP($c$) is the true positive rate, i.e., the number of images in $c$ that are correctly classified; FN($c$) is the false negative rate, i.e., the number of $c$-samples that are assigned a label different from $c$. Hence, Eq. 7 returns the probability that a sample is assigned label $c$, given that it belongs to $c$.

We find that our method robustly outperforms the Sinkhorn algorithm. Specifically, the multicommodity dynamics has the highest sensitivity 50% of the times—15 classes out of a total of 30—as shown in Figure 2. For nine classes, Sinkhorn has higher sensitivity, and for six classes, both methods give the same values of $S$.Furthermore, we find that in 2/3 (20 out of 30) of the classes, the multicommodity dynamics returns $S(c) \geq 1/2$. This means that our model predicts the correct label more than 50% of the time. In only three out of these 20 cases, Sinkhorn attains higher values of $S$, while in most instances where Sinkhorn outperforms our method, it has a lower sensitivity of $S < 1/2$. Hence, this is the case in classes where both methods have difficulty distinguishing images.

## 4.3 The impact of colors

To further assess the significance of leveraging color information, we conduct three different experiments that highlight both qualitatively and quantitatively various performance differences between the unicommodity and multicommodity approaches. As the two share the same principled dynamics based on OT with the main difference being that multicommodity does not compress the color information, we can use this analysis to better understand how fully exploiting the color information drives better classification.

*Experiment 1: Landscape of optimal cost.* Here, we focus on a qualitative comparison between the cost landscapes obtained with the two approaches. We consider the example of an individual image taken from the FD test set and plot the landscape of optimal costs $J_{\Gamma}^{\star}$ when comparing it to the train set. Results for the multicommodity dynamics ($M = 3$) and the unicommodity dynamics ($M = 1$) on grayscale images are shown in Figure 3. Here, we highlight the five lowest values of the cost and mark them in green if they correspond to correctly classified train samples and in red otherwise. At first glance, one may conclude that their performance is identical (as both dynamics classify correctly three samples out of five), and we notice how the multicommodity dynamics consistently clusters them at the bottom of the cost landscape, thus ranking them in a better order. This may explain why the cross-validated best value of $k$ (the number of nearest neighbors in the $k$-NN classifier) is higher for unicommodity methods in this dataset. On a larger sample of data, this results in better overall classification performance, as shown in Table 1.

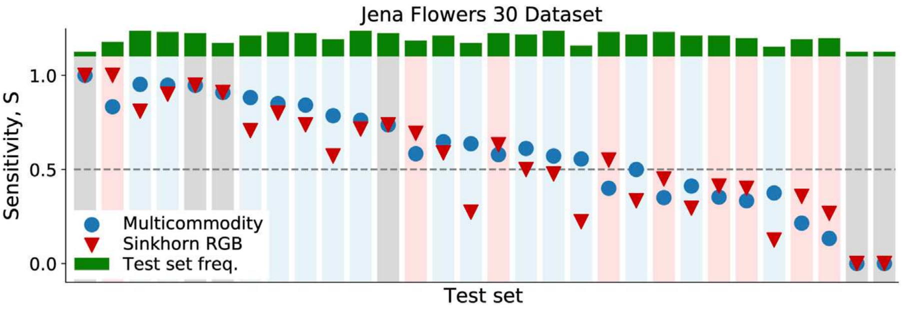


FIGURE 2
Sensitivity on the JF30 dataset. Sensitivity values are shown for the multicommodity dynamics (blue circles) and for Sinkhorn RGB (red triangles). Markers are sorted in descending order of *S*, regardless of the method. Background colors are blue, red, and gray, when *S* is higher for the multicommodity method, the Sinkhorn algorithm, or none of them, respectively. In green, we plot frequency bars for all classes in the test set.

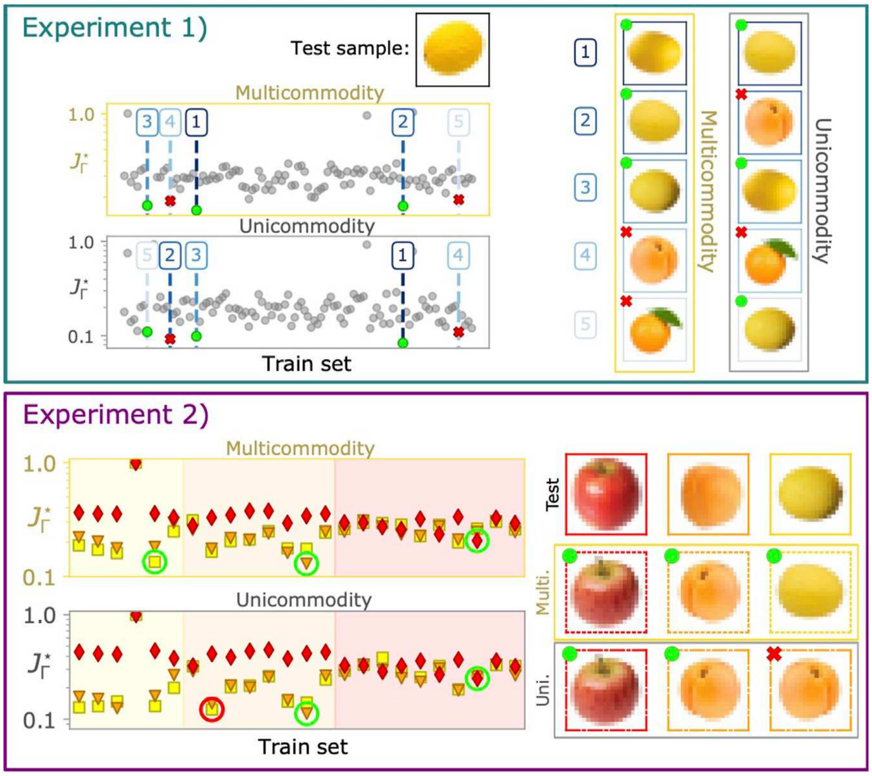


FIGURE 3
Evaluating the effect of colors. Experiment 1: The top black-framed image is the one to be classified. Predictions given by the multicommodity and unicommodity dynamics (those with lower $J_\Gamma^\star$) are shown on the right side of the panel and are displayed in a sorted fashion from worst to best (from bottom to top). Experiment 2: The top right samples are the three test images to be classified. Middle and bottom rows are predictions given by the two dynamics. Markers, backgrounds, and test images shared a color code: red for apples, orange for apricots, and yellow for melons. In both panels, green circles and red crosses are used to highlight classified and misclassified images, respectively. All algorithms are executed with their optimal configurations listed in Table 1.

*Experiment 2: Controlling for shape.* We further mark this tendency with a second experiment where we select a subset of the FD composed of images belonging to three classes of fruits that have similar shapes but different colors such as red apples, orange apricots, and yellow melons. As we expect shape to be less informative than colors in this custom set, we can assess the extent to which color plays a crucial role in the classification process. Specifically, the test set is made of three random samples, each drawn from one of these classes (top row of the rightmost panel) in Figure 3, while the train set contains the remaining instances of the classes. We plot the cost landscape $J_\Gamma^\star$ for the train set and draw in the red, orange, and yellow values of $J_\Gamma^\star$ that correspond to the samples that are compared against the test apple, apricot, and melon, respectively. We also sort the train samples so that they are grouped in three regions (highlighted by the background color in Figure 3), which correspond to train melons, apricots, and apples. With this construction, if the minimum cost among the yellow markers falls in the yellow region, it

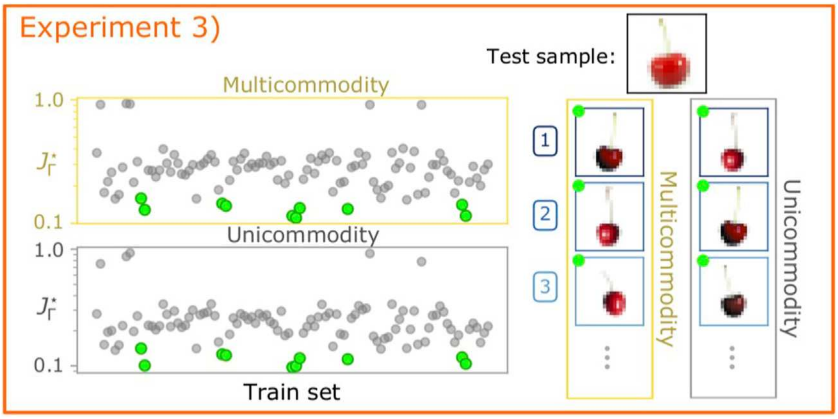


FIGURE 4
Evaluating the importance of colors: when shapes matter most. Experiment 3: The top black-framed image is the one to be classified. The best three (out of 10) predictions returned by the two dynamics are shown on the right. We mark the training samples belonging to the same class as the test image with green circles. All algorithms are executed with their optimal configurations listed in Table 1.

will correspond to a correctly classified sample (respectively, for orange and red). We further mark the yellow, orange, and red minima in green if the test and train labels correspond, i.e., the marker's and background colors are the same, and in red otherwise. Train and test samples are also in Figure 3. The multicommodity dynamics correctly label each test image. In contrast, unicommodity dynamics fails at this task, labeling a melon as an apricot. This suggests that the multicommodity approach is able to use the color information in datasets where this feature is more informative than others, e.g., shape.

*Experiment 3: When shape matters.* Having shown results on a custom dataset where shape was controlled to matter less, we now do the opposite and select a dataset where this feature should be more informative. The goal is to assess whether a multicommodity approach helps in this case as well, as its main input information may not be as relevant anymore. Specifically, we select as a test sample a cherry, whose form is arguably distinguishable from that of many other fruits in the dataset. One can expect that comparing it against the train set of the FD will result in having both unicommodity and multicommodity dynamics able to assign low $J_\Gamma^\star$ to train cherries and higher costs to other fruits. This intuition is confirmed by the results in Figure 4. Here, train cherries (in green) strongly cluster in the lower portion of the cost landscape, whereas all the other fruits have higher costs. In Figure 4, we also plot some of the correctly classified train samples. These results suggest that when color information is negligible compared to another type of information (e.g., shape), unicommodity and multicommodity formulations perform similarly. In light of this, we reinforce the claim that our multicommodity formulation can boost classification in contexts where color information does matter but may not give any advantage when other types of information are more informative. We encourage practitioners to evaluate when this is the case based on domain knowledge when available.

## 5 Conclusion

We propose a physics-informed multicommodity OT formulation for effectively using color information to improve image classification. We model colors as immiscible flows traveling on a capacitated network and propose equations for its dynamics, with the goal of optimizing flow distribution on edges. Color flows are regulated by a shared conductivity to minimize a unique cost function. Thresholding the ground cost as in [22, 23] makes our model computationally efficient.

We outperform other OT-based approaches such as the Sinkhorn algorithm on two datasets where color matters. Our model also assigns a lower cost to correctly classified images than its unicommodity counterpart, and it is more robust on datasets where items have similar shape. Thus, color information is distinctly relevant. We note that for some datasets, color information may not matter as much as another type of information (e.g., shape), which has stronger discriminative power. However, while we focused here on different color channels as the different commodities in our formulation, the ideas of this study can be extended to scenarios where other relevant information can be distinguished into different types. For instance, one could combine several features together, e.g., colors, contours, and objects' orientations when available.

Our model can be further improved. While it uses the thresholding of [22, 23] to speed up convergence (as mentioned in Section 3.1), it is still slower than Sinkhorn-based methods. Hence, investigating approaches aimed at improving its computational performance is an important direction for future work. Speed-up can be achieved, for example, with the implementation of [39], where the unicommodity OT problem on sparse topologies is solved in $O(|E|^{0.36})$ time steps. This bound has been found using a backward Euler scheme combined with the inexact Newton–Raphson method for the update of $x$ and solving Kirchhoff's law using an algebraic multigrid method [44].

Our main goal is to frame an image classification task into that of finding optimal flows of masses of different types in networks built from images. We follow physics principles to assess whether using colors as immiscible flows can give an advantage compared to other standard OT-based methods that do not incorporate such insights. The increased classification performance observed in our experiments stimulates the integration of similar ideas into deep network architectures [45] as a relevant avenue for future work. Combining their prediction capabilities with our insights on how to better exploit the various facets of the input data has the potential to push the performance of deep classifiers even further. For example, one could extend the state-of-the-art architecture of Eisenberger et al. [45], which efficiently computes implicit gradients for generic Sinkhorn layers within a neural network, by including edge, shape, and contour information for Wasserstein barycenter computation or image clustering.

## Data availability statement

The original contributions presented in the study are publicly available. This data can be found here: https://doi.org/10.7910/DVN/QDHYST, https://github.com/daniloeler/fruitdataset.

## Author contributions

All authors contributed to developing the models, conceiving the experiments, analyzing the results, and reviewing the manuscript. AL and DB conducted the experiments. All authors read and agreed to the published version of the manuscript.

## Acknowledgments

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting AL and DB.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphy.2023.1089114/full#supplementary-material

## References

1. Kaiser F, Ronellenfitsch H, Witthaut D. Discontinuous transition to loop formation in optimal supply networks. *Nat Commun* (2020) 11:5796–11. doi:10.1038/s41467-020-19567-2

2. Lonardi A, Putti M, De Bacco C. Multicommodity routing optimization for engineering networks. *Scientific Rep* (2022) 12:7474. doi:10.1038/s41598-022-11348-9

3. Lonardi A, Facca E, Putti M, De Bacco C. *Infrastructure adaptation and emergence of loops in network routing with time-dependent loads*. arXiv (2021). doi:10.48550/ARXIV.2112.10620

4. Demetci P, Santorella R, Sandstede B, Noble WS, Singh R. *Gromov-Wasserstein optimal transport to align single-cell multi-omics data*. bioRxiv (2020). doi:10.1101/2020.04.28.066787

5. Katifori E, Szöllősi GJ, Magnasco MO. Damage and fluctuations induce loops in optimal transport networks. *Phys Rev Lett* (2010) 104:048704. doi:10.1103/PhysRevLett.104.048704

6. Werman M, Peleg S, Rosenfeld A. A distance metric for multidimensional histograms. *Comput Vis Graphics, Image Process* (1985) 32:328–36. doi:10.1016/0734-189X(85)90055-6

7. Peleg S, Werman M, Rom H. A unified approach to the change of resolution: Space and gray-level. *IEEE Trans Pattern Anal Machine Intelligence* (1989) 11:739–42. doi:10.1109/34.192468

8. Rubner Y, Tomasi C, Guibas LJ. A metric for distributions with applications to image databases. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271). Bombay, India: IEEE (1998). p. 59–66. doi:10.1109/ICCV.1998.710701

9. Rubner Y, Tomasi C, Guibas LJ. The earth mover's distance as a metric for image retrieval. *Int J Comput Vis* (2000) 40:99–121. doi:10.1023/A:1026543900054

10. Baptista D, De Bacco C. Principled network extraction from images. *R Soc Open Sci* (2021) 8:210025. doi:10.1098/rsos.210025

11. Peyré G, Cuturi M. Computational optimal transport: With applications to data science. *Foundations Trends® Machine Learn* (2019) 11:355–607. doi:10.1561/2200000073

12. Koehl P, Delarue M, Orland H. Optimal transport at finite temperature. *Phys Rev E* (2019) 100:013310. doi:10.1103/PhysRevE.100.013310

13. Aurell E, Mejía-Monasterio C, Muratore-Ginanneschi P. Optimal protocols and optimal transport in stochastic thermodynamics. *Phys Rev Lett* (2011) 106:250601. doi:10.1103/PhysRevLett.106.250601

14. Leite D, De Bacco C. *Revealing the similarity between urban transportation networks and optimal transport-based infrastructures. arXiv preprint arXiv:2209.06751* (2022).

15. Baptista D, Leite D, Facca E, Putti M, De Bacco C. Network extraction by routing optimization. *Scientific Rep* (2020) 10:20806. doi:10.1038/s41598-020-77064-4

16. Ibrahim AA, Lonardi A, Bacco CD. Optimal transport in multilayer networks for traffic flow optimization. *Algorithms* (2021) 14:189. doi:10.3390/a14070189

17. Mondino A, Suhr S. An optimal transport formulation of the Einstein equations of general relativity. *J Eur Math Soc* (2022). doi:10.4171/JEMS/1188

18. Grauman K, Darrell T. Fast contour matching using approximate Earth mover's distance. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004; 27 June 2004 - 02 July 2004. Washington, DC, USA: IEEE (2004). doi:10.1109/CVPR.2004.1315035

19. Cuturi M. Sinkhorn distances: Lightspeed computation of optimal transport. In: *Advances in neural information processing systems*, Vol. 26. Red Hook, NY, USA: Curran Associates, Inc. (2013). p. 2292–300.

20. Koehl P, Delarue M, Orland H. Statistical physics approach to the optimal transport problem. *Phys Rev Lett* (2019) 123:040603. doi:10.1103/PhysRevLett.123.040603

21. Thorpe M, Park S, Kolouri S, Rohde GK, Slepčev D. A transportation $L^p$ distance for signal analysis. *J Math Imaging Vis* (2017) 59:187–210. doi:10.1007/s10851-017-0726-4

22. Pele O, Werman M. A linear time histogram metric for improved SIFT matching. In: *Computer vision – ECCV 2008*. Berlin, Heidelberg: Springer Berlin Heidelberg (2008). p. 495–508. doi:10.1007/978-3-540-88690-7_37

23. Pele O, Werman M. Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision. Kyoto, Japan: IEEE (2009). p. 460–7. doi:10.1109/ICCV.2009.5459199

24. Villani C. In: *Optimal transport: Old and new*, Vol. 338. Berlin, Heidelberg: Springer (2009). doi:10.1007/978-3-540-71050-9

25. Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: D Precup YW Teh, editors. *Proceedings of the 34th international conference on machine learning (PMLR). Proceedings of machine learning research*, Vol. 70 (2017). p. 214–23.

26. Lin T, Ho N, Jordan M. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In: *Proceedings of the 36th international conference on machine learning (PMLR)*. In: *Proceedings of machine learning research*, Vol. 97 (2019). p. 3982–91.

27. Dvurechensky P, Gasnikov A, Kroshnin A. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In: *Proceedings of the 35th international conference on machine learning (PMLR)*. In: *Proceedings of machine learning research*, Vol. 80 (2018). p. 1367–76.

28. Banavar JR, Colaiori F, Flammini A, Maritan A, Rinaldo A. Topology of the fittest transportation network. *Phys Rev Lett* (2000) 84:4745–8. doi:10.1103/PhysRevLett.84.4745

29. Ronellenfitsch H, Katifori E. Global optimization, local adaptation, and the role of growth in distribution networks. *Phys Rev Lett* (2016) 117:138301. doi:10.1103/PhysRevLett.117.138301

30. Hu D, Cai D. Adaptation and optimization of biological transport networks. *Phys Rev Lett* (2013) 111:138701. doi:10.1103/PhysRevLett.111.138701

31. Corson F. Fluctuations and redundancy in optimal transport networks. *Phys Rev Lett* (2010) 104:048703. doi:10.1103/PhysRevLett.104.048703

32. Lonardi A, Facca E, Putti M, De Bacco C. Designing optimal networks for multicommodity transport problem. *Phys Rev Res* (2021) 3:043010. doi:10.1103/PhysRevResearch.3.043010

33. Bonifaci V, Facca E, Folz F, Karrenbauer A, Kolev P, Mehlhorn K, et al. Physarum-inspired multi-commodity flow dynamics. *Theor Comput Sci* (2022) 920:1–20. doi:10.1016/j.tcs.2022.02.001

34. Bonifaci V, Mehlhorn K, Varma G. Physarum can compute shortest paths. *J Theor Biol* (2012) 309:121–33. doi:10.1016/j.jtbi.2012.06.017

35. Rinaldo A, Rodriguez-Iturbe I, Rigon R, Ijjasz-Vasquez E, Bras RL. Self-organized fractal river networks. *Phys Rev Lett* (1993) 70:822–5. doi:10.1103/PhysRevLett.70.822

36. Barabási AL. Invasion percolation and global optimization. *Phys Rev Lett* (1996) 76: 3750–3. doi:10.1103/PhysRevLett.76.3750

37. Santambrogio F. Optimal channel networks, landscape function and branched transport. *Inter Free Boundaries* (2007) 9:149–69. doi:10.4171/IFB/160

38. Ibrahim AA, Leite D, De Bacco C. Sustainable optimal transport in multilayer networks. *Phys Rev E* (2022) 105:064302. doi:10.1103/PhysRevE.105.064302

39. Facca E, Benzi M. Fast iterative solution of the optimal transport problem on graphs. *SIAM J Scientific Comput* (2021) 43:A2295–A2319. doi:10.1137/20M137015X

40. Flamary R, Courty N, Gramfort A, Alaya MZ, Boisbunon A, Chambon S, et al. POT: Python optimal transport. *J Machine Learn Res* (2021) 22:1–8.

41. Schmitzer B. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM J Scientific Comput* (2019) 41:A1443–A1481. doi:10.1137/16M1106018

42. Seeland M, Rzanny M, Alaqraa N, Wäldchen J, Mäder P. *Jena Flowers 30 dataset* (2017). doi:10.7910/DVN/QDHYST

43. Macanhã PA, Eler DM, Garcia RE, Junior WEM. Handwritten feature descriptor methods applied to fruit classification. In: *Information Technology - new generations*. Cham: Springer International Publishing (2018). p. 699–705. doi:10.1007/978-3-319-54978-1_87

44. Trottenberg U, Oosterlee CW, Schuller A. *Multigrid*. Amsterdam, Netherlands: Elsevier (2000).

45. Eisenberger M, Toker A, Leal-Taixé L, Bernard F, Cremers D. A unified framework for implicit sinkhorn differentiation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022). p. 509–18.

46. *MODI (Open Source code implementation)* (2022). Available from: https://github.com/aleable/MODI.