NEURAL INFORMATION PROCESSING SYSTEMS

UNIVERSITY OF HELSINKI

YaiYai

A?

Going Beyond Persistent Homology Using Persistent Homology

Johanna Immonen

University of Helsinki

Amauri Souza

Aalto University
Federal Institute of Ceará

Vikas Garg

Aalto University
YaiYai Ltd

Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)

Persistent homology (PH)

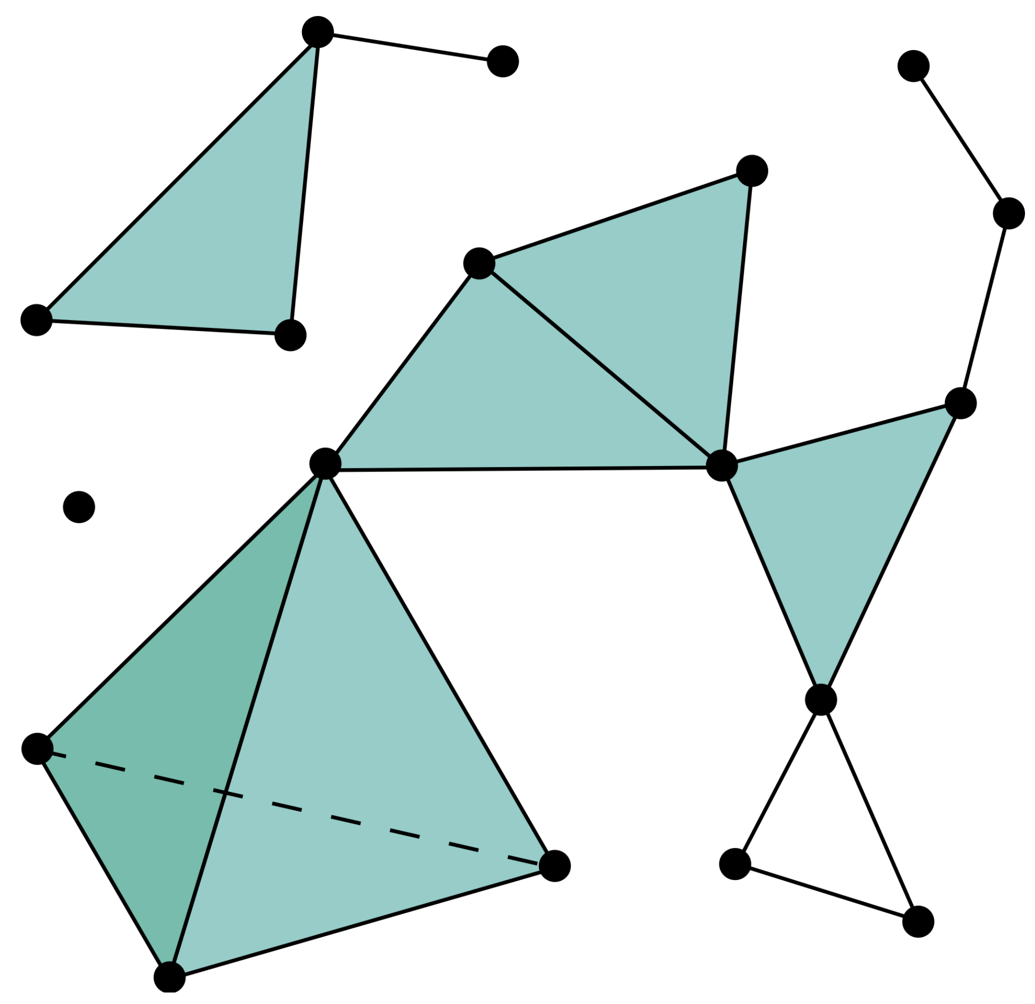
An approach to extract detailed **topological features** (e.g., persistence of connected components or cycles) of simplicial complexes (e.g., **graphs**).

Basic idea:

1) Obtain a **filtration** (i.e., sequence of sub-complexes) by applying a filtering function on simplices (elements of the original complex);
2) Keep track of the appearance (birth) and disappearance (death) of topological features, obtaining the so-called **persistence diagrams**.

Among other applications, PH has been successfully employed as a feature extractor in many disciplines, such as Astrophysics, Computer Vision, and Bioinformatics.

A simplicial complex.



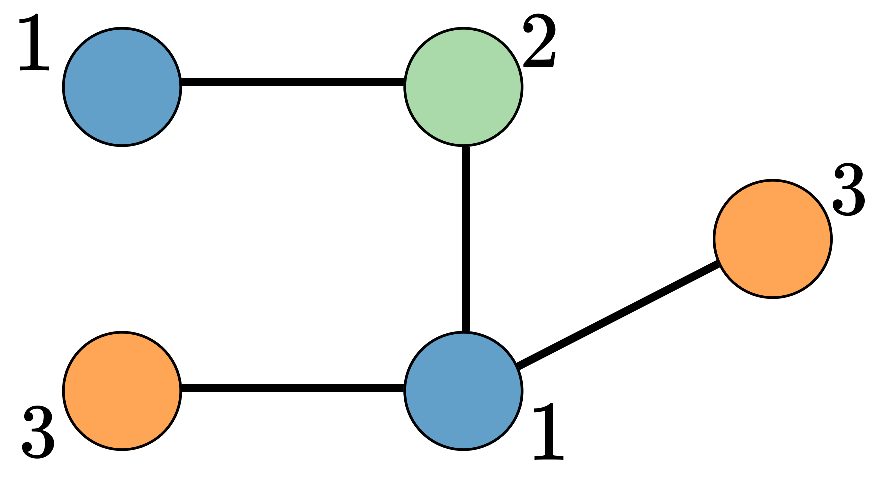
Source: Wikipedia

Persistent homology on graphs

Colors/features

Vertex-color Filtrations: Nested sequence of subgraphs $\varnothing = G^{(0)} \subseteq G^{(1)} \subseteq \ldots \subseteq G$ induced by $f : X \to (0,\infty)$

Attributed graph



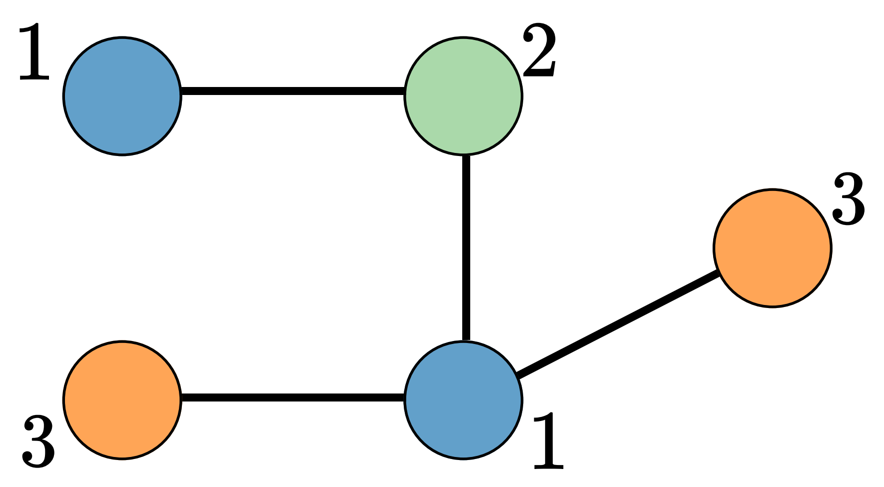
Persistent homology on graphs

Colors/features

Vertex-color Filtrations: Nested sequence of subgraphs $\varnothing = G^{(0)} \subseteq G^{(1)} \subseteq \ldots \subseteq G$ induced by $f : X \to (0,\infty)$

Filtration induced by: $\mapsto 1$ $\mapsto 2$ $\mapsto 3$

Attributed graph



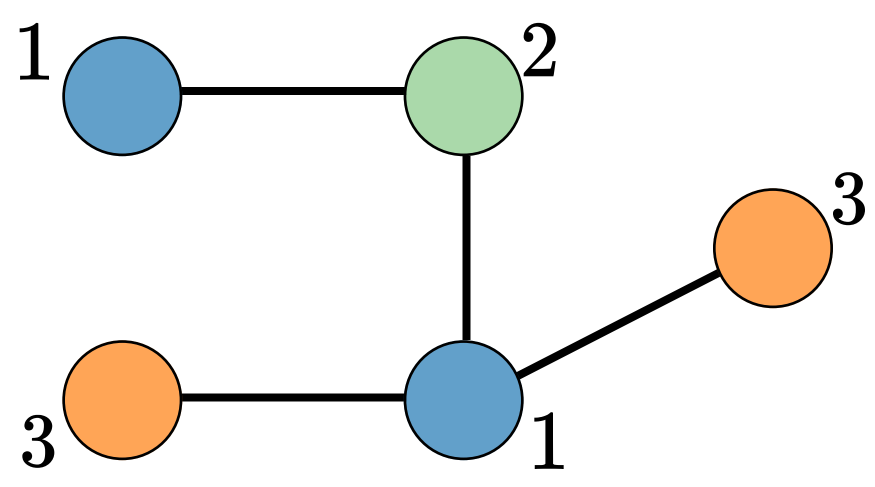
Persistent homology on graphs

Colors/features

Vertex-color Filtrations: Nested sequence of subgraphs $\varnothing = G^{(0)} \subseteq G^{(1)} \subseteq \ldots \subseteq G$ induced by $f : X \to (0,\infty)$

Filtration induced by: $\mapsto 1$ $\mapsto 2$ $\mapsto 3$

Attributed graph

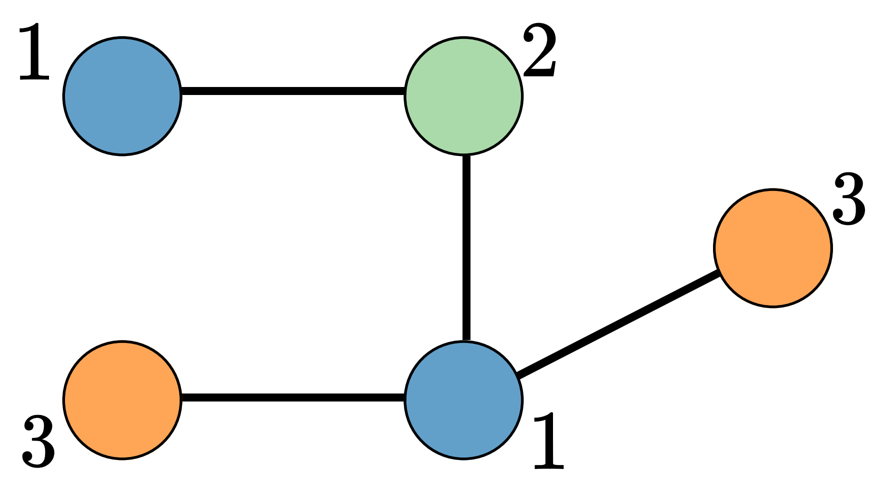


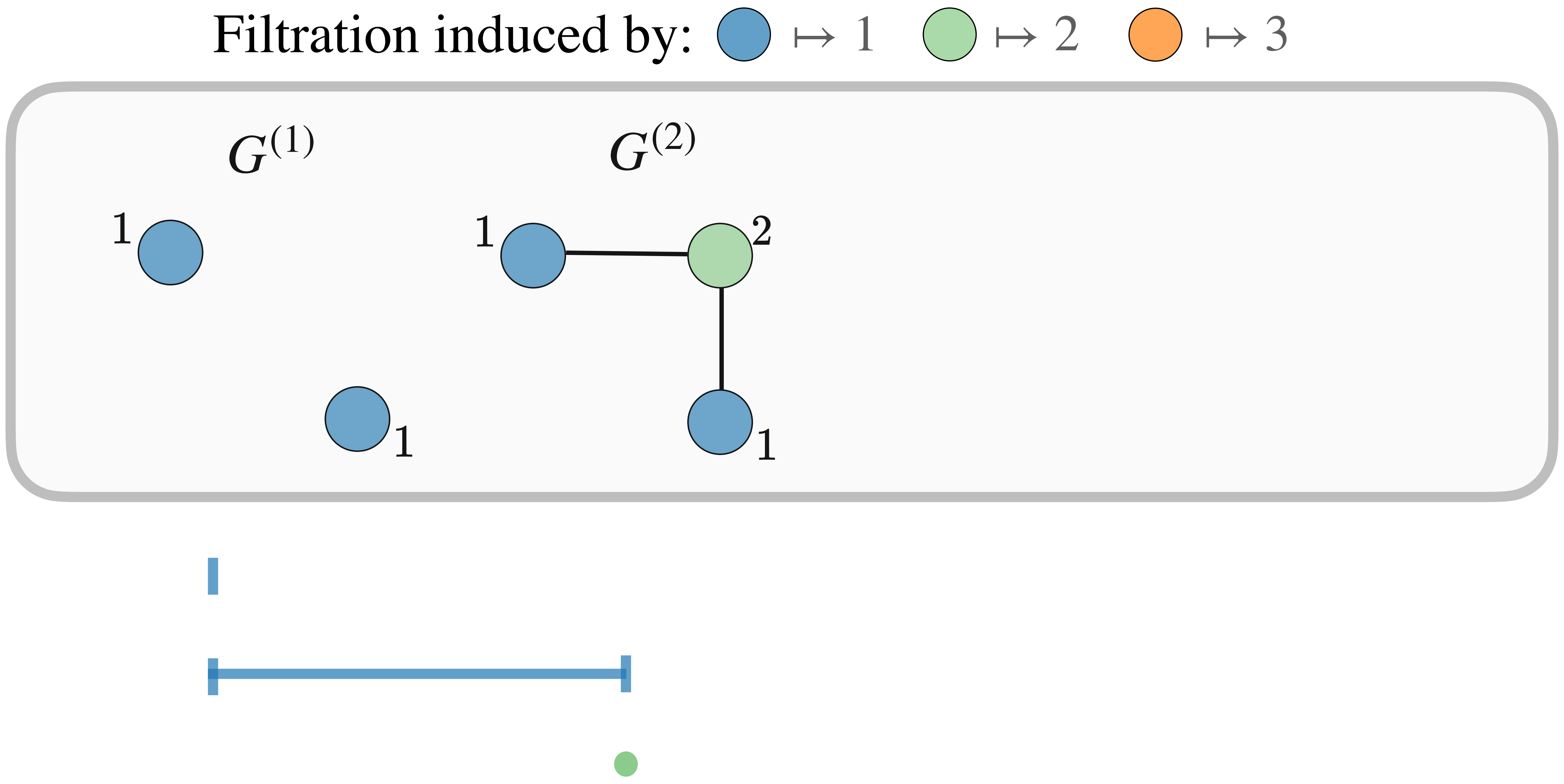
Persistent homology on graphs

Colors/features

Vertex-color Filtrations: Nested sequence of subgraphs $\varnothing = G^{(0)} \subseteq G^{(1)} \subseteq \ldots \subseteq G$ induced by $f : X \to (0,\infty)$

Attributed graph





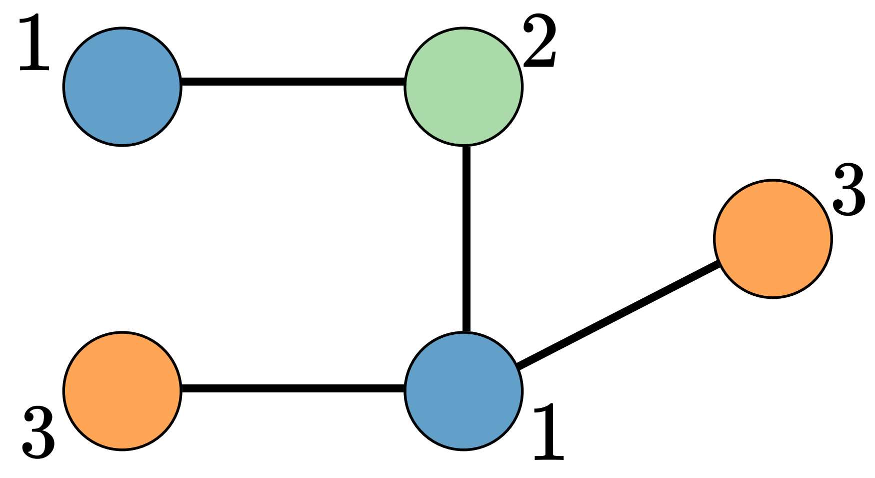
Persistent homology on graphs

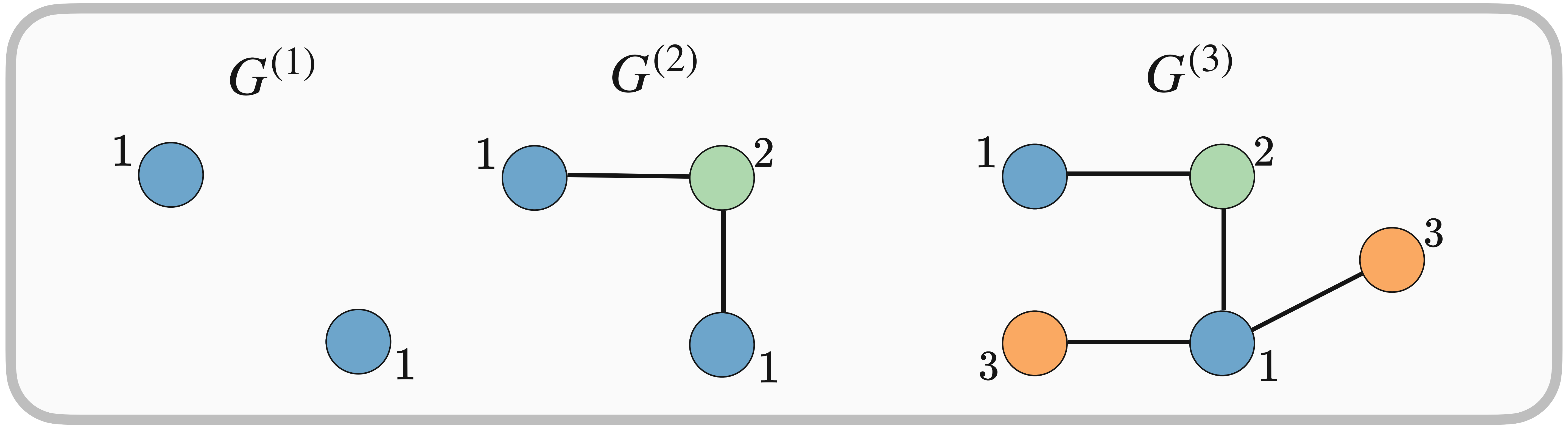
Colors/features

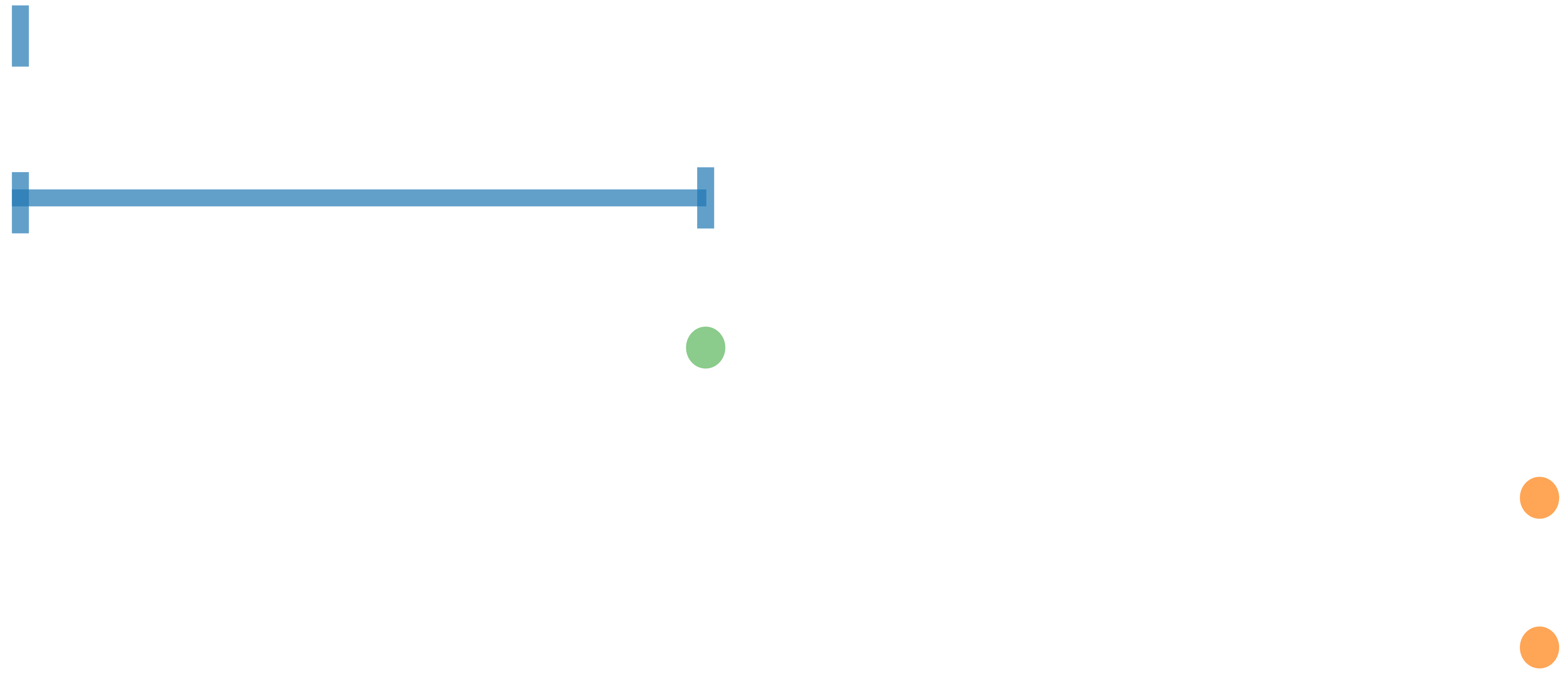
Vertex-color Filtrations: Nested sequence of subgraphs $\varnothing = G^{(0)} \subseteq G^{(1)} \subseteq \ldots \subseteq G$ induced by $f : X \to (0,\infty)$

Filtration induced by: $\mapsto 1$ $\mapsto 2$ $\mapsto 3$

Attributed graph



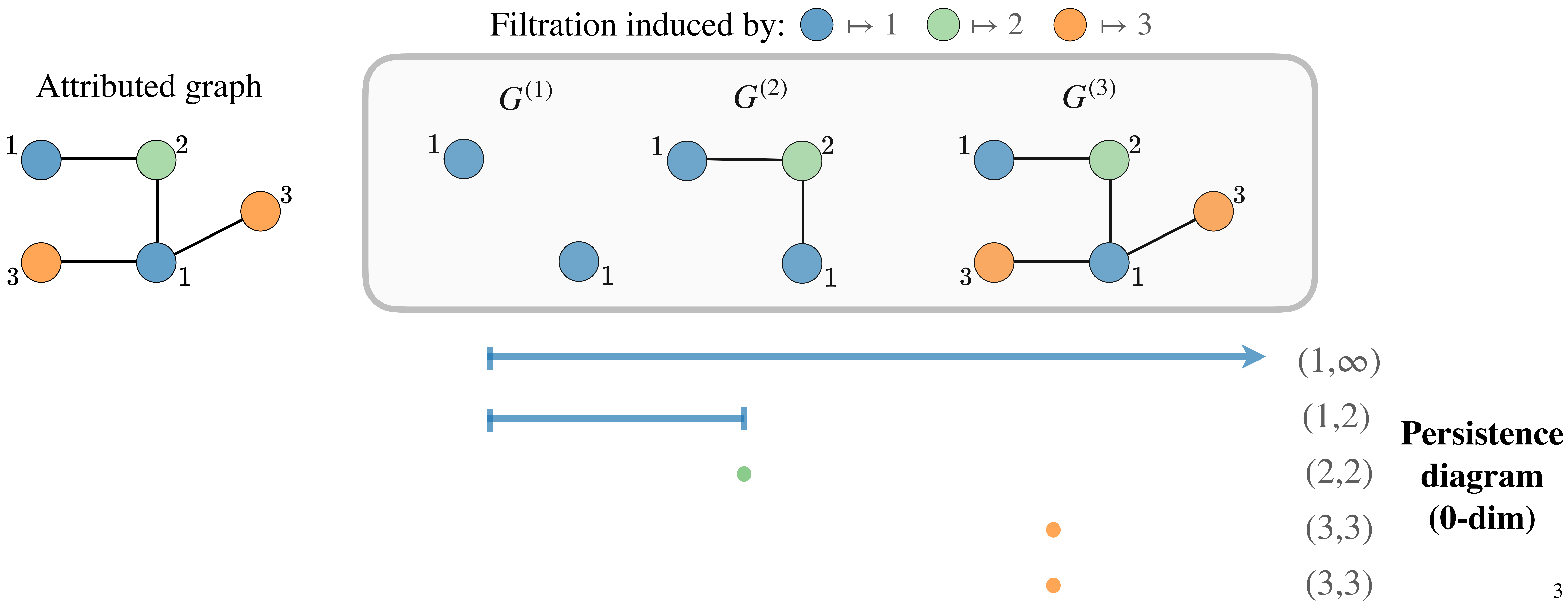




Persistent homology on graphs

Colors/features

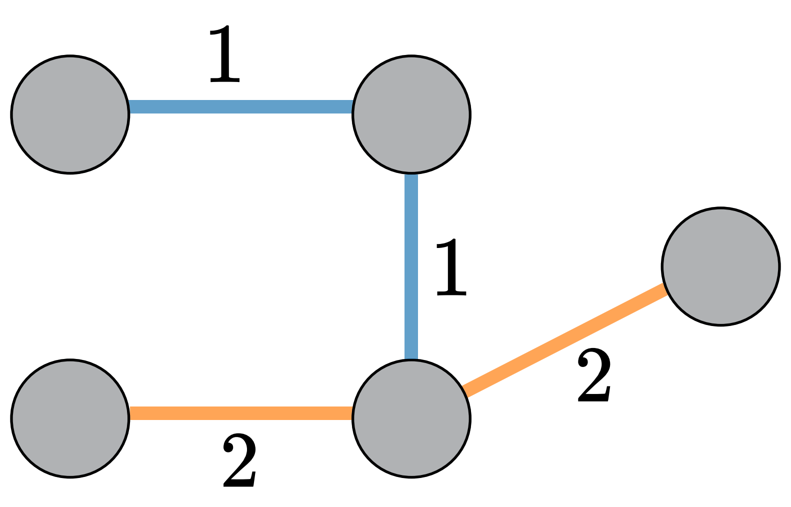
Vertex-color Filtrations: Nested sequence of subgraphs $\varnothing = G^{(0)} \subseteq G^{(1)} \subseteq \ldots \subseteq G$ induced by $f : X \to (0,\infty)$



Persistent homology on graphs (cont.)

Edge-color Filtrations

Edge-colored graph

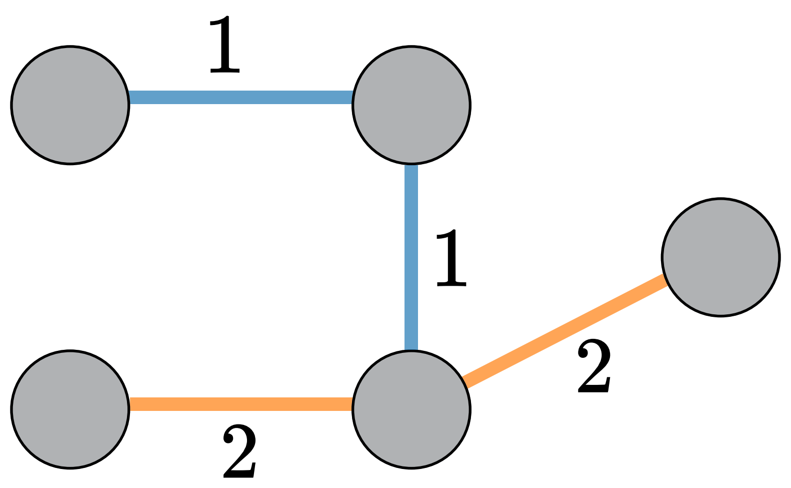


Persistent homology on graphs (cont.)

Edge-color Filtrations

Filtration induced by: $\mapsto 1$ $\mapsto 2$

Edge-colored graph

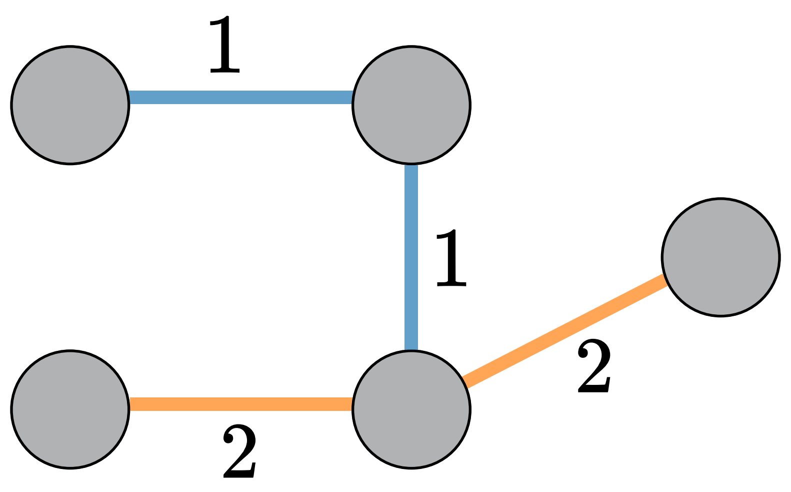


Persistent homology on graphs (cont.)

Edge-color Filtrations

Filtration induced by: $\mapsto 1$ $\mapsto 2$

Edge-colored graph

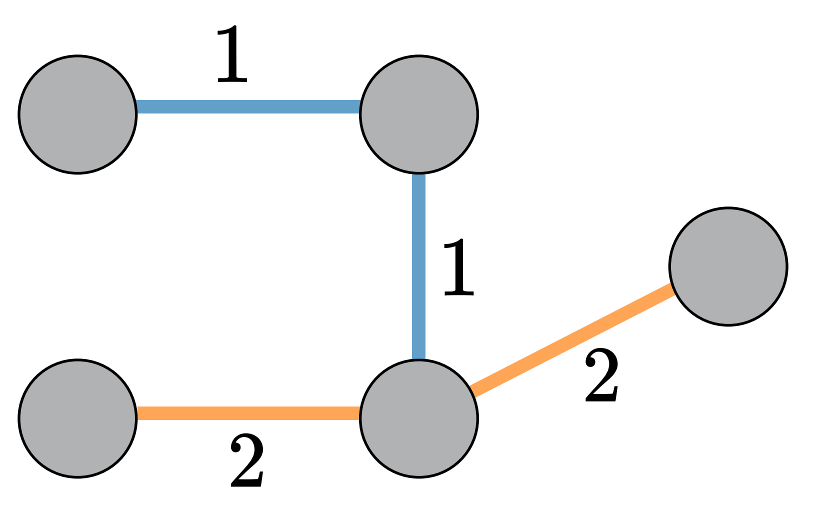


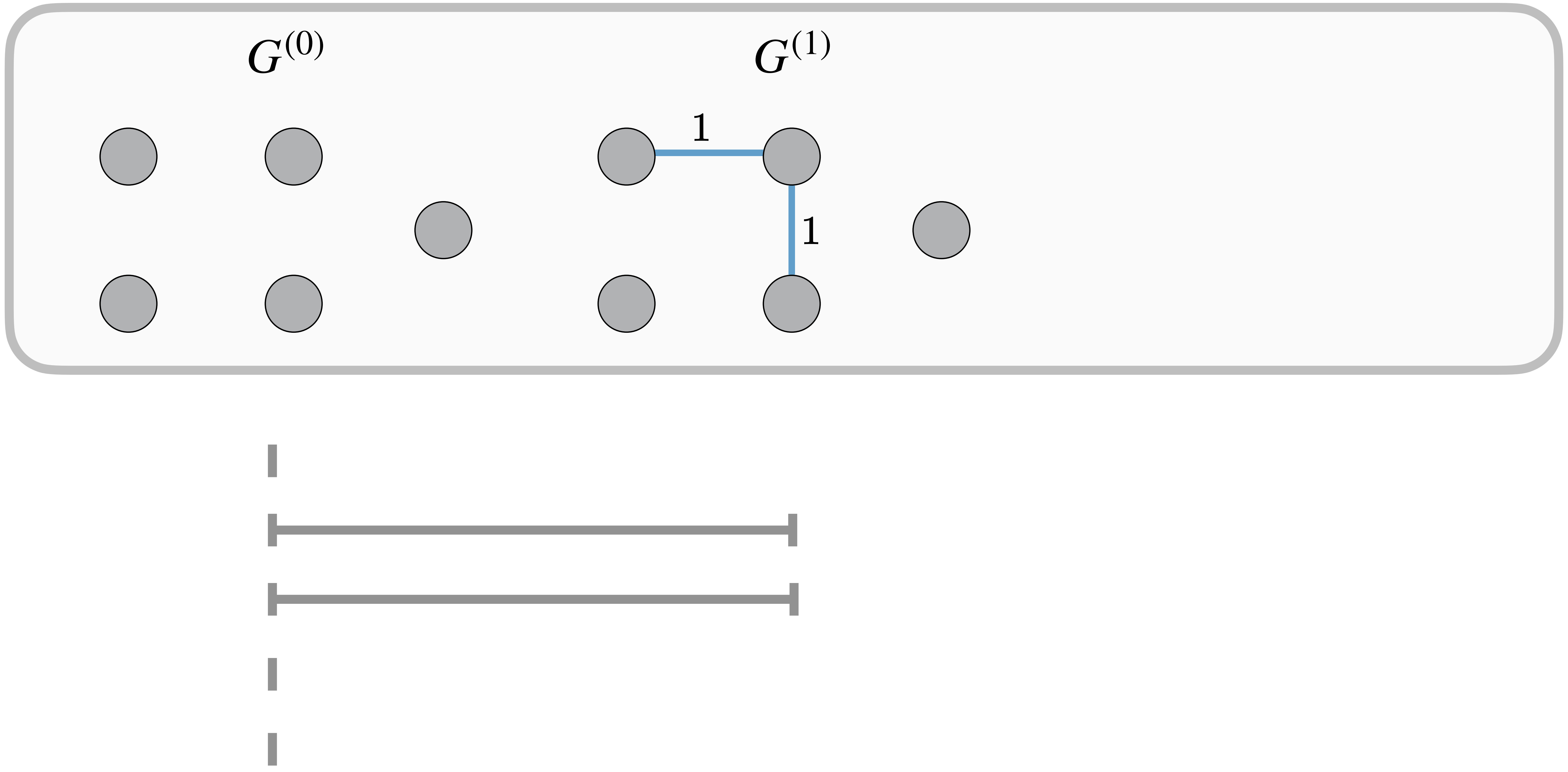
Persistent homology on graphs (cont.)

Edge-color Filtrations

Filtration induced by: ⟋ $\mapsto 1$ ⟋ $\mapsto 2$

Edge-colored graph

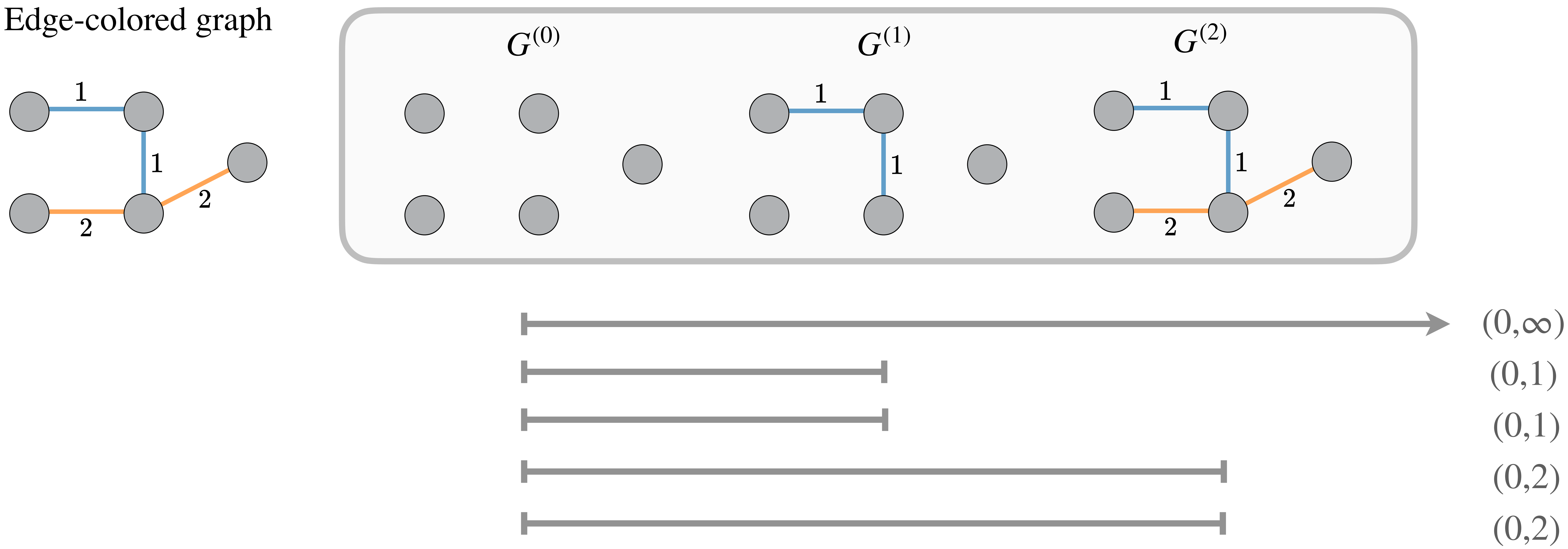




Persistent homology on graphs (cont.)

Edge-color Filtrations

Filtration induced by: 



Motivation

Persistent homology has been used to **boost the predictive capabilities of** graph neural networks (**GNNs**).

However, while the expressivity of GNNs is well-understood (e.g., in terms of the Weisfeiler-Leman test), **the theoretical underpinnings of PH on graphs are less explored**.

In this work, we want to answer two fundamental **open questions**:

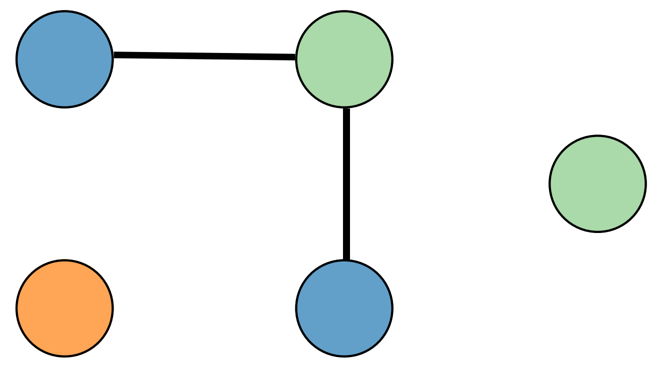
Q1: What is the expressive power of persistent homology (from color-based filtrations) on graphs?

Q2: Can we design more expressive persistence diagrams?

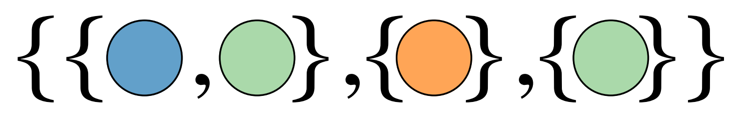
What is the expressive power of persistent homology on graphs?

An important notion: **color-separating sets**

Component-wise colors: The multiset comprising the set of colors of each connected component.

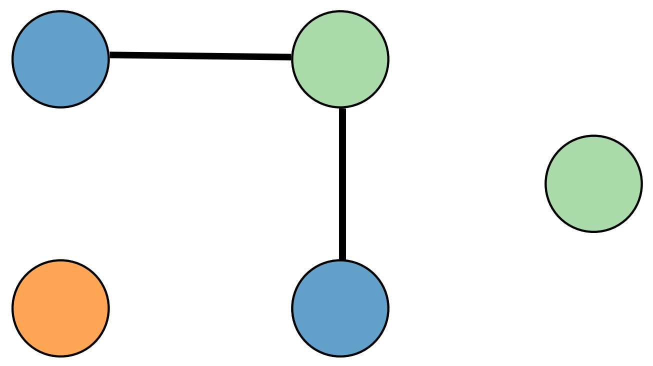


Component-wise colors:

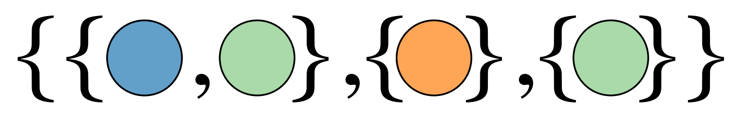


An important notion: **color-separating sets**

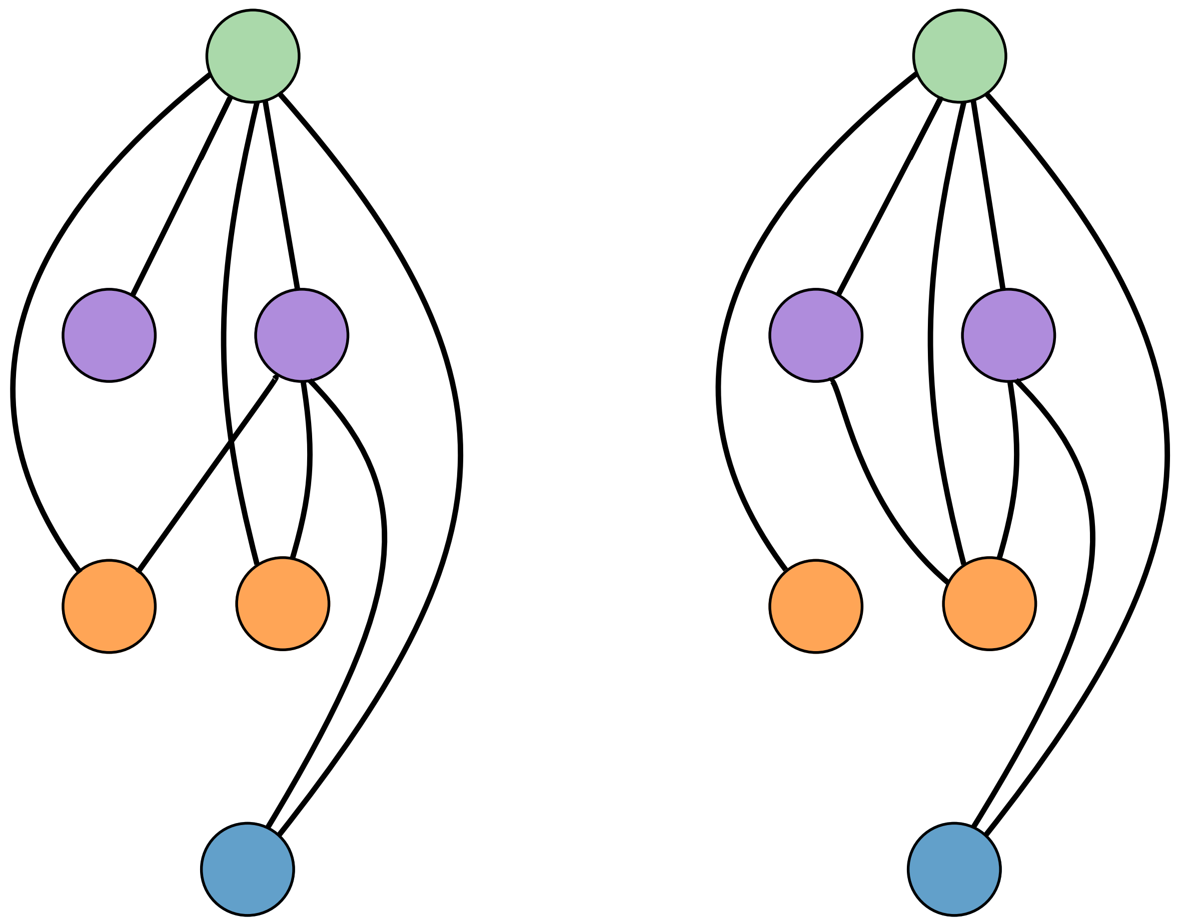
Component-wise colors: The multiset comprising the set of colors of each connected component.



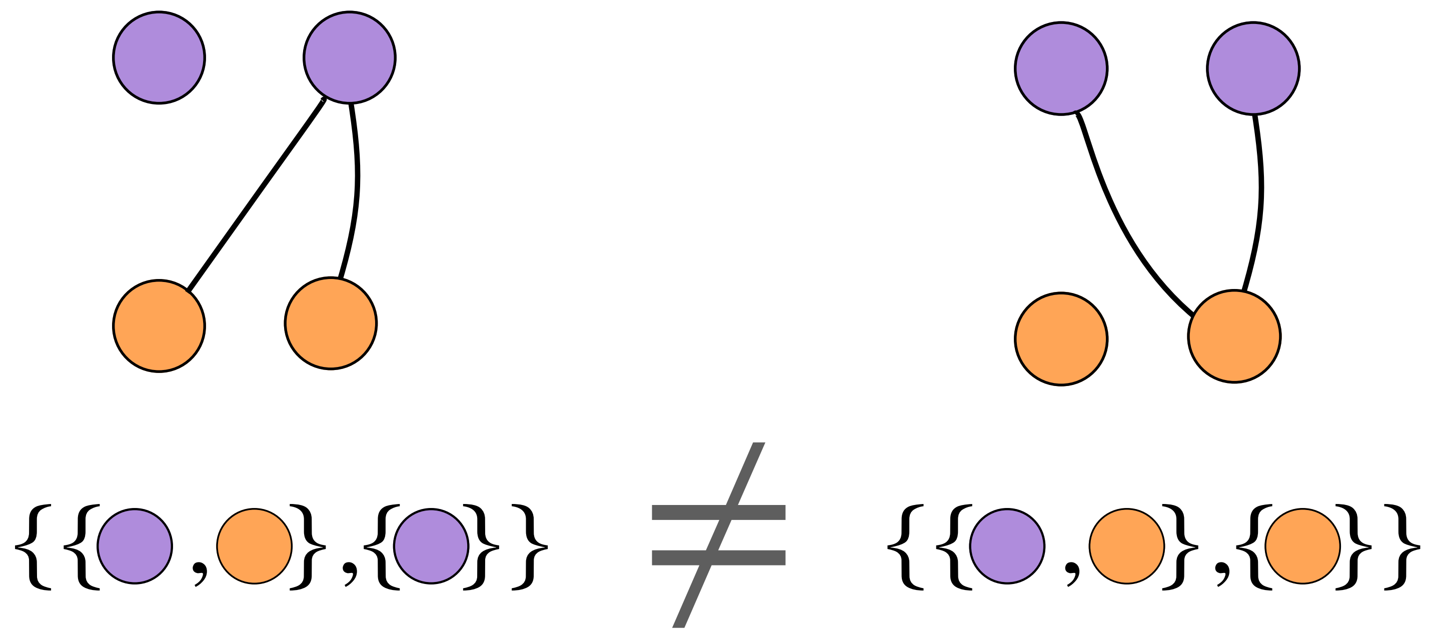
Component-wise colors:

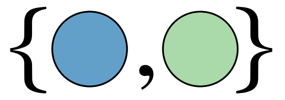


A **color-separating set** for a pair of graphs (G, G') is a set of colors Q such that, if we remove Q from G and G', we obtain subgraphs with **distinct component-wise colors**.



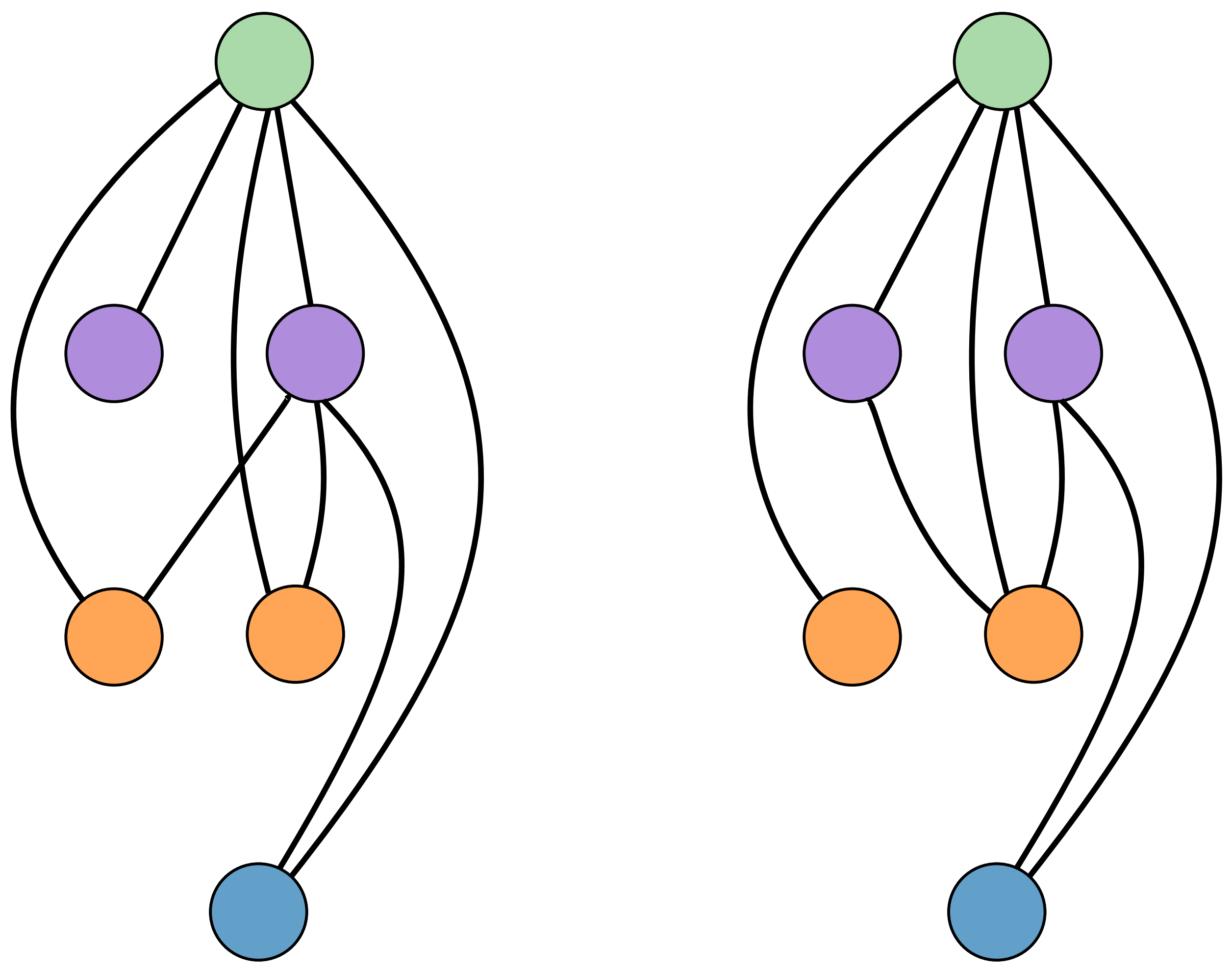
Removing blue and green, we obtain…



Thus,  is a color-separating set!

Theorem 1: On the power of vertex-color filtrations

We can obtain different vertex-color (0-dim) diagrams if and only if there is a color-separating set.

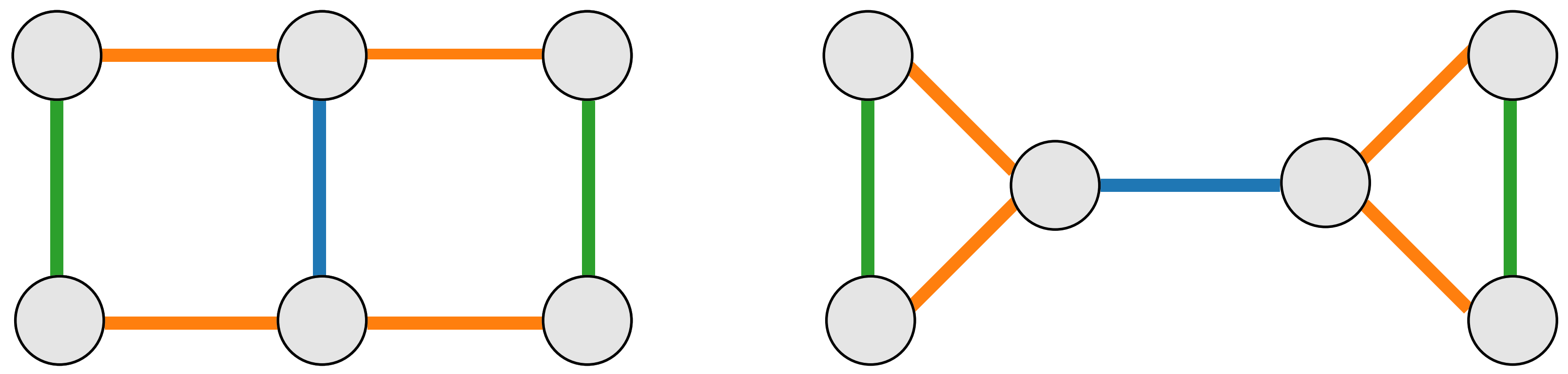


Can PH based on vertex-color filtrations distinguish these graphs?

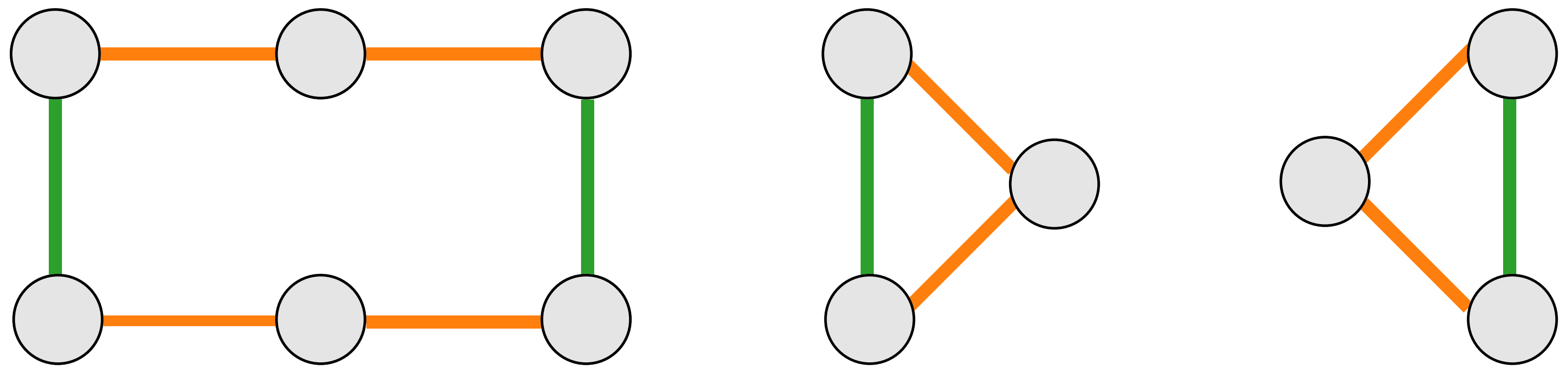
Yes!!{●,●} is a color-separating set!

Another important notion: **color-disconnecting sets**

A **color-disconnecting set** for a pair of graphs (G, G') is a set of colors Q such that, if we remove edges of colors Q from G and G', we obtain subgraphs with **different number of connected components**.



Removing **blue**, we obtain…



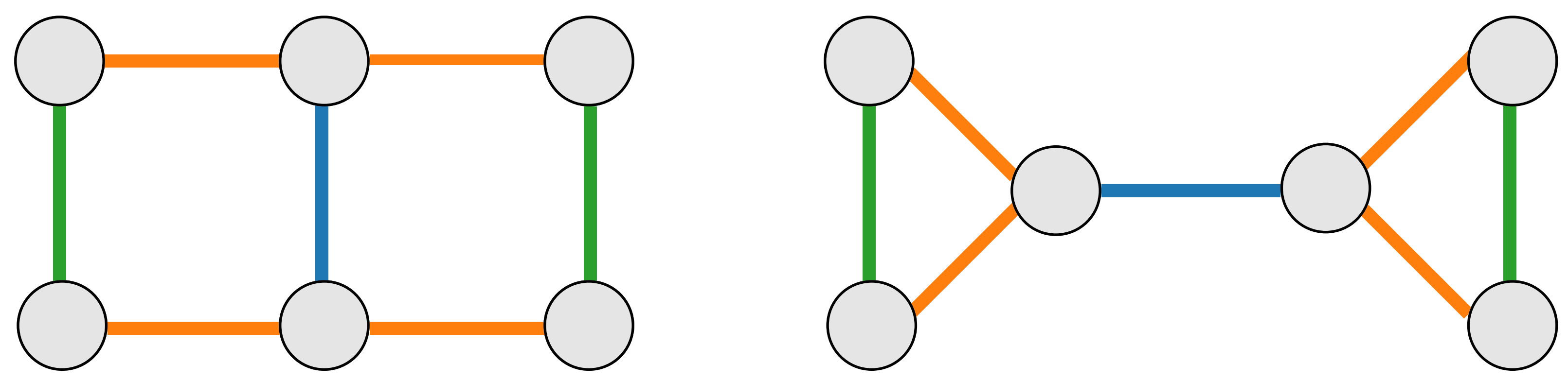
1 component

2 components

Thus, $Q = \{$**blue**$\}$ is a color-disconnecting set!

Theorem 2: On the power of edge-color filtrations

We can obtain different edge-color (0-dim) diagrams if and only if there is a color-disconnecting set.

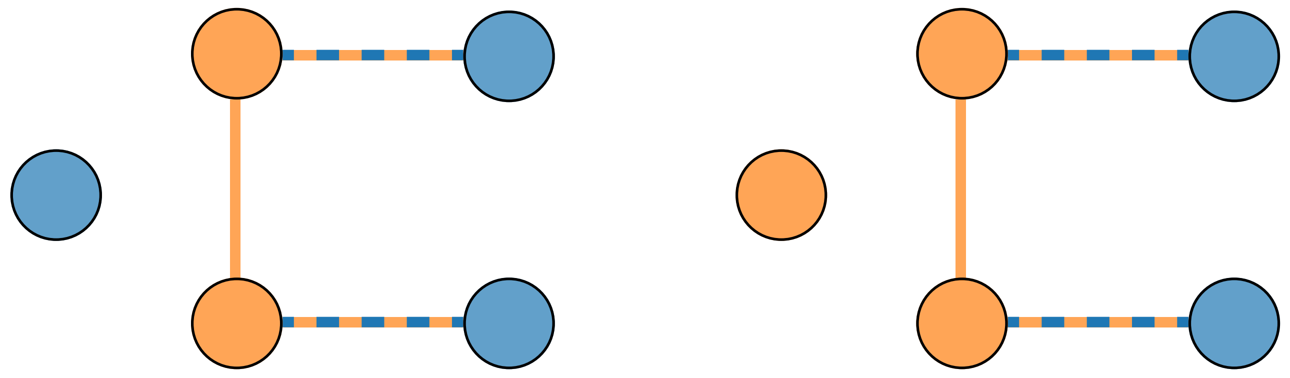


Can PH based on edge-color filtrations distinguish these graphs?

Yes!! $Q = \{\textbf{blue}\}$ is a color-disconnecting set!

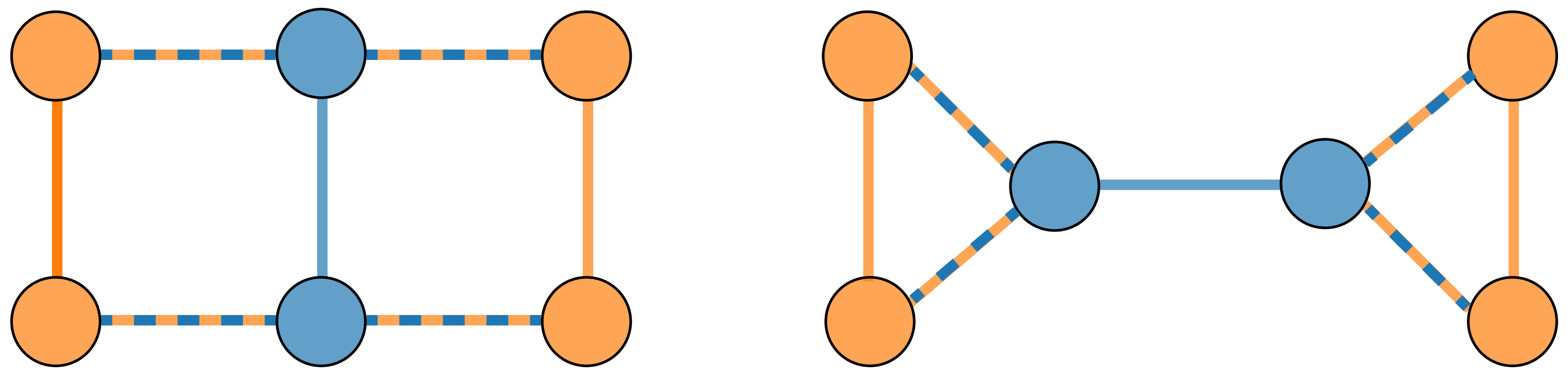
Theorem 3: Vertex-color vs. edge-color filtrations

There exist non-isomorphic graphs that vertex-color filtrations can distinguish but edge-color filtrations cannot, and vice-versa.



Vertex-color **succeeds**

Edge-color **fails**



Vertex-color **fails**

Edge-color **succeeds**

Can we design more expressive persistence diagrams?

RePHINE (**Re**fining **PH** by **I**ncorporating **N**ode-color into **E**dge-based filtration)

Idea: Given independent vertex- and edge-color filtration functions (f_v, f_e), we augment persistence diagrams from edge-color filtrations with vertex-color information.

Original birth and death time (from edge-color filtration)

Color of node u

$$w : \left(b, d, f_v(c(w)), \min_{u \in \mathcal{N}(w)} f_e(\{\{c(u), c(w)\}\}) \right)$$

Independent vertex-color filtration function.

Edge-color filtration function.

Building RePHINE diagrams

Original birth and death time (from edge-color filtration)

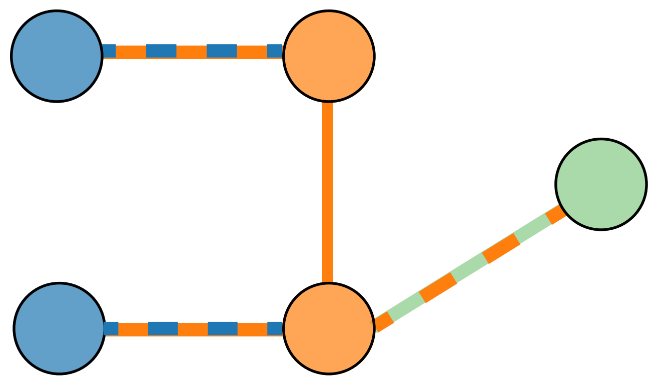
Color of node u

$$w : \left(b, d, f_v(c(w)), \min_{u \in \mathcal{N}(w)} f_e(\{\{c(u), c(w)\}\})\right)$$

Independent vertex-color filtration function.

Edge-color filtration function.

$\mapsto 1$ $\mapsto 2$ $\mapsto 3$ $\mapsto 2$ $\mapsto 3$ $\mapsto 1$



Building RePHINE diagrams

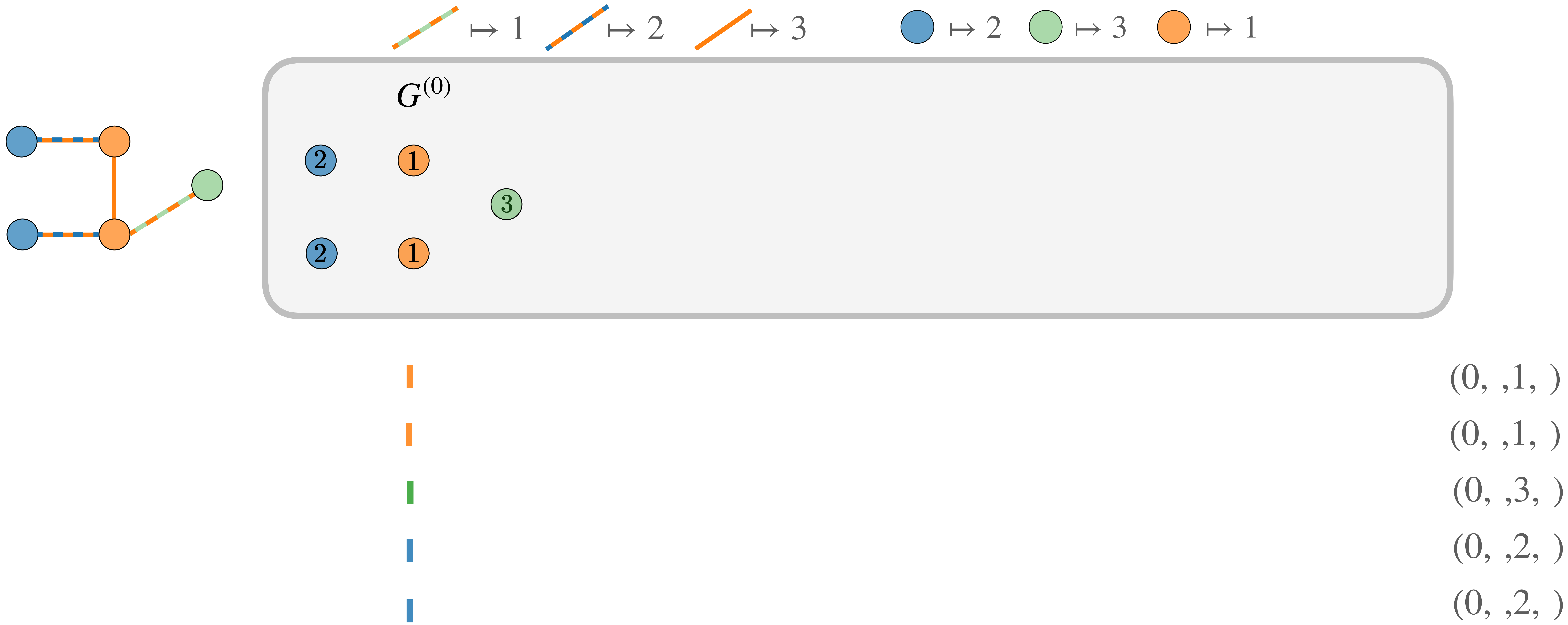
Original birth and death time (from edge-color filtration)

Color of node u

$$w : \left(b, d, f_v(c(w)), \min_{u\in\mathcal{N}(w)} f_e(\{\{c(u), c(w)\}\})\right)$$

Independent vertex-color filtration function.

Edge-color filtration function.



Building RePHINE diagrams

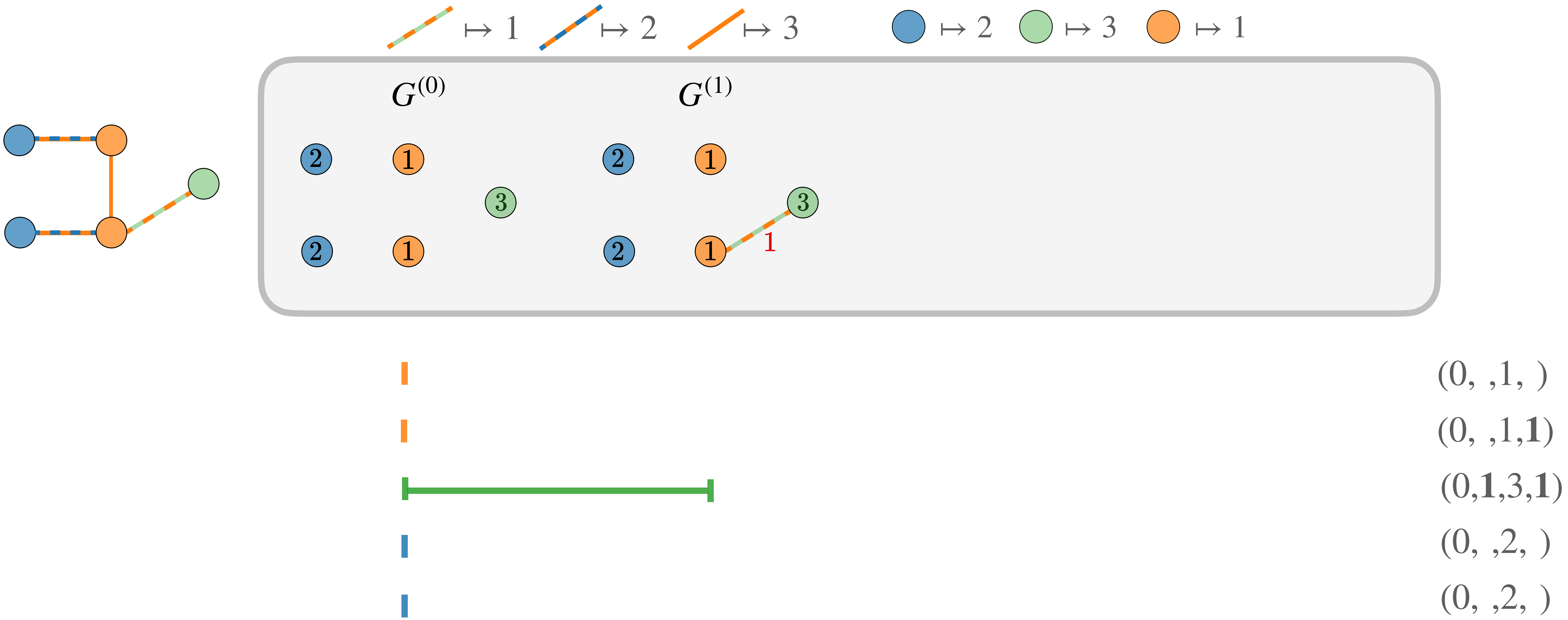
Original birth and death time (from edge-color filtration)

Color of node u

$$w : \left(b, d, f_v(c(w)), \min_{u \in \mathcal{N}(w)} f_e(\{\{c(u), c(w)\}\})\right)$$

Independent vertex-color filtration function.

Edge-color filtration function.



Building RePHINE diagrams

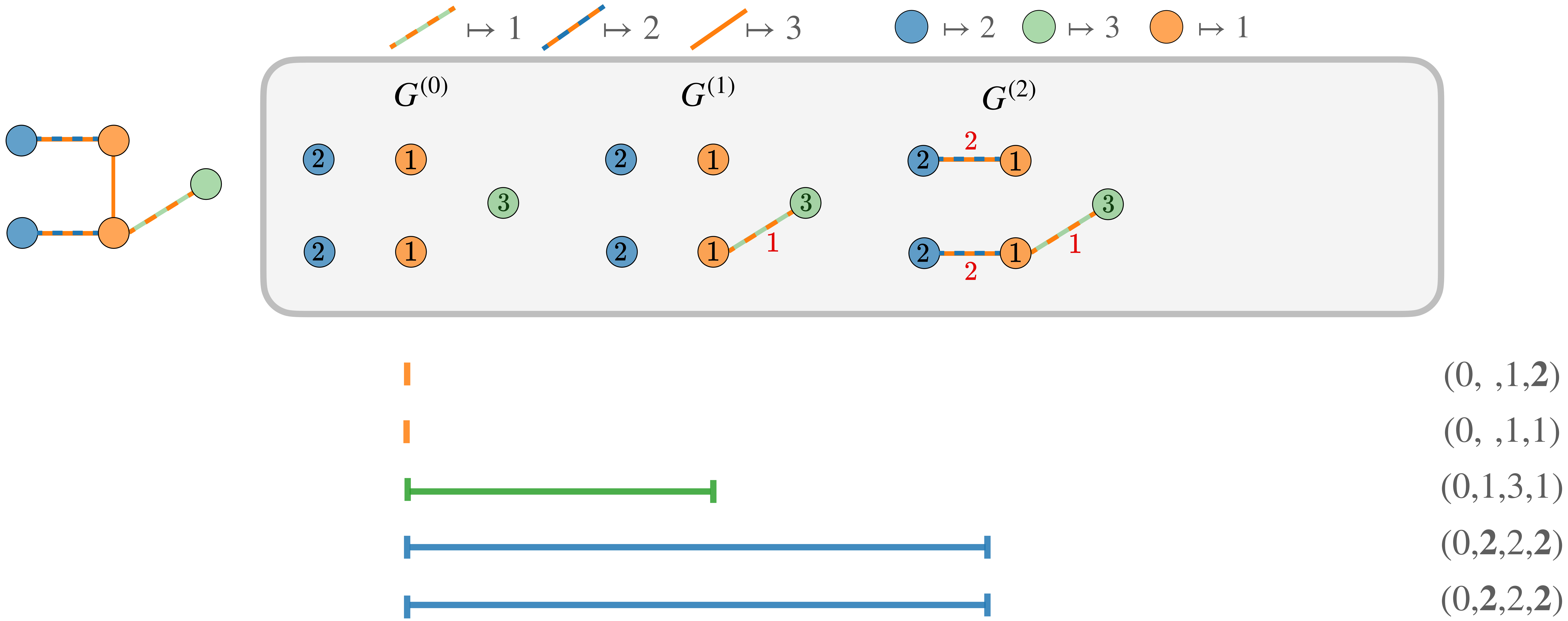
Original birth and death time (from edge-color filtration)

Color of node u

$$w : \left(b, d, f_v(c(w)), \min_{u \in \mathcal{N}(w)} f_e(\{\{c(u), c(w)\}\})\right)$$

Independent vertex-color filtration function.

Edge-color filtration function.



Building RePHINE diagrams

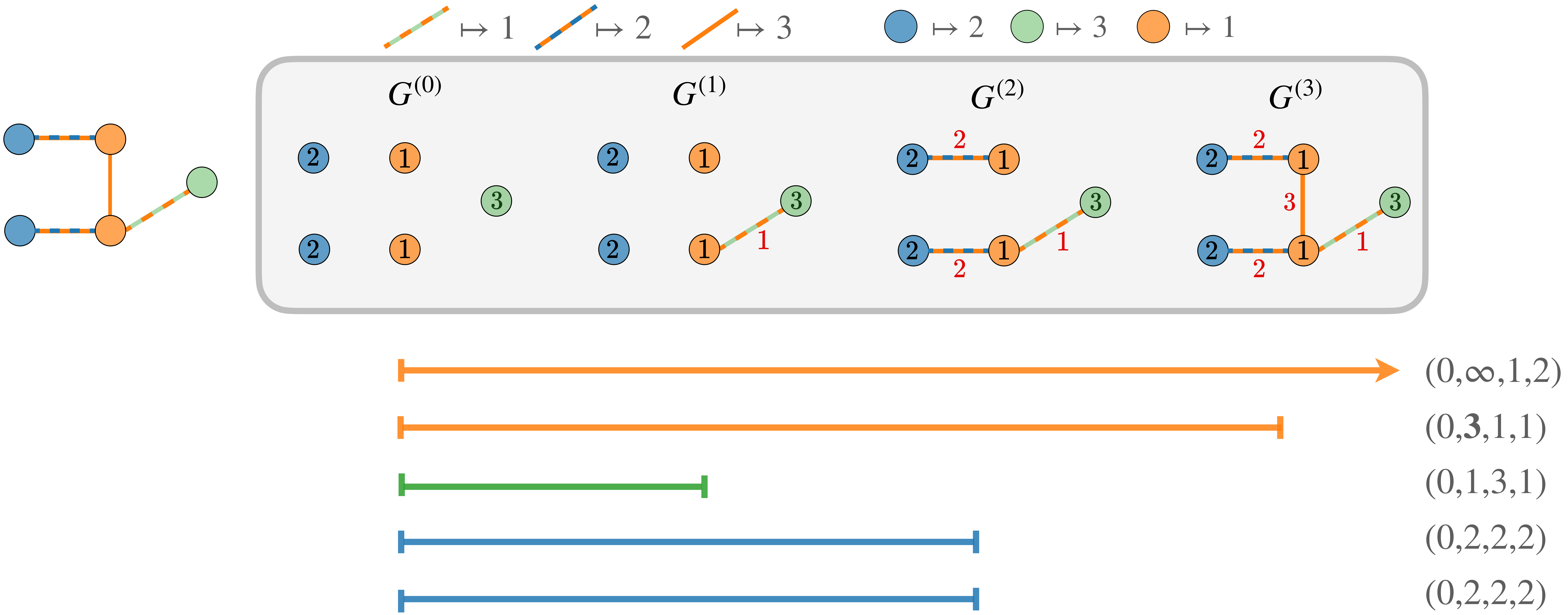
Original birth and death time (from edge-color filtration)

Color of node u

$$w : \left(b, d, f_v(c(w)), \min_{u \in \mathcal{N}(w)} f_e(\{\{c(u), c(w)\}\})\right)$$

Independent vertex-color filtration function.

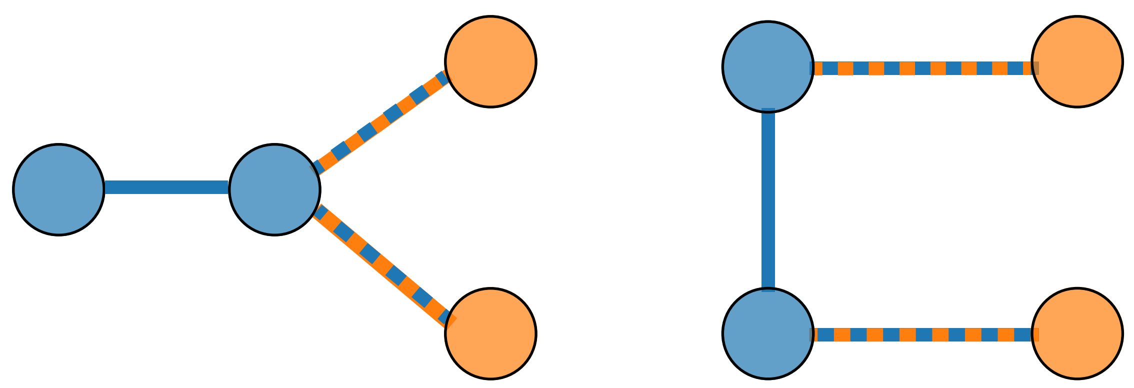
Edge-color filtration function.



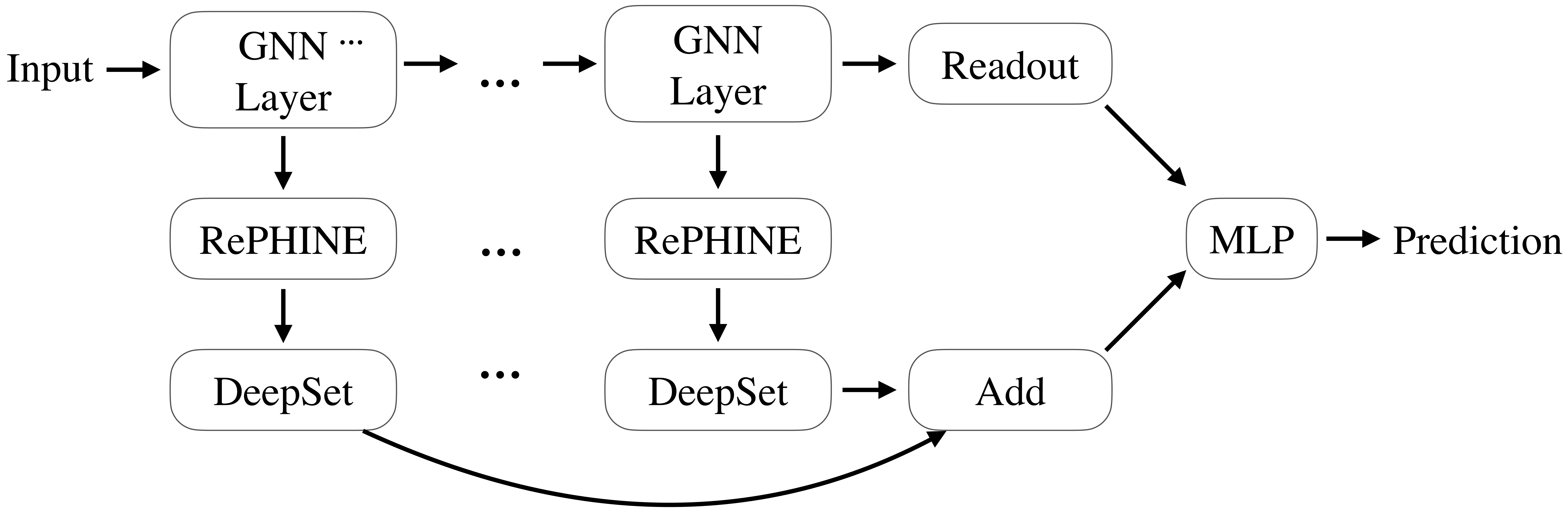
Theorem 4: RePHINE vs color-based diagrams

RePHINE is isomorphism invariant and is strictly more expressive than color-based PH.

Two graphs that color-based **PH cannot distinguish**, but **RePHINE can**.



Combining RePHINE and GNNs



Results on real-world data

We process the persistence diagrams using DeepSets and combine the resulting vectors with GNN embeddings.

Table 1: Predictive performance on graph classification. We denote in bold the best results. For ZINC, lower is better. For most datasets, RePHINE is the best-performing method.

GNN	Diagram	NCI109 ↑	PROTEINS ↑	IMDB-B ↑	NCI1 ↑	MOLHIV ↑	ZINC ↓
GCN	-	76.46 ± 1.03	70.18 ± 1.35	64.20 ± 1.30	74.45 ± 1.05	74.99 ± 1.09	0.875 ± 0.009
	PH	77.92 ± 1.89	69.46 ± 1.83	64.80 ± 1.30	79.08 ± 1.06	73.64 ± 1.29	0.513 ± 0.014
	RePHINE	**79.18** ± 1.97	**71.25** ± 1.60	**69.40** ± 3.78	**80.44** ± 0.94	**75.98** ± 1.80	**0.468** ± 0.011
GIN	-	76.90 ± 0.80	**72.50** ± 2.31	**74.20** ± 1.30	76.89 ± 1.75	70.76 ± 2.46	0.621 ± 0.015
	PH	78.35 ± 0.68	69.46 ± 2.48	69.80 ± 0.84	79.12 ± 1.23	73.37 ± 4.36	0.440 ± 0.019
	RePHINE	**79.23** ± 1.67	72.32 ± 1.89	72.80 ± 2.95	**80.92** ± 1.92	**73.71** ± 0.91	**0.411** ± 0.015

Wanna know more?

Visit our poster: **#629**

Thu 14 Dec 10:45 a.m. CST

Code: www.github.com/Aalto-QuML/rephine

Theoretical contributions of this work	
On vertex-level filtrations (Section 2 and Section 3.1):	
Inconsistency issues due to injective vertex filtrations	Lemma 1
Real holes ($d = \infty$) $\cong$ Component-wise colors	Lemma 2
Almost holes ($b \neq d, d \neq \infty$) $\cong$ Separating sets	Lemma 3
Distinct almost holes $\Rightarrow$ Color-separating set	Lemma 4
Birth time of persistence tuples $\cong$ Vertex color	Lemma 5
The expressive power of vertex-color filtrations	Theorem 1
On edge-level filtrations (Section 3.2):	
Almost holes $\cong$ Disconnecting sets	Lemma 6
Reconstruction of disconnecting sets	Lemma 7
The expressive power of edge-color filtrations	Theorem 2
Vertex-level vs. edge-level filtrations (Section 3.3):	
Vertex-level persistence $\not\succ$ edge-level persistence, and vice-versa	Theorem 3
New method (RePHINE) (Section 4):	
RePHINE is isomorphism invariant	Theorem 4
RePHINE $\succ$ vertex-, edge-, and vertex- $\cup$ edge-level diagrams	Theorem 5

Johanna Immonen
johanna.x.immonen@helsinki.fi

Amauri H. Souza
amauri.souza@aalto.fi
@amaurihsouza

Vikas Garg
vgarg@csail.mit.edu
@montsgarg