mathematics

MDPI

Article

Extending Undirected Graph Techniques to Directed Graphs via Category Theory

Sebastian Pardo-Guerra [1,2,†], Vivek Kurien George [1,2,†], Vikash Morar [1,2,†], Joshua Roldan [1,2,†] and Gabriel Alex Silva [1,2,3,*]

[1] Department of Bioengineering, University of California San Diego, La Jolla, CA 92037, USA; spardoguerra@ucsd.edu (S.P.-G.); vgeorge@ucsd.edu (V.K.G.); vmorar@ucsd.edu (V.M.); jmroldan@ucsd.edu (J.R.)
[2] Center for Engineered Natural Intelligence, University of California San Diego, La Jolla, CA 92037, USA
[3] Department of Neurosciences, University of California San Diego, La Jolla, CA 92037, USA
* Correspondence: gsilva@ucsd.edu
† These authors contributed equally to this work.

Abstract: We use Category Theory to construct a 'bridge' relating directed graphs with undirected graphs, such that the notion of direction is preserved. Specifically, we provide an isomorphism between the category of simple directed graphs and a category we call 'prime graphs category'; this has as objects labeled undirected bipartite graphs (which we call prime graphs), and as morphisms undirected graph morphisms that preserve the labeling (which we call prime graph morphisms). This theoretical bridge allows us to extend undirected graph techniques to directed graphs by converting the directed graphs into prime graphs. To give a proof of concept, we show that our construction preserves topological features when applied to the problems of network alignment and spectral graph clustering.

Keywords: undirected graphs; directed graphs; spectral clustering; network alignment; category theory

MSC: 00A69; 05C50; 18A99; 68R10

Citation: Pardo-Guerra, S.; Kurien George, V.; Morar, V.; Roldan, J.; Silva, G.A. Extending Undirected Graph Techniques to Directed Graphs via Category Theory. *Mathematics* **2024**, *12*, 1357. https://doi.org/10.3390/math12091357

Academic Editor: Andrea Scozzari

Received: 9 April 2024
Revised: 28 April 2024
Accepted: 28 April 2024
Published: 29 April 2024

1. Introduction

Networks naturally arise in many real-world situations. Examples vary from macro-structures, such as social networks [1] and economic trade networks [2], to microscopic structures involving protein–protein interactions [3], transcriptional regulation networks [4], gene regulatory networks [5], and both biological and artificial neural networks [6,7]. While real-world networks are often modeled as directed graphs, their computational analysis is not only challenging but far more cumbersome and restricted. Thus, it would be helpful to analyze and solve certain classes of problems for directed graphs from an undirected graph framework. Here, we tackle this problem by using notions and principles of Category Theory (CT) within a graph context. Put simply, CT studies abstract structures and their relations. These structures, or *categories*, are composed of a collection of things we called "objects", and a collection of relations between two objects that we call "morphisms". Originally, CT developed within pure mathematics; much more recently, it started to be used broadly across the natural sciences and engineering, including applications in machine learning and artificial neural networks [8], biological networks [9,10], and social networks [11].

The way we bridge a directed graph framework with an undirected one is by first considering a category of undirected graphs that encode the notion of direction. This category, which we call the *prime graphs category*, has as objects undirected graphs equipped by a 'prime labeling', and as morphisms undirected graphs morphisms that preserve

the prime labeling. Indeed, it is the prime labeling that provides a notion of direction over the structure of the undirected graph, allowing us to define a unique direction when transforming to a directed graph, and vice versa. With this in mind, we construct a bijective functor that relates the category of simple directed graphs with the category of prime graphs, such that the notion of direction is preserved. It is worth mentioning that one can always relate a directed graph to an undirected graph in a trivial way: by simply considering the underlying structure and 'forgetting' the direction. This correspondence gives rise to a 'forgetful' functor which is not invertible. In [12], Miller provides one of the first nontrivial transformations between simple directed graphs and undirected graphs by the construction of "gadgets". These gadgets are used to encode the notion of direction, just as our prime label does. However, each gadget adds seven nodes to the corresponding undirected graphs, while in our framework, we are just adding a prime label node. Additionally, we are the first to address the problem of converting directed graph morphisms into undirected graphs, while preserving the notion of direction. This latter aspect is crucial for both of our applications; in network alignment, the labeling and its preservation through prime graph morphisms ensures the mapping between appropriate nodes through the node similarity metric, while in spectral graph clustering, the labeling plays a role in determining edge weights.

Network alignment is a technique that allows us to compare two networks. This is performed by "putting one on top of the other" in such a way that the structure—or topology—between the networks being compared is preserved as much as possible, and the similarity between the networks is quantified. To date, several network alignment tools exist for undirected networks (see [13–16]), but to the best of our knowledge, none exist for directed graphs. Our framework, hence, proposes to perform network alignment on directed graphs via their corresponding prime graphs (which are undirected). Within this line of applications, we show the efficacy of our approach empirically by using synthetically generated pairs of networks whose pairwise similarity is known and controlled by the graph generator's pairwise correlation coefficient. Our results in Section 3.1.1 show that there is a strong statistical correspondence between the generated networks and their resulting pairwise network similarity scores.

Spectral clustering is a widely used and robust technique that considers the spectrum—or eigenvalues and eigenvectors—of the *graph Laplacian* matrix to partition the nodes of a graph into clusters. More precisely, one can cluster the nodes of a graph by sorting the components of the eigenvector corresponding to the second-smallest eigenvalue of the characteristic polynomial of the graph Laplacian matrix. Initially, spectral clustering was developed for undirected graphs using their adjacency and Laplacian matrices [17]. Later, the technique was extended to directed graphs [18], where a transition probability matrix—or random walk—is used to overcome the asymmetry found in their adjacency and Laplacian matrices. There are heuristic techniques that circumvent the latter construction by making the adjacency matrix of a directed graph symmetric; therefore, they can define a symmetric graph Laplacian matrix [19], which they then use to find the cuts of the directed graph. However, while they are able to show empirically that the resulting cuts are the same, they do not prove equivalence, as we do. Precisely, we prove that our framework preserves minimum cuts, and consequently, it preserves clusters in directed graphs and their respective prime graph counterparts.

The paper is organized as follows. Section 2 gives preliminary notions about directed and undirected graphs from a category theory perspective. We describe the category of prime graphs and prime graph morphisms. We also present the concepts and theoretical results used in the application of our framework. In Section 3, we discuss the applications of network alignment and spectral clustering for directed graphs via functoriality. Finally, Section 4 gives some concluding arguments.

2. Methods

Mathematically, a *graph* $G = (V, E)$ is a structure that consist of a set of vertices V (nodes) and a set $E \subseteq V \times V$, which we call edges (connections). Within this context, we say that a graph is undirected if its set of edges are connections without directionality. A graph is directed if all its edges are connections with a direction. Now, given an undirected graph, we represent an edge between the nodes u and v by the unordered pair (u, v), equivalently (v, u). Instead, for a directed graph, an edge from node u to node v is represented by the ordered pair (u, v); conceptually, we can think of this edge as an arrow with initial node u and terminal node v. Also, we say that the nodes u and v are neighbors if there is at least one edge connecting nodes u and v. Throughout this work, we will consider simple directed graphs, this is, directed graphs with no multiple edges and no self loops.

Intuitively, a graph morphism is a function between the vertex sets that preserves the structure or topology of the graphs; this is, it preserves the edges under transformation:

Definition 1. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two undirected graphs. An undirected graph morphism is a function $f : V_1 \longrightarrow V_2$ that maps adjacent vertices in G_1 into adjacent vertices in G_2. Algebraically, this means that for any pair of vertices $u, v \in V_1$ with $(u, v) \in E_1$, we have $(f(u), f(v)) \in E_2$.*

For two directed graphs G_1 and G_2, we say that a function $V_1 \xrightarrow{f} V_2$ is a *directed graph morphism* if f maps initial nodes into initial nodes, and terminal nodes into terminal nodes. One can verify that the composition of two graph morphism always yields another graph morphism, in both undirected and directed cases. Further, the composition is an associative operation that has a neutral element, called the identity morphism (which coincides with the identity map). With this in mind, one can show that the collection of all undirected graphs and all undirected graph morphisms (**UndGraph**) forms a category. Similarly, the collection of all simple directed graphs and all directed graph morphisms (**DGraph**) forms a category. As we see next, the category **DGraph** is isomorphic to a subcategory of **UndGraph**; this subcategory, in fact, is the category of prime graphs (**PGraph**) that we define.

Before describing the isomorphism between **DGraph** and **PGraph**, we will detail the category of prime graphs and its connection to simple directed graphs. Conceptually, a prime graph is an undirected graph which admits a 'prime labeling' on its set of nodes (Figure 1). By allowing a "prime labeling", we mean that there exists a labeling function on the vertex set with the following two properties; any prime labeled node has only non-prime labeled nodes as neighbors (and vice versa), and any prime labeled node is always adjacent to its non-prime labeled counterpart node.

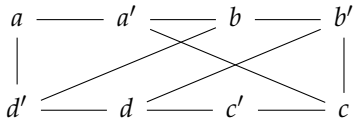
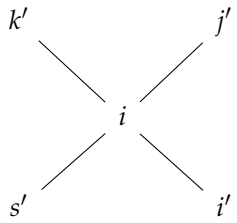


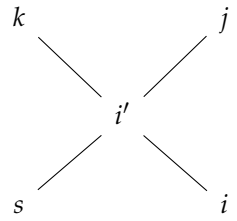
Figure 1. A prime graph with eight nodes; four nodes have a prime labeling, while the other four nodes have a non-prime labeling.

Definition 2. *Let $G_u = (V_u, E_u)$ be an undirected graph with $2n$ vertices, and let $I = \{a_1, a_2, \cdots, a_n\}$. We say that G_u admits a **prime labeling** if there exists a bijective function $\varphi : V_u \longrightarrow I \cup I'$ such that for $v \in V_u$, one has the following three cases:*

(i) If $\varphi(v) = i$ and $i \in I$, then for each neighbor u of v, $\varphi(u) = k'$ for some $k' \in I'$. We visualize this as



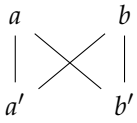
(ii) If $\varphi(v) = i'$ and $i' \in I'$, then for each neighbor u of v, $\varphi(u) = k$ for some $k \in I$. We visualize this as



(iii) For each $v \in V_u$, if $\varphi(v) = i$ and $i \in I$, then there exists $u \in V_u$ a neighbor of v such that $\varphi(u) = i'$ and $i' \in I'$.

For illustrative purposes, consider the following graphs:

The graph on the left is not a prime graph, as the prime labeled nodes are connected to each other. For the same reasons, the graph on the right is not a prime graph; however, this last graph becomes a prime graph if we endow it with the following prime labeling:



We also observe that one can visualize a prime graph as an undirected bipartite graph. This is a consequence of the definition of a prime labeling, as prime labeled node only have non-prime labeled nodes as neighbors, and vice versa. Please see in Figure 2.

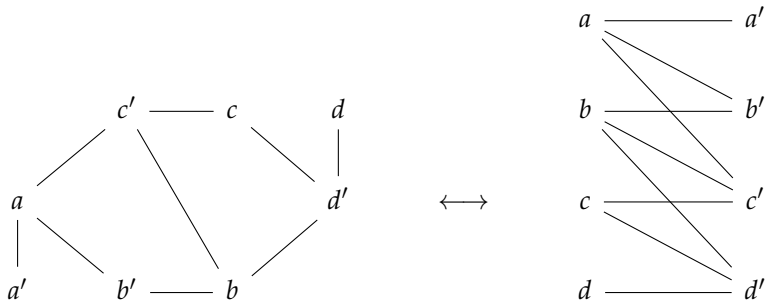


Figure 2. A prime graph corresponds to a bipartite graph. The set of nodes of a prime graph can be written as the disjoint union of prime labeled nodes and the non-prime labeled nodes.

Now, due to condition (iii), we can naturally induce a directed graph from a prime graph. In this case, the direction of an arrow will be given by the 'prime' label vertex. For instance, if we consider the linear undirected graph G

$$\cdot \text{——} \cdot \text{——} \cdot \text{——} \cdot$$

we can give the prime labeling

$$i' \text{——} i \text{——} j' \text{——} j$$

which induces the directed graph

$$w_i \longrightarrow w_j,$$

having a directed edge from an initial vertex w_i to a terminal vertex w_j.

In terms of morphisms, we can think of a prime graph morphism as an undirected graph morphism that preserves the prime and non-prime labelings. Formally:

Definition 3. *Let G_{u_1} and G_{u_2} be two prime graphs, with labeling functions φ_{I_1} and φ_{I_2}, respectively. A* prime graph morphism *is an undirected graph morphism $f\colon V_{u_1} \longrightarrow V_{u_2}$ that satisfies the following conditions:*

(i) (Non-prime label preservation) If $v \in V_{u_1}$ with $\varphi_{I_1}(v) = i$ for some $i \in I_1$, then $f(v) \in V_{u_2}$ is such that

$$\varphi_{I_2}(f(v)) = j$$

for $j \in I_2 \subseteq I_2 \cup I_2'$.

(ii) (Prime label preservation) If $w \in V_{u_1}$ with $\varphi_{I_1}(w) = i'$ for some $i' \in I_1'$, then $f(w) \in V_{u_2}$ is such that

$$\varphi_{I_2}(f(w)) = j'$$

for $j' \in I_2' \subseteq I_2 \cup I_2'$.

(iii) If $v, w \in V_{u_1}$ are adjacent vertices with $\varphi_{I_1}(v) = k$ and $\varphi_{I_1}(w) = k'$, then one always has that

$$\varphi_{I_2}(f(w)) = \big(\varphi_{I_2}(f(v))\big)'.$$

The above can be rephrased by saying that a prime graph morphism is an undirected graph morphism compatible with the non-prime and prime labelings. From an algebraic perspective, this compatibility condition means that, for each prime graph morphism $f : V_{u_1} \longrightarrow V_{u_2}$, there exists a function $f_I : I_1 \cup I_1' \longrightarrow I_2 \cup I_2'$ making

$$\begin{array}{ccc} V_{u_1} & \xrightarrow{\varphi_{I_1}} & I_1 \cup I_1' \\ {\scriptstyle f}\big\downarrow & & \big\downarrow{\scriptstyle f_I} \\ V_{u_2} & \xrightarrow[\varphi_{I_2}]{} & I_2 \cup I_2' \end{array}$$

a commutative diagram; that is, $f_I \circ \varphi_{I_1} = \varphi_{I_2} \circ f$.

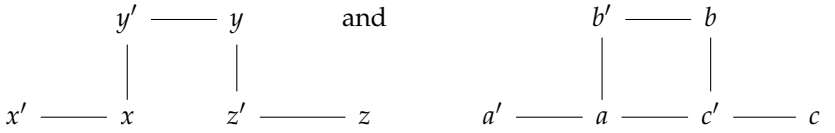
Furthermore, the composition of two prime graph morphisms results in a prime graph morphism itself. This follows from the fact that the composition of undirected graph morphisms is a closed operation, and also from the fact that the composition of prime graph morphisms preserves the prime and non-prime labeling conditions. Expressed in diagrams, this latter aspect is equivalent to saying that the commutativity of the large diagram is a consequence of the commutativity of the smaller diagrams:

$$\begin{array}{ccc} V_{u_1} & \xrightarrow{\varphi_I} & I_1 \cup I_1' \\ f\downarrow & & \downarrow f_{I_1} \\ V_{u_2} & \xrightarrow{\varphi_{I_2}} & I_2 \cup I_2' \\ g\downarrow & & \downarrow g_I \\ V_{u_3} & \xrightarrow{\varphi_I} & I_3 \cup I_3'. \end{array}$$

Additionally, as the composition operation of undirected graph morphisms is an associative operation, it follows that the composition of prime graph morphisms is also an associative operation. Further, for any prime graph G_u, its identity prime graph morphism coincides with the identity map defined on the undirected graph G_u. Considering the above, one has that the collection of prime graphs, along with the collection of prime graph morphisms, form a category.

Theorem 1. *The collection of all prime graphs, and all prime graph morphisms, defines a category denoted by* ***PGraph****.*

It is worth noting that not every undirected graph morphism induces a prime graph morphism. To see this, let us consider the following prime graphs:



Then, the function with correspondence rule $x' \longmapsto a'$, $x \longmapsto a$, $y' \longmapsto c'$, $y \longmapsto b$, $z' \longmapsto b'$, and $z \longmapsto a$ is an undirected graph morphism which is not a prime graph morphism. In fact, this illustrates that the category **PGraph** is not a full subcategory of **UndGraph**. Consequently, both the graph isomorphism problem and the subgraph isomorphism problem for prime graphs differentiates that for undirected graphs, which are known to be open and NP-complete, respectively.

2.1. **DGraph** *and* **PGraph** *Are Isomorphic Categories*

The first part of this subsection describes the functors $\mathcal{L}$: **DGraph** $\longrightarrow$ **PGraph** and $\mathcal{M}$: **PGraph** $\longrightarrow$ **DGraph**. The second part of this subsection shows that $\mathcal{L}$ and $\mathcal{M}$ are inverse of each other.

2.1.1. The Functors $\mathcal{L}$ and $\mathcal{M}$

The following two propositions describe the assignment on objects and morphisms of functor $\mathcal{L}$: **DGraph** $\longrightarrow$ **PGraph**:

Proposition 1. *Let G_d be a directed graph in* ***DGraph****. Then, G_d induces a prime graph G_u in* ***PGraph****.*

Proof of Proposition 1. Let $G_d = (V_d, E_d)$ be a finite directed graph in **DGraph**. Without loss of generality, let us suppose that V_d is represented by the set $\{v_1, v_2, \cdots, v_n\}$. Thus, by denoting $v_{i'} = v_i'$ for each $i' \in \{1', \cdots, n'\}$, the prime graph G_u has vertex set

$$V_u := \{v_1, v_2, \cdots, v_n, v_1', v_2', \cdots, v_n'\}.$$

Now, to define the edge set E_u, we will consider the following two cases:

(i) for each $i \in \{1, \ldots, n\}$, the tuple (v_i, v_i') defines an edge in G_u;
(ii) for $i \neq j$ in $\{1, \cdots, n\}$, we have that (v_i, v_j') defines an edge in G_u if, and only if, there exists a directed edge $(v_i, v_j) \in E_d$.

In other words,

$$E_u = \begin{cases} (v_i, v'_j) & \text{if } i \neq j \text{ and } (v_i, v_j) \in E_d \\ (v_i, v'_i) & \text{for each } i \in \{1, \cdots, n\}. \end{cases}$$

We claim that G_u admits an I-labeling. Let $I = \{1, 2, \cdots, n\}$ and $I' = \{1', 2', \cdots, n'\}$ be the non-prime and prime sets, respectively. We define the labeling function $V_u \xrightarrow{\varphi} I \cup I'$ as follows. For each $v \in V_u$,

$$\varphi(v) = \begin{cases} i & \text{if } v = v_i \text{ with } v_i \in \{v_1, v_2, \cdots, v_n\} \\ i' & \text{if } v = v'_i \text{ with } v'_i \in \{v'_1, v'_2, \cdots, v'_n\} \end{cases}$$

Clearly, φ is a bijective function. Moreover, based on how G_u is defined, condition (iii) of Definition 2 is satisfied. Thus, it suffices to show that G_u equipped by φ satisfies conditions (i) and (ii). Now, if $v \in V_u$ is such that $\varphi(v) = i$, then $v = v_i$ for $i \in I$. Hence, as the incident edges to vertex $v_i \in V_u$ have the form (v_i, v'_j)—that is, when $(v_i, v_j) \in E_d$—or (v_i, v'_i), it follows that the neighbors of v_i also have the form v'_j, with $j \neq i$, or v'_i. In any case, one has that $\varphi(v'_j) = j'$ for any possible neighbor v'_j of v. Likewise, if $\varphi(v) = i'$ for some $i' \in I'$, then $v = v'_i$. Again, based on how the undirected graph G_u is defined, one has that the incident edges to vertex v'_i in G_u are either of the form (v_j, v'_i) (namely, when $(v_j, v_i) \in E_d$) or (v_i, v'_i). Thus, the neighbors of v'_i can be either v_j (for some $j \neq i$) or v_i, which in turn implies that $\varphi(v_j) = j$ for any possible neighbor v_j of v. Therefore, φ defines a prime labeling function on G_u. □

Figure 3 displays the correspondence between a directed graph G_d and its corresponding prime graph G_u. Notice that the prime labeled nodes encode for the notion of incoming edges of the corresponding directed graph.

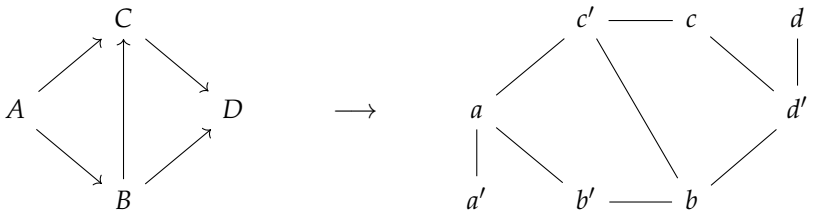


Figure 3. A directed graph G_d and its corresponding prime graph G_u.

Proposition 2. *Let G_{d_1} and G_{d_2} be two directed graphs in **DGraph**, and let $V_{d_1} \xrightarrow{f} V_{d_2}$ be a directed graph morphism. Then, f induces a prime graph morphism $\bar{f}$ between the prime graphs G_{u_1} and G_{u_2}.*

Proof of Proposition 2. Let $G_{d_1}, G_{d_2} \in$ **DGraph**, and let us take $f : V_{d_1} \longrightarrow V_{d_2}$, a directed graph morphism. Without loss of generality, assume that the vertex sets are $V_{d_1} = \{v_1, v_2, \cdots, v_n\}$ and $V_{d_2} = \{w_1, w_2, \cdots, w_m\}$, respectively. Then, by Proposition 1, the vertex set of the corresponding prime graphs G_{u_1} and G_{u_2} are given by

$$V_{u_1} = \{v_1, v_2, \cdots, v_n, v'_1, v'_2, \cdots, v'_n\}$$

and

$$V_{u_2} = \{w_1, w_2, \cdots, w_m, w'_1, w'_2, \cdots, w'_n\},$$

respectively. With this in mind, we define $\bar{f} : V_{u_1} \longrightarrow V_{u_2}$ by

$$\bar{f} = \begin{cases} \bar{f}(v_i) = f(v_i) & \text{for } v_i \in \{v_1, v_2, \cdots, v_n\} \\ \bar{f}(v'_i) = f(v_i)' & \text{for } v'_i \in \{v'_1, v'_2, \cdots, v'_n\}, \end{cases}$$

where $f(v_i)' = w'_k$ is the prime labeled vertex in V_{u_2} such that $f(v_i) = w_k$. We claim that $\bar{f}$ is a prime graph morphism. Indeed, for any edge $(v_i, v'_j) \in E_{u_1}$, with $i \neq j$, it corresponds to a directed edge (v_i, v_j) in E_{d_1}. Thus, as f preserves adjacencies—in the directed case—we have that $(f(v_i), f(v_j)) \in E_{d_2}$, which in turn implies that $(f(v_i), f(v_j)') \in E_{u_2}$. Considering this, one has that

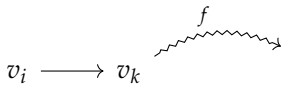
$$\bar{f}(v_i, v'_j) = \left(\bar{f}(v_i), \bar{f}(v'_j)\right) = \left(f(v_i), f(v_j)'\right) \in E_{u_2}.$$

Moreover, for edges of the form $(v_i, v'_i) \in E_{u_1}$, one obtains

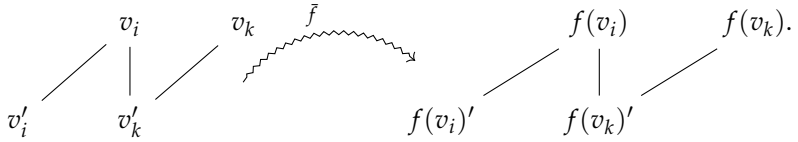
$$\bar{f}(v_i, v'_i) = \left(\bar{f}(v_i), \bar{f}(v'_i)\right) = \left(f(v_i), f(v_i)'\right) \in E_{u_2}.$$

Thus, $\bar{f}$ preserves adjacencies. Now, based on how $\bar{f}$ is defined, it is clear that $\bar{f}$ is compatible with the labeling, as it preserves the prime and non-prime labelings: $\left(\bar{f}(v_i)\right)' = f(v_i)' = \bar{f}(v'_i)$. Therefore, $\bar{f}$ is a prime graph morphism. □

To better visualize Proposition 2, let us consider a directed graph morphism f that maps a directed edge $(v_i, v_k) \in E_{d_1}$ into a directed edge $(f(v_i), f(v_k)) \in E_{d_2}$:



Then, in the prime graph context, one has a prime graph morphism $\bar{f}$ mapping



Remark 1. *If $G_d = (V_d, E_d)$ is a directed graph, then its corresponding prime graph $G_u = (V_u, E_u)$ satisfies that*

$$|E_u| = |E_d| + |V_d|.$$

This follows from the fact that an edge in G_u has either the form (v_i, v'_j)—corresponding to a directed edge (v_i, v_j) in G_d—or the form (v_i, v'_i).

Considering the results from above, we have the following:

Theorem 2. *The map $\mathcal{L}$: **DGraph** $\longrightarrow$ **PGraph** that assigns to each object $G_d \in$ **DGraph** the object $G_u \in$ **PGraph** and to each morphism $f \in$ **DGraph** the morphism $\bar{f} \in$ **PGraph** defines a functor from **DGraph** to **PGraph**.*

Proof Theorem 2. Observe that the object and morphism assignments of functor $\mathcal{L}$: **DGraph** $\longrightarrow$ **PGraph** are exactly Proposition 1 and Proposition 2, respectively. Thus, it suffices to show that $\mathcal{L}$ preserves identity morphisms and composition of morphisms.

The fact that $\mathcal{L}$ preserves identity morphisms follows from Proposition 2 as $\overline{Id}$ coincides with the identity map Id. To see that $\mathcal{L}$ preserves compositions, we will show that, given the directed graph morphisms $f : V_{d_1} \longrightarrow V_{d_2}$ and $g : V_{d_2} \longrightarrow V_{d_3}$, one has that $\overline{g \circ f} = \bar{g} \circ \bar{f}$. Following the notation so far, we will denote the vertex set of the directed graph G_{d_1} by $V_{d_1} = \{v_1, v_2, \cdots, v_n\}$, the vertex set of the directed graph G_{d_2} by $V_{d_2} = \{w_1, w_2, \cdots, w_m\}$, and the vertex set of the directed graph G_{d_3} by $V_{d_3} = \{z_1, z_2, \cdots, z_l\}$. Then, by Proposition 10, the vertex sets of their corresponding prime graphs are given by

$$V_{u_1} = \{v_1, v_2, \cdots, v_n, v'_1, v'_2, \cdots, v'_n\},$$

$$V_{u_2} = \{w_1, w_2, \cdots, w_m, w'_1, w'_2, \cdots, w'_m\},$$
$$\text{and } V_{u_3} = \{z_1, z_2, \cdots, z_l, z'_1, z'_2, \cdots, z'_l\},$$

respectively.

Now, on the one hand, the image under functor $\mathcal{L}$ of the directed graph morphism defined by the composition $(g \circ f)$ is the prime graph morphism $\overline{(g \circ f)} : V_{u_1} \longrightarrow V_{u_3}$, whose correspondence rule is given by

$$\overline{(g \circ f)} = \begin{cases} \overline{(g \circ f)}(v_i) = (g \circ f)(v_i) & \text{for } v_i \in \{v_1, v_2, \cdots, v_n\} \\ \overline{(g \circ f)}(v'_i) = \big((g \circ f)(v_i)\big)' & \text{for } v'_i \in \{v'_1, v'_2, \cdots, v'_n\}. \end{cases}$$

Here, $\big((g \circ f)(v_i)\big)' = z'_r$ denotes the prime labeled vertex on V_{d_3} such that $(g \circ f)(v_i) = z_r$.

On the other hand, the image of f under $\mathcal{L}$ is the prime graph morphism $\bar{f} : V_{u_1} \longrightarrow V_{u_2}$ given by

$$\bar{f} = \begin{cases} \bar{f}(v_i) = f(v_i) & \text{for } v_i \in \{v_1, v_2, \cdots, v_n\} \\ \bar{f}(v'_i) = f(v_i)' & \text{for } v'_i \in \{v'_1, v'_2, \cdots, v'_n\}, \end{cases}$$

while the image of g under $\mathcal{L}$ is the prime graph morphism $\bar{g} : V_{u_2} \longrightarrow V_{u_3}$ given by

$$\bar{g} = \begin{cases} \bar{g}(w_i) = f(w_i) & \text{for } w_i \in \{w_1, w_2, \cdots, w_m\} \\ \bar{g}(w'_i) = f(w_i)' & \text{for } w'_i \in \{w'_1, w'_2, \cdots, w'_m\}. \end{cases}$$

Note that, as prime graph morphisms preserve the prime vertex and non-prime vertex labelings, one then has that $\bar{f}(v_i) \in \{w_1, \cdots, w_m\}$ and $\bar{f}(v'_i) \in \{w'_1, \cdots, w'_m\}$. This way, when considering the composite morphism $\bar{g} \circ \bar{f} : V_{u_1} \longrightarrow V_{u_3}$, one has that, for each $v_i \in \{v_1, v_2, \cdots, v_n\}$,

$$(\bar{g} \circ \bar{f})(v_i) = \bar{g}(\bar{f}(v_i)) = \bar{g}(f(v_i)) = g(f(v_i)) = (g \circ f)(v_i) = \overline{(g \circ f)}(v_i),$$

whereas for each $v'_i \in \{v'_1, v'_2, \cdots, v'_n\}$,

$$(\bar{g} \circ \bar{f})(v'_i) = \bar{g}(\bar{f}(v'_i)) = \bar{g}(f(v_i)') = \big(g(f(v_i))\big)' = \big((g \circ f)(v_i)\big)' = \overline{(g \circ f)}(v'_i).$$

Hence, $\overline{(g \circ f)}$ and $(\bar{g} \circ \bar{f})$ have the same correspondence rule, and hence, $\overline{(g \circ f)} = (\bar{g} \circ \bar{f})$. Consequently,

$$\mathcal{L}(g \circ f) = \overline{(g \circ f)} = \bar{g} \circ \bar{f} = \mathcal{L}(g) \circ \mathcal{L}(f),$$

that is, $\mathcal{L}$ preserves compositions of morphisms. □

Remark 2. *The functor $\mathcal{L}$ preserves topological features. For instance, connectivity is preserved under $\mathcal{L}$. Also, if G_d is a complete directed graph, then its corresponding prime graph $\mathcal{L}(G_d)$ is a complete bipartite graph.*

On the other end, the next two propositions describe the assignment on objects and morphisms of functor $\mathcal{M}$: **PGraph** $\longrightarrow$ **DGraph**:

Proposition 3. *Let G_u be a prime graph. Then, G_u induces a simple directed graph G_d.*

Proof of Proposition 3. Let G_u be a prime graph, and let $\varphi_I : V_u \longrightarrow I \cup I'$ be its labeling function, with $I = \{1, \cdots, n\}$. If the vertex set of G_u is given by

$$V_u = \{v_1, v_2, \cdots, v_n, v'_1, v'_2, \cdots, v'_n\},$$

then, we define the vertex set of the directed graph G_d, induced by G_u, as

$$V_d = \{v_1, v_2, \cdots, v_n\}.$$

In order to define the edge set E_d, we must consider all incident edges to the set of prime label vertices v_i' in V_u. For instance, if v_j' is adjacent to v_i in G_u (with $j \neq i$), then we will obtain a directed edge in G_d from vertex v_i towards vertex v_j. In case v_i' is the only adjacent vertex to v_i in G_u, then the corresponding vertex v_i in G_d will not have incoming directed edges. In other words, the set E_d is given by

$$E_d = \{(v_i, v_j) \mid \text{ whenever } i \neq j \text{ and } (v_i, v_j') \in E_u\}$$

□

Proposition 4. *Let $G_{u_1}, G_{u_2} \in$ **PGraph** and let $g : V_{u_1} \longrightarrow V_{u_2}$ be a prime graph morphism. Then, g induces a directed graph morphism $\tilde{g}$ between their corresponding directed graphs G_{d_1} and G_{d_2}.*

Proof of Proposition 4. Let $G_{u_1}, G_{u_2} \in$ **PGraph**, and let $f : V_{u_1} \longrightarrow V_{u_2}$ be a prime graph morphism. Without loss of generality, let us assume that $V_{u_1} = \{v_1, \cdots, v_n, v_1', \cdots, v_n'\}$ and $V_{u_2} = \{w_1, \cdots, w_m, w_1', \cdots, w_m'\}$. Then, by Proposition 3, the vertex sets V_{d_1} and V_{d_2} are given by $\{v_1, \cdots, v_n\}$ and $\{w_1, \cdots, w_m\}$, respectively. With this in mind, we define the map $\tilde{f} : V_{d_1} \longrightarrow V_{d_2}$ as follows. For each $v_i \in V_{d_1}$, we set $\tilde{f}(v_i) = f(v_i)$. We claim that $\tilde{f}$ preserves adjacencies. Indeed, given $(v_i, v_j) \in E_{d_1}$, we have that $(v_i, v_j') \in E_{u_1}$). Thus, as f is a prime graph morphism, we obtain that $(f(v_i), f(v_j')) \in E_{u_2}$. Moreover, since f is compatible with the prime and non-prime labelings, it follows that $f(v_j') = f(v_j)'$, from which we have that $(f(v_i), f(v_j)) \in E_{d_2}$. Therefore,

$$\tilde{f}(v_i, v_j) = (\tilde{f}(v_i), \tilde{f}(v_j)) = (f(v_i), f(v_j)) \in E_{d_2},$$

showing that $\tilde{f}$ is a directed graph morphism. □

Considering the above, we have the following:

Theorem 3. *The map $\mathcal{M}$: **PGraph** $\longrightarrow$ **DGraph** that assigns to each object $G_u \in$ **PGraph** the object $G_d \in$ **DGraph**, and to each morphism $f \in$ **PGraph** the morphism $\tilde{f} \in$ **DGraph** is a functor.*

Proof of Theorem 3. By Propositions 3 and 4, it suffices to show that the map $\mathcal{M}$ preserves identity morphisms and composition of morphisms.

The fact that $\mathcal{M}$ preserves identity morphisms follows from Proposition 4, as $\tilde{Id}$ coincides with the identity map Id. To see that $\mathcal{M}$ preserves compositions, we will prove that, given the prime graph morphisms $f : V_{u_1} \longrightarrow V_{u_2}$ and $g : V_{u_2} \longrightarrow V_{u_3}$, one has that $\widetilde{g \circ f} = \tilde{g} \circ \tilde{f}$. Without loss of generality, let us suppose that

$$\begin{aligned} V_{u_1} &= \{u_1, u_2, \cdots, u_n, u_1', u_2', \cdots, u_n'\}, \\ V_{u_2} &= \{v_1, v_2, \cdots, v_m, v_1', v_2', \cdots, v_m'\} \text{ and} \\ V_{u_3} &= \{w_1, w_2, \cdots, w_l, w_1', w_2', \cdots, w_l'\}. \end{aligned}$$

Following along with the notation, we will denote the vertex sets of the directed graphs G_{d_1}, G_{d_2} and G_{d_3} by $V_{d_1} = \{u_1, u_2, \cdots, u_n\}$, $V_{u_2} = \{v_1, v_2, \cdots, v_m\}$ and $V_{u_3} = \{w_1, w_2, \cdots, w_l\}$, respectively. Now, on the one hand, the image under $\mathcal{M}$ of the prime graph morphism $g \circ f$ is the directed graph morphism $\widetilde{(g \circ f)} : V_{d_1} \longrightarrow V_{d_3}$, whose correspondence rule is given by

$$\widetilde{(g \circ f)}(u_i) = (g \circ f)(u_i), \forall u_i \in V_{d_1}.$$

On the other hand, the directed graph morphism $\tilde{f} : V_{d_1} \longrightarrow V_{d_2}$ is defined by $\tilde{f}(u_i) = f(u_i)$, for all $u_i \in V_{d_1}$. Likewise, the directed graph morphism $\tilde{g} : V_{d_2} \longrightarrow V_{d_3}$ is defined by $\tilde{g}(v_j) = g(v_j)$, for all $v_j \in V_{d_2}$. Therefore, the composition $(\tilde{g} \circ \tilde{f}) : V_{d_1} \longrightarrow V_{d_3}$ is a directed graph morphism such that, for each vertex $u_i \in \{u_1, u_2, \cdots, u_n\}$,

$$(\tilde{g} \circ \tilde{f})(u_i) = \tilde{g}(\tilde{f}(u_i)) = \tilde{g}(f(u_i)) = g(f(u_i)) = (g \circ f)(u_i) = \widetilde{(g \circ f)}(u_i).$$

The above shows that $(\tilde{g} \circ \tilde{f})$ and $\widetilde{(g \circ f)}$ have the same correspondence rule. Since both functions have the same domain and codomain, we obtain that $(\tilde{g} \circ \tilde{f}) = \widetilde{(g \circ f)}$. Therefore,

$$\mathcal{M}(g \circ f) = \widetilde{(g \circ f)} = \tilde{g} \circ \tilde{f} = \mathcal{M}(g) \circ \mathcal{M}(f),$$

that is, $\mathcal{M}$ preserves compositions of morphisms. □

2.1.2. The Functors $\mathcal{L}$ and $\mathcal{M}$ Are Isomorphisms

Recall that an isomorphism between two categories $\mathcal{C}$ and $\mathcal{D}$ is a functor $T : \mathcal{C} \longrightarrow \mathcal{D}$ that is a bijection on both objects and morphisms. In other words, a functor $T : \mathcal{C} \longrightarrow \mathcal{D}$ is an isomorphism if, and only if, there exists a functor $S : \mathcal{D} \longrightarrow \mathcal{C}$ for which the compositions $T \circ S$ and $S \circ T$ are the identity functors $Id_{\mathcal{D}}$ and $Id_{\mathcal{C}}$, respectively. In this case, we say that the categories $\mathcal{C}$ and $\mathcal{D}$ are isomorphic.

Proposition 5. *Let G_d be a directed graph in **DGraph**. Then*

$$(\mathcal{M} \circ \mathcal{L})(G_d) = G_d.$$

Proof of Proposition 5. Let $G_d \in$ **DGraph**. Without loss of generality, we can assume that $V_d = \{v_1, \cdots, v_n\}$. Then, the prime graph $\mathcal{L}(G_d) = G_u$ has the vertex set

$$V_u = \{v_1, v_2, \cdots, v_n, v_1', v_2', \cdots, v_n'\},$$

and the edge set

$$E_u = \begin{cases} (v_i, v_j') & \text{if } i \neq j \text{ and } (v_i, v_j) \in E_d \\ (v_i, v_i') & \text{for each } v_i \in V_d. \end{cases}$$

If we now consider the directed graph induced by G_u, that is, $\mathcal{M}(G_u) = G_d^*$, then, the vertex set V_d^* is given by $\{v_1, \cdots, v_n\}$ and its edge by

$$E_d^* = \{(v_i, v_j) \mid i \neq j \text{ and } (v_i, v_j') \in E_u\}.$$

In this way, as $(v_i, v_j') \in E_u$ if, and only if, $(v_i, v_j) \in E_d$, it follows that $(v_i, v_j) \in E_d^*$ if, and only if, $(v_i, v_j) \in E_d$. Moreover, since G_d and G_d^* have the same vertex set $\{v_1, v_2, \cdots, v_n\}$, we can conclude that $G_d = G_d^*$. Therefore,

$$(\mathcal{M} \circ \mathcal{L})(G_d) = G_d.$$

□

Proposition 6. *Let $G_{d_1}, G_{d_2} \in$ **DGraph**, and let $f : V_{d_1} \longrightarrow V_{d_2}$ be a directed graph morphism. Then,*

$$(\mathcal{M} \circ \mathcal{L})(f) = f.$$

Proof of Proposition 6. Let G_{d_1} and G_{d_2} be directed graphs, and let $f : V_{d_1} \longrightarrow V_{d_2}$ be a directed graph morphism. Without loss of generality, assume that $V_{d_1} = \{v_1, \cdots, v_n\}$. Then, by Theorem 2, we obtain the prime graphs $\mathcal{L}(G_{d_1}) = G_{u_1}$ and $\mathcal{L}(G_{d_2}) = G_{u_2}$, along with the prime graph morphism $\mathcal{L}(f) = \bar{f}$ defined by

$$\bar{f} = \begin{cases} \bar{f}(v_i) = f(v_i) & \text{for } v_i \in \{v_1, v_2, \cdots, v_n\} \\ \bar{f}(v_i') = f(v_i)' & \text{for } v_i' \in \{v_1', v_2', \cdots, v_n'\}. \end{cases}$$

Here, $V_{u_1} = \{v_1, v_2, \cdots, v_n, v_1', v_2', \cdots, v_n'\}$. Now, as

$$\mathcal{M}(G_{u_i}) = \mathcal{M}(\mathcal{L}(G_{d_i})) = (\mathcal{M} \circ \mathcal{L})(G_{d_i}) = G_{d_i},$$

for $i = 1, 2$, it suffices to show that the functions $\mathcal{M}(\bar{f})$ and f have the same correspondence rule, since both have the same domain and codomain sets. By considering $\mathcal{M}(\bar{f}) = \widetilde{\bar{f}}$, we see that $\widetilde{\bar{f}}(v_i) = \bar{f}(v_i) = f(v_i)$, for each vertex $v_i \in \{v_1, \cdots, v_n\}$. Therefore, $\mathcal{M}(\bar{f}) = f$, and thus,

$$(\mathcal{M} \circ \mathcal{L})(f) = f.$$

□

Corollary 1. *The functor* $\mathcal{L} : \textbf{\textit{DGraph}} \longrightarrow \textbf{\textit{PGraph}}$ *and the functor* $\mathcal{M} : \textbf{\textit{PGraph}} \longrightarrow \textbf{\textit{DGraph}}$ *satisfy*

$$(\mathcal{M} \circ \mathcal{L}) = Id_{\textbf{\textit{DGraph}}}.$$

On the other end, we have the following results:

Proposition 7. *Let* G_u *be a prime graph. Then*

$$(\mathcal{L} \circ \mathcal{M})(G_u) = G_u.$$

Proof of Proposition 7. Let $G_u \in$ **PGraph**. Without loss of generality, assume that the vertex set V_u is given by the set $\{v_1, v_2, \cdots, v_n, v_1', v_2', \cdots, v_n'\}$. Then $G_d = \mathcal{M}(G_u)$ is the directed graph with vertex set $G_d = \{v_1, v_2, \cdots, v_n\}$, and edge set

$$E_d = \{(v_i, v_j) \mid i \neq j \text{ and } (v_i, v_j') \in E_u\}.$$

Now, if we denote by G_u^* the prime graph induced by G_d, that is, $\mathcal{L}(G_d) = G_u^*$, then, $V_d^* = \{v_1, \cdots, v_n, v_1', \cdots, v_n'\}$, and

$$E_u^* = \begin{cases} (v_i, v_j') & \text{if } i \neq j \text{ and } (v_i, v_j) \in E_d \\ (v_i, v_i') & \text{for each } v_i \in V_d. \end{cases}$$

Notice that $(v_i, v_j') \in E_u^*$ if, and only if, $(v_i, v_j) \in E_d$, which holds if, and only if, $(v_i, v_j') \in E_u$. Since both graphs G_u and G_u^* have as vertex set the set $\{v_1, v_2, \cdots, v_n, v_1', \cdots, v_n'\}$, we conclude that $G_u = G_u^*$. Therefore,

$$(\mathcal{L} \circ \mathcal{M})(G_u) = G_u.$$

□

Proposition 8. *Let* $f : V_{u_1} \longrightarrow V_{u_2}$ *be a prime graph morphism. Then*

$$(\mathcal{L} \circ \mathcal{M})(f) = f.$$

Proof of Proposition 8. Let $f : V_{u_1} \longrightarrow V_{u_2}$ be a prime graph morphism, and let us suppose that $V_{u_1} = \{v_1, \cdots, v_n, v_1', \cdots, v_n'\}$. By Theorem 3, we have the directed graphs $\mathcal{M}(G_{u_1}) = G_{d_1}$ and $\mathcal{M}(G_{u_2}) = G_{d_2}$, along with the directed graph morphism $\mathcal{M}(f) = \tilde{f} : V_{d_1} \longrightarrow V_{d_2}$ defined by $\tilde{f}(v_i) = f(v_i)$, for each $i = 1, \cdots, n$. Now, as

$$\mathcal{L}(G_{d_i}) = \mathcal{L}(\mathcal{M}(G_{u_i})) = (\mathcal{L} \circ \mathcal{M})(G_{u_i}) = G_{u_i}$$

for $i = 1, 2$, it suffices to show that $\mathcal{L}(\tilde{f})$ and f have the same correspondence rule, as these have the same domain and codomain sets. By considering that $\mathcal{L}(\tilde{f}) = \bar{\tilde{f}}$ is such that

$$\bar{\tilde{f}} = \begin{cases} \bar{\tilde{f}}(v_i) = \tilde{f}(v_i) = f(v_i) & \text{for } v_i \in \{v_1, v_2, \cdots, v_n\} \\ \bar{\tilde{f}}(v_i') = \tilde{f}(v_i)' = f(v_i)' & \text{for } v_i' \in \{v_1', v_2', \cdots, v_n'\}, \end{cases}$$

we see that $\mathcal{L}(\tilde{f})$ and f have the same correspondence rule. Thus, $\mathcal{L}(\tilde{f}) = f$, which in turn implies that

$$(\mathcal{L} \circ \mathcal{M})(f) = f.$$

□

Corollary 2. *The functor* $\mathcal{L}$: ***DGraph*** $\longrightarrow$ ***PGraph*** *and the functor* $\mathcal{M}$: ***PGraph*** $\longrightarrow$ ***DGraph*** *satisfy*

$$(\mathcal{L} \circ \mathcal{M}) = Id_{\textbf{PGraph}}.$$

Therefore, by Corollaries 1 and 2, we have the following:

Theorem 4. *The categories* ***DGraph*** *and* ***PGraph*** *are isomorphic.*

We highlight that the property of functoriality on graph morphisms is what allowed us to extend network alignment techniques for undirected graphs to directed graphs. As previously mentioned, an alignment of two networks consists of a mapping between the nodes of the compared networks. In doing this, one aims to preserve as much of the structure (or topology) between the considered networks as possible. Thus, when transforming two directed graphs into their corresponding prime graphs, the way we relate these prime graphs is by defining a prime graph morphism. This prime graph morphism preserves the topology and the labeling conditions, which ultimately gives the notion of direction. Consequently, by functoriality, these prime graph morphisms correspond to the directed graph morphisms used when aligning the compared directed graphs.

2.2. Prime Transformation Algorithm

This section outlines an algorithm (Algorithm 1) to convert a directed graph to a prime graph, reflecting definitions and constructions found in the previous section. Then, we discuss the time and space complexity for creating and storing a prime graph from a directed graph. Note that the algorithm does not consider edge weights because weighting schemes can vary based on application.

The input to Algorithm 1 is a directed graph, and the output is its corresponding prime graph. The algorithm operates on each edge of the directed graph. As such, the algorithm's time complexity is $O(n + e)$, where e is the number of edges and n is the number of nodes in the directed graph. While additional space is required to store the prime graph, the space necessary scales according to $O(n + e)$. The extra space needed to store a prime graph results from the additional nodes and edges it contains relative to its directed graph counterpart. For prime graphs, the number of nodes is always double that of directed graphs because each node in the directed graph spawns an additional prime node. Additionally, an edge is created between each pair of prime and non-prime nodes; therefore, additional n edges are required.

Algorithm 1 Prime Transformation Algorithm

Input DGraph
Output PGraph

procedure MAKEPGRAPH
  $dirEdges \leftarrow DGraph.edges$ ▷ Initialize edge list
  $dirNodes \leftarrow DGraph.nodes$ ▷ Initialize node list
  $PGraph \leftarrow$ Empty Graph ▷ Initialize an empty prime graph
  for n in dirNodes **do**:
    $nonPrimeNode \leftarrow n.label$ ▷ store the label of node n
    $primeNode \leftarrow n.label + 'p'$ ▷ create a label for the n's prime node
    $PGraph \leftarrow AddNode(primeNode)$ ▷ add prime node
    $PGraph \leftarrow AddNode(nonPrimeNode)$ ▷ add non-prime node
    $PGraph \leftarrow AddEdge(primeNode, nonPrimeNode)$ ▷ add an edge between prime and non-prime node pairs
  for e in dirEdges **do**:
    $source \leftarrow e.initial.label$ ▷ store edge head label
    $target \leftarrow e.terminal.label$ ▷ store edge tail label
    $primeNode \leftarrow e.terminal.label + 'p'$ ▷ store the tail node's prime label
    $PGraph \leftarrow AddEdge(source, primeNode)$ ▷ add edge to PGraph using labels
    $PGraph \leftarrow AddEdge(primeNode, target)$ ▷ add edge to PGraph using labels
  return PGraph

3. Numerical Results

This section contains the two applications of the isomorphisms between the category of simple directed graphs **DGraph** and the category of prime graphs **PGraph**. To perform the computations, we relied on two things: the explicit description of the objects and morphisms of the category **PGraph**, and the explicit definition of functors $\mathcal{L}$ and $\mathcal{M}$.

3.1. Network Alignment

Network alignment is a technique that consists of mapping the nodes of the compared networks in such a way that the structure, or topology, is preserved as much as possible. While mapping the nodes, the alignment looks to maximize node similarity and preserve the edge topology. Network alignment algorithms use an objective function—which is defined over candidate alignments—to find the best possible alignment between network pairs. This objective function represents a score, usually based on *node and edge similarity*. By considering this score, the aligner follows a "search algorithm" to find the fittest network map that maximizes the objective function. For a deeper understanding of network alignment and its applications, we recommend the reader see [13–15].

3.1.1. Synthetic Network Model

To demonstrate the efficacy of our correspondence between directed graphs and prime graphs, we generated a synthetic dataset that allowed us to precisely control the ground truth network similarity by generating correlated graph pairs. In this way, by using a network aligner, we calculated the network similarity scores between pairs of correlated prime graphs that were generated from pairs of correlated directed graphs (where correlation values for each pair were known). Our results indicate that the alignment scores between pairs of prime graphs are closely related to the correlation coefficients used to generate the corresponding directed graphs.

To generate the synthetic dataset, we used a stochastic block model (SBM) network topology [20] with the following parameters and values. The directed network consisted of 150 total nodes, organized into 3 blocks with 50 nodes per block, as well as 10 to 20 percent connection density inter-block and 1 percent intra-block connection density. For each discrete value of graph pair correlation coefficients ρ, we generated 50 pairs of directed graphs. It is the correlation coefficient ρ that determines the similarity between graphs. This

way, we first created a directed graph from an edge probability matrix, which is defined by the previously mentioned inter- and intra-block connection density, and then, we use the coefficient ρ to adjust the probability matrix for a second directed graph in the pair. Observe that the closer ρ is to 1, the more similar the probability matrices will be, which means that it is more likely that the two generated graphs will be similar. Our results iterate through discrete values of ρ in the range of $[0.5, 1]$ at 0.1 increments.

In summary, after generating all the directed graph pairs, we converted them to prime graphs, ran the network alignment algorithm for each prime graph pair, and plotted the similarity score as a function of ρ using the highest similarity score out of the 5 alignment attempts per pair.

3.1.2. Numerical Result

We used both edge and node similarity scores to optimize network alignments. The optimizer uses a cost function that consists of an equally weighted linear combination of edge and node similarity scores. We used EC, ICS, and S3 edge similarity scores along with a graphlet orbit degree-based node similarity score (see the references [13–15,21,22] for details on each score's definition). Finally, we used SANA [13], an off-the-shelf network aligner, and the aforementioned similarity scores for alignment optimization.

Figure 4 is a box and whisker plot (see [23]), which contains empirical alignment scores for varying values of the graph correlation coefficient. When the graph pairs are perfectly correlated ($\rho = 1$), that is, they are exact copies, but the mapping between nodes is unknown, the alignment tool finds the perfect mapping between the graphs. As the ρ values decrease, the alignment scores decrease. The variance in the alignment scores primarily results from the network alignment algorithm's random initialization, stochastic optimization routines, finite number of optimization steps, as well as because the optimization function is not guaranteed to be convex. A secondary source of alignment score variability is from the stochastic graph pair generation process (see Section 3.1.1). Variation in alignment scores results in an R^2 [24] value of 0.98 with a corresponding p-value of 9×10^{-6}. These results indicate that calibrating for the slope yields a usable network alignment metric.

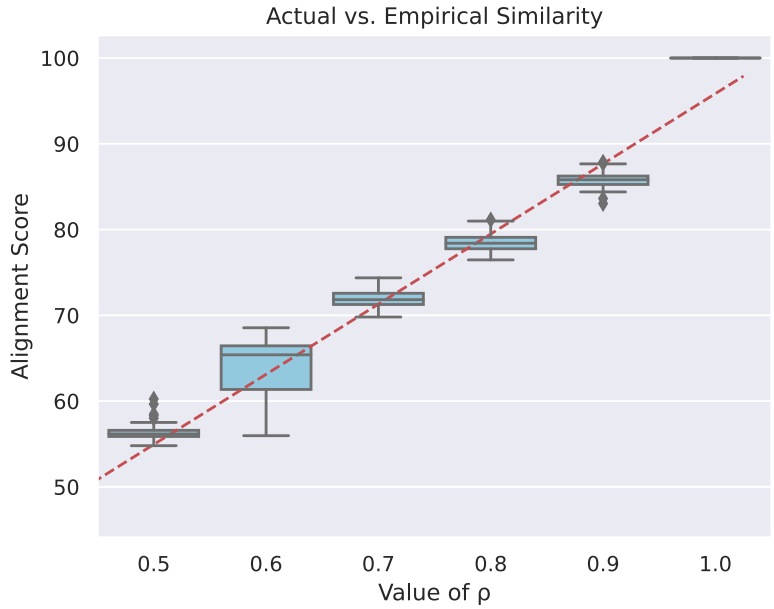


Figure 4. Box and whisker plot of network alignment scores of prime graphs as a result of varying the underlying similarity between directed graph pairs. Each column in the plot corresponds to a specific value of ρ, which is the correlation coefficient between a generated pair of graphs. The red line is the linear regression of all of the boxes together with respect to the ρ values. It has an R^2 value of 0.98 with a corresponding p-value of 9×10^{-6}.

3.2. Minimum Cuts and Spectral Clustering in Prime Graphs

We start by showing the connection between cuts in a directed graph and cuts in its corresponding prime graph. Afterwards, we see that the functorial correspondence preserves clusters. This will allow us to use spectral clustering and show that a weighted prime graph maintains the same minimum cuts as its directed graph counterpart.

Recall that a cut in an undirected graph $G_u = (V_u, E_u)$ is a partition of the vertex set V_u into a subset S and its complement $\bar{S} = V_u \setminus S$. Likewise, one defines a cut in a directed graph $G_d = (V_d, E_d)$ as a partition of the vertex set $V_d = S \dot{\cup} \bar{S}$. Now, for each weighted directed graph $G_d = (V_d, E_d)$ with weight values given by $\{w_{v_i,v_j}\}_{(v_i,v_j)\in E_d}$, there is a weighted prime graph $G_u = (V_u, E_u)$ whose weight values are given as follows. For each edge in G_u of the form (v_i, v_j'), we set the weight value $w'_{v_i,v_j'}$ equal to $(w_{v_i,v_j})/2$. For the remaining edges in G_u of the form (v_i, v_i'), we give a weight value $w'_{v_i,v_i'}$ satisfying

$$w'_{v_i,v_i'} > \sum_{v_k|(v_k,v_i')\in E_u} \frac{w'_{v_k,v_i'}}{2}.$$

With this in mind, we have a correspondence between cuts in G_d and cuts in G_u with lesser volume. Recall that, given a weighted undirected graph G_u, we define the volume of a cut $(C, \bar{C})$ (where $\bar{C} = V_u \setminus C$) as

$$vol_u(\partial C) = \sum_{x\in C, y\in \bar{C}} w_{x,y}.$$

Notice that $vol_u(\partial C)$ considers those edges whose endpoints belong to different components C and $\bar{C}$. Furthermore, as the graph G_u is undirected, it follows that $vol_u(\partial C) = vol_u(\partial \bar{C})$. Considering this, for the weighted prime graph G_u, with weight values described as above, we have the following:

Proposition 9. *Any cut in the weighted prime graph G_u can be reduced to a cut of the form $\big((S \cup S'), (\bar{S} \cup \bar{S}')\big)$ with lower volume. Here, S and $\bar{S}$ are subsets that partition the non-prime nodes of V_u, and S' and $\bar{S}'$ denote their prime node counterparts, which partition the set of prime nodes of V_u.*

Proof of Proposition 9. Suppose that $(C, \bar{C})$ is a cut of the prime graph G_u such that $v_i \in C$ and $v_i' \in \bar{C}$. This implies that the weight of the edge (v_i, v_i') adds to the volume of the cut. Now, if we consider the cut defined by the sets $C \cup \{v_i'\}$ and $\bar{C} \setminus \{v_i'\}$, the weight of the edge (v_i, v_i') will no longer add to the volume of the cut; instead, we will be adding those weights corresponding to the edges that are adjacent to v_i', which turns out to be at most

$$\sum_{v_k|(v_k,v_i')\in E_u} w_{v_k,v_i'} < w_{v_i,v_i'}.$$

By continuing with this process, we will obtain a cut of the form $\big((S \cup S'), (\bar{S} \cup \bar{S}')\big)$, where S and $\bar{S}$ are subsets of V_u that partition the non-prime vertices, whereas S' and $\bar{S}'$ are their corresponding prime nodes which partition the prime vertices of V_u. Note that, by construction, this last cut has lower volume than the initial cut $(C, \bar{C})$. □

The above proposition shows that the cuts with less volume in the weighted prime graph G_u are those with form $\big((S \cup S'), (\bar{S} \cup \bar{S}')\big)$. These cuts, in turn, are clearly in a one-to-one correspondence between cuts of the form $(S, \bar{S})$ in the directed graph G_d. We now show that the volume of a directed graph G_d is preserved when transformed to its corresponding weighted prime graph G_u.

Definition 4. *Let $G_d = (V_d, E_d)$ be a directed graph. A circulation on G_d is a function $F : E_d \longrightarrow \mathbb{R}^+ \cup \{0\}$ that assigns each directed edge to a non-negative value such that*

$$\sum_{\substack{u\in V_d\\(u,v)\in E_d}} F\big((u,v)\big) = \sum_{\substack{w\in V_d\\(v,w)\in E_d}} \big((v,w)\big),$$

for every vertex $v \in V_d$.

Intuitively, a circulation is a flow in the directed graph that is conserved at each vertex; that is, the flow into each vertex equals the flow out of each vertex. We define the volume crossing the cut $(S, \bar{S})$ in G_d as

$$vol_d(\partial S) = \sum_{\substack{u\in S\\v\in\bar{S}}} F(u,v).$$

Note that, as F is a circulation on G_d, we obtain that $vol_d(\partial(S)) = vol_d(\partial(\bar{S}))$.

We now prove that the volume crossing a cut $(S, \bar{S})$ in a directed graph G_d is the same as the volume of the cut $\big((S \cup S'), (\bar{S}, \bar{S}')\big)$ in its corresponding weighted prime graph G_d. To that end, we will assume that the weights of the directed graph G_d are given by a circulation $F : E_d \longrightarrow \mathbb{R}^+ \cup \{0\}$, and the weights of its corresponding prime graph G_u are given as where previously stated.

Proposition 10. *Let G_d be a weighted directed graph with weights given by a circulation F, and let G_u be its corresponding weighted prime graph. Then, for any cut $(S, \bar{S})$ of G_d, the corresponding cut $\big((S \cup S'), (\bar{S}, \bar{S}')\big)$ in the prime graph G_u satisfies*

$$vol_u\big(\partial(S \cup S')\big) = vol_d(\partial S).$$

Proof of Proposition 10. Let G_d be a weighted directed graph whose weights are given by the circulation $F : E_d \longrightarrow \mathbb{R}^+ \cup \{0\}$. In this case, $w_{u,v} = F(u,v)$. Let us denote by G_u its corresponding weighted prime graph, and by $w'_{i,j}$ its weight values. Now, for the sake of clarity, we will denote by $(S_d, \bar{S}_d)$ the cut in the weighted directed graph G_d, and denote by $\big((S \cup S'), (\bar{S}, \bar{S}')\big)$ the corresponding cut in the weighted prime graph G_u. Then, by definition, we have that

$$vol_u\big(\partial(S \cup S')\big) = \sum_{\substack{u\in S\cup S'\\v\in\bar{S}\cup\bar{S}'}} w'_{u,v}.$$

Notice that, by the way prime graphs are constructed, there are no edges between nodes in S and nodes in $\bar{S}$. The same can be said between nodes in S' and nodes in $\bar{S}'$. Hence,

$$vol_u\big(\partial(S \cup S')\big) = \sum_{\substack{u\in S,\\v\in\bar{S}'}} w'_{u,v} + \sum_{\substack{u\in S'\\v\in\bar{S}}} w'_{u,v} = \sum_{\substack{u\in S,\\v\in\bar{S}'}} w'_{u,v} + \sum_{\substack{u\in\bar{S}\\v\in S'}} w'_{u,v}$$

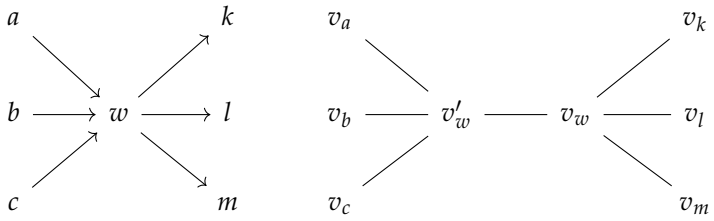
$$= \sum_{\substack{u\in S_d,\\v\in\bar{S}_d}} \frac{F(u,v)}{2} + \sum_{\substack{u\in\bar{S}_d\\v\in S_d}} \frac{F(u,v)}{2} = \frac{1}{2}\cdot\big(vol_d(\partial S) + vol_d(\partial\bar{S})\big)$$

$$= \frac{1}{2}\big(2\cdot vol_d(\partial S)\big) = vol_d(\partial S).$$

□

Consider now any vertex w in G_d lying in a cluster. Then, in its corresponding prime graph G_u, the nodes v_w and v'_w belong to the same cluster. This latter aspect follows because we have assigned a high weight value to the edge between v'_w and v_w in G_u. Now, from the

local topology around the vertex w in G_d, and the local topology around the vertices v_w and v'_w in G_u:



we can see that $deg(v'_w) = indeg(w) + 1$ and $deg(v_w) = outdeg(w) + 1$, where $indeg(w)$ and $outdeg(w)$ denote the indegree and outdegree of vertex $w \in V_d$, respectively. Further, observe that for any vertex in G_d within the same cluster as w, we obtain that its corresponding prime and non-prime nodes belong to the same cluster as v_w (and v'_w), as the functor $\mathcal{L}$ preserves connectivity. Therefore, by functoriality, each cluster in a directed graph G_d induces a cluster in its corresponding prime graph G_u with twice the number of nodes, namely the prime and non-prime vertices associated to the vertices of the cluster in the directed graph.

Proposition 11. *If C is a cluster in the directed graph G_d, then its corresponding cluster in $\mathcal{L}(G_d) = G_u$ has twice the number of nodes as C.*

Numerical Example: Spectrum of Graph Laplacian

For the technical application, we considered a method involving the graph Laplacian for optimal clustering. For further references, we recommend the reader to see [17,18].

To illustrate that prime graphs preserve the cuts of a simple directed graph, we numerically compute minimum cuts of directed graphs and prime graphs using the graph Laplacian. The directed graph consists of 1000 nodes split into two clusters. The first cluster consists of 450 nodes, and the second cluster consists of 550 nodes. The edge connectivity parameters of the directed graph are as follows. In the first cluster, the inter-cluster connectivity was 0.5, that is, there was a 50% chance of connection between nodes; in the second cluster, the inter-cluster connectivity was 0.4; finally, the intra-cluster connectivity was 0.1. Upon constructing the directed graph, we generated an associated prime graph. In Figure 5, we show the values of the sorted eigenvector associated with the second-smallest eigenvalue. This numerical example thus shows that the minimum cut is preserved between directed graphs and prime graphs.

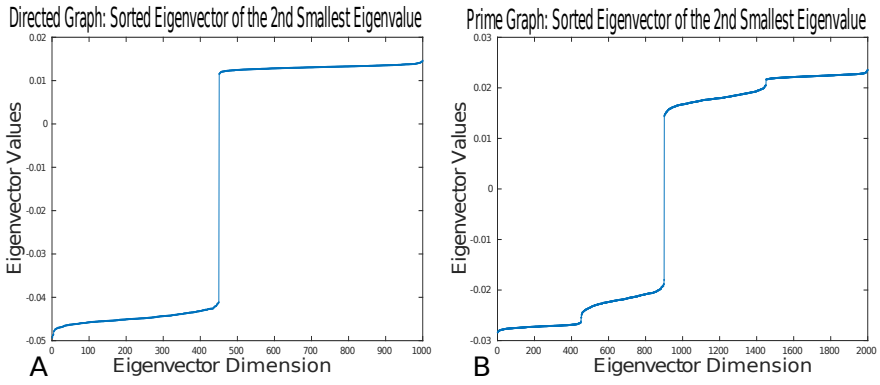


Figure 5. (**A**) Plot of the sorted values in the eigenvector associated with the second-smallest eigenvalue for the directed graph. (**B**) Plot of the sorted values in the eigenvector associated with the second-smallest eigenvalue for the prime graph derived from the directed graph generated in (**A**).

4. Discussion

This work shows a novel construction that reinforces the power of CT as a tool to formalize structures and their relations. In this case, we use CT to bridge a directed graph framework to an undirected graph framework, so that not only directionality is preserved but also several topological features. This bridge enables the use of undirected graph techniques to obtain information from systems that are represented as directed graphs. Both the computational and space complexity of the transformations are $O(N)$, where N is the number of nodes in the network. As an empirical demonstration, we provide a new option to perform network alignment for directed graphs. This is relevant since network alignment tools do not exist for a directed graph setting. Furthermore, our transformation does preserve network similarity between directed graphs and their prime graph counterparts; we attained an R^2 value of 0.98 (with a corresponding p-value of 9×10^{-6}) between the network aligner results, i.e., the similarity metric, a known graph generation correlation coefficient. Because we proved that our construction leads to an invertible transformation, there is only one prime graph that describes a simple directed graph and vice versa; as such, and in that sense, our transformation is error-free. Be that as it may, our transformation does not mitigate errors inherent in postprocessing the resultant graphs, for example, not achieving an R^2 value of 1 in the network alignment task.

Although the process of making an adjacency matrix of a directed graph symmetric is not new [19], nor is transforming a directed graph into an undirected graph [12], our framework is an advance. We proved that the minimum cuts are preserved when going from a directed graph framework to a prime graph framework and vice versa. These results, in turn, imply that clusters are preserved when moving from one setting to the other. As a proof of concept, we proved cluster preservation by generating a directed SBM network with known intra-block and inter-block connectivity.

While this work is a step towards a new application of existing network alignment tools, there is much left in this area to be explored in future work. Adoption of this technique may be limited by and rely upon showing additional mathematical proofs for commonly used techniques on graphs, for example, answering how the existing undirected node and edge similarity metrics might be skewed by the prime graph transformation. Another avenue for the application of prime graphs is to take advantage of their bipartite nature in problems such as the graph isomorphism problems for directed graphs. It is worthwhile studying the complexity of checking for equivalence between arbitrarily labeled **DGraph**s and **PGraphs**. Lastly, a categorical bridge, now between a multidirected graphs setting to a prime graph setting, might unlock new ways to study high complex data.

Author Contributions: Conceptualization, S.P.-G. and J.R.; methodology, S.P.-G. and V.K.G.; software, V.K.G. and V.M.; validation, S.P.-G., V.K.G., V.M., J.R. and G.A.S.; formal analysis, S.P.-G.; investigation, S.P.-G. and V.K.G.; resources, G.A.S.; data curation, V.K.G. and V.M.; writing—original draft preparation, S.P.-G.; writing—review and editing, S.P.-G. and G.A.S.; visualization, S.P.-G.; supervision, G.A.S.; project administration, G.A.S.; funding acquisition, G.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received unrestricted funds to the Center for Engineered Natural Intelligence at the University of California San Diego.

Data Availability Statement: All data and figures of this paper were generated synthetically. The code used in this work can be found in the link https://github.com/vgeorgeucsd/prime_graphs, accessed on 8 April 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Scott, J. Social network analysis. *Sociology* **1988**, *22*, 109–127. [CrossRef]
2. Dimitrova, T.; Petrovski, K.; Kocarev, L. Graphlets in Multiplex Networks. *Sci. Rep.* **2020**, *10*, 1928. [CrossRef] [PubMed]
3. Pržulj, N.; Wigle, D.A.; Jurisica, I. Functional topology in a network of protein interactions. *Bioinformatics* **2004**, *20*, 340–348. [CrossRef] [PubMed]

4. Shen-Orr, S.S.; Milo, R.; Mangan, S.; Alon, U. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.* **2022**, *31*, 64–68. [CrossRef] [PubMed]
5. Chu, B.K.; Tse, M.J.; Sato, R.R.; Read, E.L. Markov State Models of gene regulatory networks. *Syst. Biol.* **2017**, *11*, 1–17. [CrossRef] [PubMed]
6. Buibas, M.; Silva, G.A. A framework for simulating and estimating the state and functional topology of complex dynamic geometric networks. *Neural Comput.* **2011**, *23*, 183–214. [CrossRef] [PubMed]
7. Silva, G.A. The effect of signaling latencies and refractory node states on the dynamics of networks. *Neural Comput.* **2019**, *31*, 2492–2522. [CrossRef] [PubMed]
8. Fong, B.; Spivak, D.; Tuyéras, R. Backprop as functor: A compositional perspective on supervised learning. In Proceedings of the 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), Vancouver, BC, Canada, 24–27 June 2019; pp. 1–13.
9. Haruna, T. Theory of interface: Category theory, directed networks and evolution of biological networks. *Biosystems* **2013**, *114*, 125–148. [CrossRef]
10. Northoff, G.; Tsuchiya, N.; Saigo, H. Mathematics and the Brain: A Category Theoretical Approach to Go Beyond the Neural Correlates of Consciousness. *Entropy* **2019**, *21*, 1234. [CrossRef]
11. Otter, N.; Porter, M.A. A unified framework for equivalences in social networks. *arXiv* **2020**, arXiv:2006.10733.
12. Miller, G.L. Graph isomorphism, general remarks. *J. Comput. Syst. Sci.* **1979**, *18*, 128–142. [CrossRef]
13. Mamano, N.; Hayes, W.B. SANA: Simulated annealing far outperforms many other search algorithms for biological network alignment. *Bioinformatics* **2017**, *33*, 2156–2164. [CrossRef]
14. Vijayan, V.; Saraph, V.; Milenković, T. MAGNA++: Maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics* **2015**, *31*, 2409–2411. [CrossRef] [PubMed]
15. Sun, Y.; Crawford, J.; Tang, J.; Milenković, T. Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In Proceedings of the International Workshop on Algorithms in Bioinformatics, Atlanta, GA, USA, 10–12 September 2015; pp. 16–39.
16. Trung, H.T.; Toan, N.T.; Van Vinh, T.; Dat, H.T.; Thang, D.C.; Hung, N.Q.V.; Sattar, A. A comparative study on network alignment techniques. *Expert Syst. Appl.* **2020**, *140*, 112883. [CrossRef]
17. Chung, F.R.; Graham, F.C. *Spectral grAph Theory*; Number 92; American Mathematical Society: Providence, RI, USA, 1997.
18. Chung, F. Laplacians and the Cheeger inequality for directed graphs. *Ann. Comb.* **2005**, *9*, 1–19. [CrossRef]
19. Satuluri, V.; Parthasarathy, S. Symmetrizations for clustering directed graphs. In Proceedings of the 14th International Conference on Extending Database Technology, Uppsala, Sweden, 21–24 March 2011; pp. 343–354.
20. Chung, J.; Pedigo, B.D.; Bridgeford, E.W.; Varjavand, B.K.; Helm, H.S.; Vogelstein, J.T. GraSPy: Graph Statistics in Python. *J. Mach. Learn. Res.* **2019**, *20*, 1–7.
21. Hayes, W.B. An introductory guide to aligning networks using sana, the simulated annealing network aligner. In *Protein-Protein Interaction Networks*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 263–284.
22. Milenković, T.; Pržulj, N. Uncovering biological network function via graphlet degree signatures. *Cancer Inform.* **2008**, *6*, CIN–S680. [CrossRef]
23. Frigge, M.; Hoaglin, D.C.; Iglewicz, B. Some implementations of the boxplot. *Am. Stat.* **1989**, *43*, 50–54. [CrossRef]
24. Wright, S. Correlation and causation. *J. Agric. Res.* **1921**, *20*, 557–585.