frontiers

# Misc-GAN: A Multi-Scale Generative Model for Graphs

**Dawei Zhou**[†,*] **Lecheng Zheng**[†], **Jiejun Xu**[‡], **and Jingrui He**[†]

[†]*Arizona State University,* [‡]*HRL Laboratories, LLC.*

Correspondence*:
Jingrui He, jingrui.he@asu.edu

## ABSTRACT

Characterizing and modeling the distribution of a particular family of graphs are essential for studying real-world networks in a broad spectrum of disciplines, ranging from market-basket analysis to biology, from social science to neuroscience. However, it is unclear how to model the complex graph organizations and learn generative models from an observed graph. The key challenges come from the non-unique, high-dimensional nature of graphs, as well as the graph community structures at different granularity levels. In this paper, we propose a multi-scale graph generative model named *Misc-GAN*, which models the underlying distribution of the graph structures at different levels of granularity, and then 'transfers' such hierarchical distribution from the graphs in the domain of interest to a unique graph representation. The empirical results on both synthetic and real data sets demonstrate the effectiveness of the proposed framework.

## 1 INTRODUCTION

Graph is a fundamental tool for depicting and modeling complex systems in various domains, ranging from market-basket analysis to biology, from social science to neuroscience. Characterizing and modeling the distribution of a particular family of graphs are essential in many real-world applications. For example, in financial fraud detection, generative models are adopted to produce synthetic financial networks, when the empirical studies need to be conducted by the third parties without divulging the private information Fich and Shivdasani (2007); in drug discovery and development, sampling from the generic model can facilitate the discovery of new medicines which equip with new configurations while preserving the property of the existing medicines Gómez-Bombarelli et al. (2016); in social network analysis, the distributions on graphs can be used to discover new graph structures and generate the evolving graphs You et al. (2018a).

Generative models of graphs have been well studied for decades. Traditional graph generative models Erdös and Rényi (1959); Albert and Barabási (2002); Leskovec et al. (2010) are usually built upon some structural premises, e.g., heavy tails for the nodes' degree distribution, small diameters, and densification in graph evolution. More recent studies on the deep generative models, e.g., Goodfellow et al. (2014); Kingma and Welling (2013), reveal a surge of research interest in modeling graphs. For example, Liu et al. (2017) proposes a deep model for learning characteristic topological features from the given graphs via generative adversarial networks (GAN); You et al. (2018a) uses a deep autoregressive model to efficiently learn the complex joint probability of all the nodes and edges from an observed set of graphs.

*Dawei Zhou and Lecheng Zheng contributed equally to this work.

However, real-world networks typically exhibit hierarchical distribution over graph communities, while the existing graph generative models are either restricted to certain structural premises Erdös and Rényi (1959); Albert and Barabási (2002); Leskovec et al. (2010) or unable to capture the hierarchical community structures over the graphs Grover et al. (2018); Li et al. (2018); Simonovsky and Komodakis (2018). Developing graph generative models that can capture not only the low connectivity patterns at the level of individual nodes and edges, but also the higher-order connectivity patterns, i.e., the hierarchical community structures in the given graphs, will significantly improve the fidelity of graph generative models and help reveal more intriguing patterns in various domains. For instance, given an author-collaborative network, research groups of well-established and closely collaborated researchers could be identified by the existing graph clustering methods in the lower-level granularity. While, from a coarser level, we may find these research groups constitute large-scale communities which correspond to various research topics or subjects. Moreover, different from image data or text data, a graph with $n$ nodes can be represented by $n!$ equivalent adjacency matrices with node permutation, which increases the difficulty of training the generative model in the first place.

In this paper, we aim to address the following two open questions: ($\boldsymbol{Q1}$) How to capture the community structures at different levels of granularity and generate a unique graph representation that preserves such hierarchical graph structures? ($\boldsymbol{Q2}$) How to alleviate the high complexity of modeling the numerous representations of graphs and ensure the fidelity of the proposed graph generative model? To address the preceding challenges, we propose a generic generative model of graphs (*Misc-GAN*) to learn the underlying distribution of graph structures at different levels of granularity. In particular, our proposed framework consists of three key steps. First, it coarsens the input graph into the structured representations of different levels (i.e., granularity). Then, inspired by the success of deep generative models in image translation Goodfellow et al. (2014); Kingma and Welling (2013), a cycle-consistent adversarial network (CycleGAN) Zhu et al. (2017) is adopted to learn the graph structure distribution and generate a synthetic coarse graph at each granularity level. At last, the *Misc-GAN* framework defines a reconstruction process, which reconstructs the graphs at each granularity level and aggregates them into a unique representation.

The main contributions of this paper can be summarized from three aspects:

1. A novel problem setting which aims to model the complex distribution of community structures at different granularity levels in the real networks.
2. A graph generative model which is capable of modeling hierarchical topology features from single or a set of observed graphs and produce high-quality domain specific synthetic graphs.
3. Extensive experiments and case-studies on both real and synthetic data sets, showing the effectiveness of the proposed framework *Misc-GAN*.

The rest of this paper is organized as follows. We briefly review some related work in Section 2, formally define the multi-scale domain adaptive graph generation problem in Section 3 and present the formulation and implementation of our proposed *Misc-GAN* framework in Section 4. The empirical studies are conducted in Section 5. Finally, we conclude this paper in Section 6.

## 2 RELATED WORK

In this section, we briefly review the related studies on the generative adversarial network, multi-scale analysis of graph and cycle consistency.

## 2.1 Generative Adversarial Network

In Goodfellow et al. (2014), the authors propose the generative adversarial networks (GANs) to create a generative model and a discriminative model and compete them with each other in the adversarial setting. The authors denote $P_z(z)$ to be the prior of the input noise variables $z$ and $G(z;\theta_g)$ to represent a mapping to data space, where G is a differentiable function represented by a multi-layer perceptron with parameters $\theta_g$. $G(z)$ maps the noise variables to data space and it aims to generate samples as genuine as possible. The authors also define $D(x;\theta_d)$ to be another multi-layer perceptron or discriminator distinguishing whether the given samples are drawn from the real-world data set or from the fake data set. $D(x)$ is the probability of $x$ coming from the real-world data set rather than the generated data set. In this min-max game, the discriminator $D$ aims to maximize the probability of assigning the correct label to both the real samples and the faked samples generated by the generator $G$, while the generator $G$ aims to minimize the probability that the discriminator $D$ successfully distinguishes the faked samples from the real samples. The objective of this min-max game is written as:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim P_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

In this paper, generative adversarial network is the basis to transfer graphs from one domain to another domain, and meanwhile, the local valuable structures of graphs are preserved.

## 2.2 Multi-Scale Analysis of Graphs

Multi-scale analysis of graphs has been studied for years in machine learning with wide applications in numerous areas, such as simplification and compression of graphs Safro and Temkin (2011); Cour et al. (2005), dynamics of graphs at different resolutions Lee and Maggioni (2011); Gao et al. (2016), graph visualization Stolte et al. (2003), recommendation systems Gou et al. (2011) and so on. The common assumption of multi-scale analysis is that the given data in a high dimensional space has a much lower dimensional intrinsic geometry. Take the document text as an example, the dependencies among words constrain the distribution of word frequency in a lower dimensional space. Diffusion wavelets Coifman and Maggioni (2006) is one common method used in multi-scale analysis which allows us to construct functions on the graph for statistical learning tasks by producing coarser and coarser graphs at different resolution levels. In this paper, we adopt the concept of multi-scale analysis to capture the local structure of graphs at different resolution levels and then reconstruct the graph while preserving these important local structures.

## 2.3 Cycle Consistency

The concept of cycle consistency has been applied to various computer vision problems, including image matching Huang and Guibas (2013); Zhou et al. (2015), co-segmentation Wang et al. (2013, 2014), style transfer Zhu et al. (2017); Chang et al. (2018), and structure from motion Zach et al. (2010); Wilson and Snavely (2013). The idea of cycle consistency constrain is utilized as a regularizer in these algorithms, such as cycle consistency loss used in Zhou et al. (2016); Godard et al. (2017) to push the mappings to be as consistent with each other as possible in the supervised convolution neural network training. Zhu et al. (2017) proposes the Cycle-Consistent generative adversarial network to learn two mappings or generators $G : X \to Y$ and $F : Y \to X$ between two domains $X$ and $Y$. The authors introduce two adversarial discriminators $D_X$ and $D_Y$, where $D_X$ aims to distinguish the images $x$ drawn from the real data set $X$ from the fake images generated by $F(Y)$; similarly, $D_Y$ aims to distinguish the images $y$ drawn from data

set $Y$ from the fake images generated by $G(X)$. In this paper, we apply this concept to find the graph transfer mappings between domain $X$ and domain $Y$, such that the transferred graph from domain $Y$ to domain $X$ is sufficiently similar to the graph in domain $X$.

## 3 PROBLEM DEFINITION

In this section, we introduce the notation and problem definition of this paper. The main symbols and notations are summarized in Table 1. We use ordinary lowercase letters to denote scalars, boldface lowercase letters to denote vectors, and boldface uppercase letters to denote matrices and tensors. Moreover, the elements (e.g., entries, fibers and slices) in a matrix or a tensor are represented in the same way as the Matlab, e.g., $M(i,j)$ is the element at the $i^{\text{th}}$ row and $j^{\text{th}}$ column of the matrix $\boldsymbol{M}$, and $\boldsymbol{M}(i,:)$ is the $i^{\text{th}}$ row of M, etc.

**Table 1.** Symbols and Notations.

| Symbol | Definition and Description |
|---|---|
| $G_s, G_t$ | the source domain graph and the target domain graph |
| $\tilde{G}_t$ | the generated graph of the target domain |
| $\boldsymbol{A}_s, \boldsymbol{A}_t, \tilde{\boldsymbol{A}}_t$ | the adjacency matrices of $G_s$, $G_t$ and $\tilde{G}_t$ |
| $V_s, V_t$ | the sets of nodes in $G_s$ and $G_t$ |
| $E_s, E_t$ | the sets of edges in $G_s$ and $G_t$ |
| $G_s^{(l)}, G_t^{(l)}$ | the induced $l$-th granularity coarse graphs of $G_s$ and $G_t$ |
| $n_s, n_t$ | number of nodes in $G_s$ and $G_t$ |
| $m_s, m_t$ | number of edges in $G_s$ and $G_t$ |
| $L$ | number of granularity levels |
| $\mathcal{F}^{(l)}, \mathcal{B}^{(l)}$ | the generators in the forward and backward GAN at the $l$-th layer |
| $D_{\mathcal{F}}^{(l)}, D_{\mathcal{B}}^{(l)}$ | the discriminators in the forward and backward GAN at the $l$-th layer |

The goal of this paper is to generate a synthetic target domain graph $\tilde{G}_t$, by learning mapping functions between the source domain graph $G_s$ and the target domain graph $G_t$. Without loss of generality, in this paper, we assume that there exists a universal structure distribution $p_{data}$, which defines the structural role of each entity, i.e., node, edge, and subgraph, of the observed graphs. Many existing graph generative models Bojchevski et al. (2018); You et al. (2018b) are designed to learn the structure distribution of $G$ at a single scale, and therefore they might overlook some intriguing patterns in the underlying networks, e.g., the multi-level cluster-within-cluster structures Ravasz and Barabási (2003). Fig. 1 presents an illustrative example of the hierarchical structures in collaboration networks. In particular, the graph exhibits four-level hierarchies including (L1) all the entities in the collaboration network, (L2) early-stage researchers, (L3) mid-career researchers and (L4) senior researchers. It is unclear how to characterize such hierarchical structures and generate domain-specific synthetic graphs. Moreover, the generative model needs to be scalable when modeling large-scale networks that have exponentially many representations. With the above notations and objects, we formally define our problem as follows:

PROBLEM 1. ***Multi-Level Structure-Preserving Graph Generation***

***Input:*** *(i) a target domain graph $G_t = (V_t, E_t)$, (ii) a source domain graph $G_s = (V_s, E_s)$, (iii) the number*

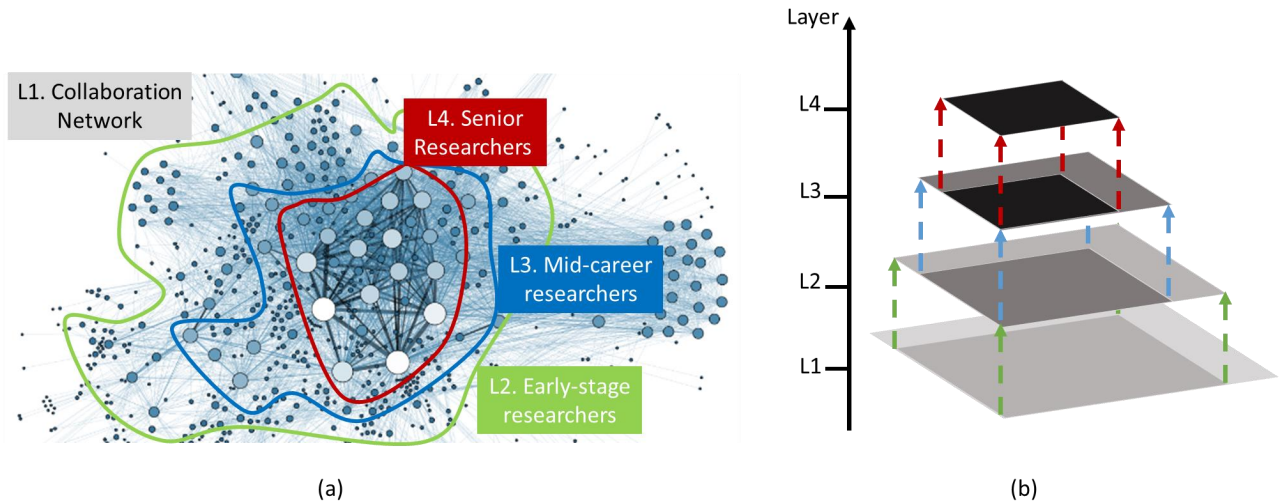



**Figure 1.** An illustration example. (a) presents a visualization of the collaboration network Grandjean (2016). (b) shows the hierarchical structure of the research communities, from early-stage researchers to mid-career researchers and senior researchers.

*of granularity levels $L$.*
***Output:*** *(i) a mapping function $\mathcal{F}$ that can translate any source-domain graphs to the corresponding target domain graphs while preserving the hierarchical structure distribution over the observed target graph $G_t$, (ii) a generated synthetic target domain graph $\tilde{G}_t$.*

## 4 PROPOSED FRAMEWORK

In this section, we present our multi-scale graph generative model *Misc-GAN*, which simultaneously characterize and model the structural distribution of the observed graphs at multiple scales. In particular, we first formulate our framework into a generic optimization problem, and then discuss the details on three modules, i.e., multi-scale graph representation module, graph generation module, and graph reconstruction module, in our proposed framework 2.

### 4.1 A Generic Joint Learning Framework

To address the proposed problem of multi-level structure-preserving graph generation, our joint learning framework should primarily focus on the following aspects. First (*problem setting*), the existing methods are mainly restricted to a single granularity level of graph structures, which might increase the possibility of overlooking the hierarchical community structures in the observed graphs. Thus, the graph generation model should be able to capture the community structures at multiple levels of granularity and generate a unique graph representation. Second (*graph generation performance*), it is unclear how to alleviate the high complexity and ensure the fidelity of the graph generation. This is crucial especially if the observed graphs are noisy and large-scale. With these objectives in mind, we propose a generic graph generation framework as an optimization problem with the following objective function:

$$
\begin{aligned}
\mathcal{L} =& \mathcal{L}_{ms} + \mathcal{L}_{\mathcal{F}} + \mathcal{L}_{\mathcal{B}} + \mathcal{L}_{cyc} \\
=& \underbrace{KL(\sum_{l=1}^{L} w^{(l)} \mathcal{F}^{(l)}(G_s^{(l)}) + b, G_t)}_{\mathcal{L}_{ms}\text{: multi-scale reconstruction loss}} \\
&+ \underbrace{\alpha \sum_{l=1}^{L} \mathbb{E}_{G_t^{(l)} \sim P_{data}(G_t^{(l)})}[\log D_{\mathcal{F}}^{(l)}(G_t^{(l)})] + \mathbb{E}_{G_s^{(l)} \sim P_{data}(G_s^{(l)})}[\log(1 - D_{\mathcal{F}}^{(l)(l)}(\mathcal{F}(G_s^{(l)})))]}_{\mathcal{L}_{\mathcal{F}}\text{: forward adversarial loss}} \\
&+ \underbrace{\beta \sum_{l=1}^{L} \mathbb{E}_{G_s^{(l)} \sim P_{data}(G_s^{(l)})}[\log D_{\mathcal{B}}^{(l)}(G_s^{(l)})] + \mathbb{E}_{G_t^{(l)} \sim P_{data}(G_t^{(l)})}[\log(1 - D_{\mathcal{B}}^{(l)}(\mathcal{B}^{(l)}(G_t^{(l)})))]}_{\mathcal{L}_{\mathcal{B}}\text{: backward adversarial loss}} \\
&+ \underbrace{\gamma \sum_{l=1}^{L} \mathbb{E}_{G_s^{(l)} \sim P_{data}(G_s^{(l)})}[\|\mathcal{B}^{(l)}(\mathcal{F}^{(l)}(G_s^{(l)})) - G_s^{(l)}\|_1] + \mathbb{E}_{G_t^{(l)} \sim P_{data}(G_t^{(l)})}[\|\mathcal{F}^{(l)}(\mathcal{B}^{(l)}(G_t^{(l)})) - G_t^{(l)})\|_1]}_{\mathcal{L}_{cyc}\text{: cycle consistency loss}}
\end{aligned}
$$

where the objective consists of four terms. The first term $L_{ms}$ is the multi-scale reconstruction loss, which is designed to minimize the Kullback-Leibler ($KL$) divergence Moreno et al. (2004) between the target graph $G_t$ and the generated graph $\tilde{G}_t$, i.e., $\tilde{G}_t = \sum_{l=1}^{L} w^{(l)} \mathcal{F}(G_s^{(l)}) + b$. We generalize the conventional $KL$ divergence to our problem setting to compare two graphs as follows

$$
KL(\tilde{G}_t, G_t) = \sum_{i=1}^{n} \sum_{j=1}^{n} (A_t(i,j) + \epsilon) \log \frac{A_t(i,j) + \epsilon}{\tilde{A}_t(i,j) + \epsilon} \tag{2}
$$

where $\boldsymbol{A_t}$ and $\tilde{\boldsymbol{A}}_{\boldsymbol{t}}$ are the adjacency metrics of $\tilde{G}_t$ and $G_t$, $\epsilon$ is a constant with a small value to avoid $\log(0)$ or division by 0. The second term $\mathcal{L}_{\mathcal{F}}$ learns a forward mapping function $\mathcal{F}$ from the source graph $G_s$ to $G_t$. The discriminator $D_{\mathcal{F}}^{(l)}$ aims to figure out whether the given graph is a real graph from the target domain or a fake graph generated by the generator $\mathcal{F}$ which is transferred from the source domain graph. Similar to the second term, the third term $\mathcal{L}_{\mathcal{B}}$ defines a backward adversarial loss, which aims to learn the mapping function from the target domain to the source domain. The fourth term $\mathcal{L}_{cyc}$ is the cycle consistency loss, which is introduced to further reduce the space of possible mapping function. We argue that learning such bi-directional mapping can largely prevent the learned mapping functions from contradicting each other. At last, we also introduce three positive constants, i.e., $\alpha, \beta, \gamma$, to balance the impact of these four terms in the overall objective function. Follow the min-max scheme of GAN, we aim to solve:

$$
\mathcal{F}^{*(l)}, w^{*(l)}, b^* = \arg \min_{\mathcal{F}^{(l)}, \mathcal{B}^{(l)}, w^{(l)}, b} \max_{D_{\mathcal{F}}^{(l)}, D_{\mathcal{B}}^{(l)}} \mathcal{L}, l = 1, \ldots, L \tag{3}
$$

## 4.2 Network Architecture

Here, we present our *Misc-GAN* framework (Fig. 2). Overall, our framework can be separated into three stages (i.e., modules). In the first stage, our framework takes the input graphs $G_t$ and explores the hierarchical structures by constructing the coarse graphs in $L$ levels of granularity (w.r.t. $L$ layers in Fig. 2). In the second stage, our framework trains an independent graph generative model and produces the multi-scale coarse graph in each layer. In the third stage, our framework autonomously combines the outputs from the previous stage to construct the synthetic graph $\tilde{G}_t$ that preserves the hierarchical topology features of the given graphs $G_t$.

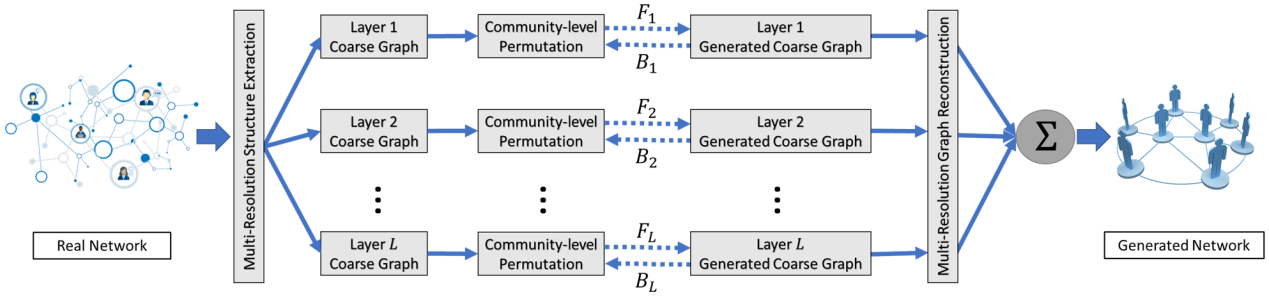



**Figure 2.** The proposed *Misc-GAN* framework.

**Multi-Scale Graph Representation Module.** In this module, we explore the hierarchical cluster-within-cluster structures in order to better characterize the given graph $G_t$, by using the multi-scale approaches, e.g., hierarchical clustering Johnson (1967), algebraic multigrid (AMG) Ruge and Stüben (1987). In particular, given a symmetric matrix $\boldsymbol{A}_t$, the multi-scale approaches recursively construct a multi-scale hierarchy of increasingly coarser graphs as follows

$$\boldsymbol{P}^{(l-1)\prime} \dots \boldsymbol{P}^{(1)\prime} \boldsymbol{A}_t \boldsymbol{P}^{(1)} \dots \boldsymbol{P}^{(l-1)} = \boldsymbol{A}_t^{(l)} \tag{4}$$

where $l = 1, \dots, L$, $\boldsymbol{P}^{(1)}, \dots, \boldsymbol{P}^{(l-1)}$ are the coarsening operators , and $A_l$ is the coarse graph at the $l$-th layer. Based on Eq. 4, we construct a set of coarse graphs with multiple scales from the target domain graph $G_t$. These coarse graphs will be fed into the following graph generative module in order to learn the hierarchical structures of $G_t$.

**Graph Generation Module.** It is challenging to learn the underlying structure distribution $p_{data}$ of the target domain graph $G_t$, as the graph with $n$ nodes can be represented by $n!$ equivalent adjacency matrices with node permutations You et al. (2018a). Some recent works have been proposed to tackle this issue. For example, Simonovsky and Komodakis (2018) proposes an approximate graph matching scheme that requires $O(n^4)$ operations in the worst case; You et al. (2018a) develops a tree-structure node ordering scheme, which is based on breadth-first-search (BFS) to reduce the computational complexity. However, these methods may either suffer form the intractable time complexity, or not well preserve the hierarchical structures of the given networks.

Here, we propose a multi-scale graph generation scheme, which models the complex distribution of graph structures over a pyramid of coarse graphs rather than the original graphs. The intuitions are in the following two aspects: (1) directly training from the coarse graphs facilitates the learning process of the generative model, as the coarse graphs serve as the abstractions of the original graphs; (2) this scheme provides the flexibility for the users to decide the granularity-level of the coarse graphs to be learned, which could be attractive when we need to model the large-scale networks. To be more specific, the graph generation

module at each layer (shown in Fig. 2) can be separated into three steps: First, we partition the graph into multiple non-overlapping subgraphs using state-of-the-art graph clustering methods Ester et al. (1996); Schaeffer (2007). Then, based on the detected communities, we generate a set of block diagonal matrices by shuffling community blocks over the diagonals, which are used to characterize the community-level graph structures. At last, the generated block diagonal matrices are fed into an independent graph generative model to generate the synthetic coarse graphs at each layer.

**Graph Reconstruction Module.** In this stage, we first adopt the multi-scale approaches to reconstruct the graph from coarse to fine as follows

$$\boldsymbol{R}^{(1)\prime} \dots \boldsymbol{R}^{(l-1)\prime} \boldsymbol{A}_t^{(l)} \boldsymbol{R}^{(l-1)} \dots \boldsymbol{R}_1 = \tilde{\boldsymbol{A}}_t^{(l)} \tag{5}$$

where $l = 1, \dots, L$, $\boldsymbol{R}^{(1)}, \dots, \boldsymbol{R}^{(l-1)}$ are the reconstruction operations, and $\tilde{\boldsymbol{A}}_t^{(l)}$ is the reconstructed adjacency matrix from the $l$-th layer. After that, all the reconstructed graphs are in the same scale as the target graph $G_t$, which could be aggregated into a unique one by a linear function, $\tilde{\boldsymbol{A}}_{\boldsymbol{t}} = \sum_{l=1}^{L} w^{(l)} \tilde{\boldsymbol{A}}_t^{(l)} + b$, where $w^{(1)}, \dots, w^{(L)}$ are the non-negative weights, and $b$ is a bias.

### 4.3 Training Details

We apply the technique of cycleGAN to transfer graph from one domain to another domain. Different from the density property of images, the adjacency matrix for a graph is much sparser. In our algorithm, two convolution layers are used to capture the hierarchical structure information of the graph. Because the adjacency matrix of a graph is sparser than the dense matrix of an image, we set the size of stride to be 4, the size of kernels to be $4 \times 4$ matrices, and the number of kernels to be 32 for each convolution layer. Then, $k$ iterations of ResNet He et al. (2016) are applied to accelerate the convergence. Finally, two deconvolution layers are used to reconstruct the adjacency matrix with similar settings used in convolution layers.

Second, following the strategy mentioned in Shrivastava et al. (2017); Zhu et al. (2017), we update two discriminators with the history of the generated graph $\tilde{\boldsymbol{A}}_t^{(l)}$ in the $l$-th layer to reduce the vibration of the model. For all the experiments, we set the training iterations to be 250. Adam solver Kingma and Ba (2014) with a batch size of 1 is used to minimize the loss function, and all networks are trained with a learning rate of 0.0002 in the tensorflow deep learning framework.

## 5 EXPERIMENT

In this section, we demonstrate the performance of our proposed *Misc-GAN* framework on real networks. Moreover, we present a case study to illustrate the effectiveness of *Misc-GAN* in learning the topological features at different levels of granularity.

### 5.1 Experiment Setup

**Data sets:** We evaluate our proposed algorithm on seven real-world networks from the Stanford Network Analysis Project (SNAP) Leskovec and Krevl (2015). The statistics of data sets are summarized in Table 2. In particular, Email is a communication network, where an edge exists if one person sends at least one email to another person; Facebook is a social network, where each edge represents a social connection between the users in Facebook; Wiki is a voting network, which is used by Wikipedia to elect administrators among the huge contributors; P2P is a file-sharing network, where each node represents a host and each edge

| Network | Type | Nodes | Edges |
|---|---|---|---|
| Email | Directed | 1,005 | 25,571 |
| Facebook | Undirected | 4,039 | 88,234 |
| Wiki | Directed | 8,292 | 14,547,910 |
| P2P | Directed | 10,876 | 39,994 |
| Gnu | Directed | 6,301 | 20,777 |
| Bitcoin | Directed | 5,881 | 35,592 |
| CA | Undirected | 5,242 | 14,496 |

**Table 2.** Statistics of the network data sets.

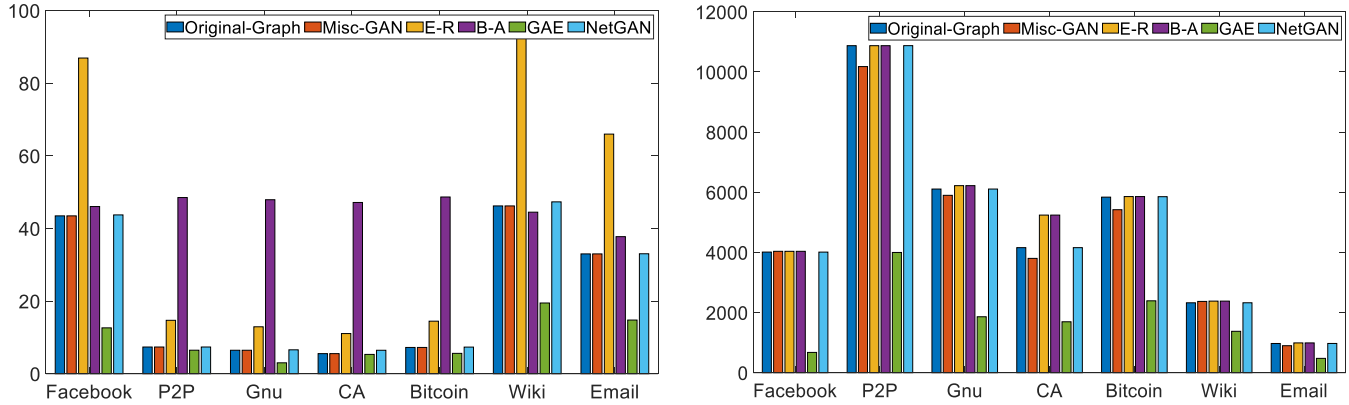
represents a connection between hosts; GNU is another Gnutella peer-to-peer file sharing network, which is similar to P2P network; Bitcoin is a who-trusts-whom network that covers the bitcoin trading information on the Bitcoin OTC platform, where each node represents a user and each edge represents the trustfulness between two users; CA is a collaboration network from arXiv, where each node represents an author and each edge represents the collaborations between authors. For different weights in a graph, i.e., Bitcoin graph, we convert the values of edges to binary values in order to feed them to our model.

**Comparison Methods:** We compare *Misc-GAN* with two random graph models, i.e., Erdös-Rényi (E-R) model Erdös and Rényi (1959) and Barabási-Albert (B-A) model Albert and Barabási (2002), and one recent deep graph generative model, i.e., GAE Kipf and Welling (2016). All the graph statistics are outlined in Table 2. In our setting, the graphs in Table 2 are target domain graphs, and the source domain graphs are generated under a random normal distribution with the same numbers of nodes and edges as the target domain graphs.

**Repeatability:** All the data sets are publicly available. We will release the code of our algorithms through the authors' website after the paper is published. The experiments are performed on a Windows machine with four 3.5GHz Intel Cores and 256GB RAM.

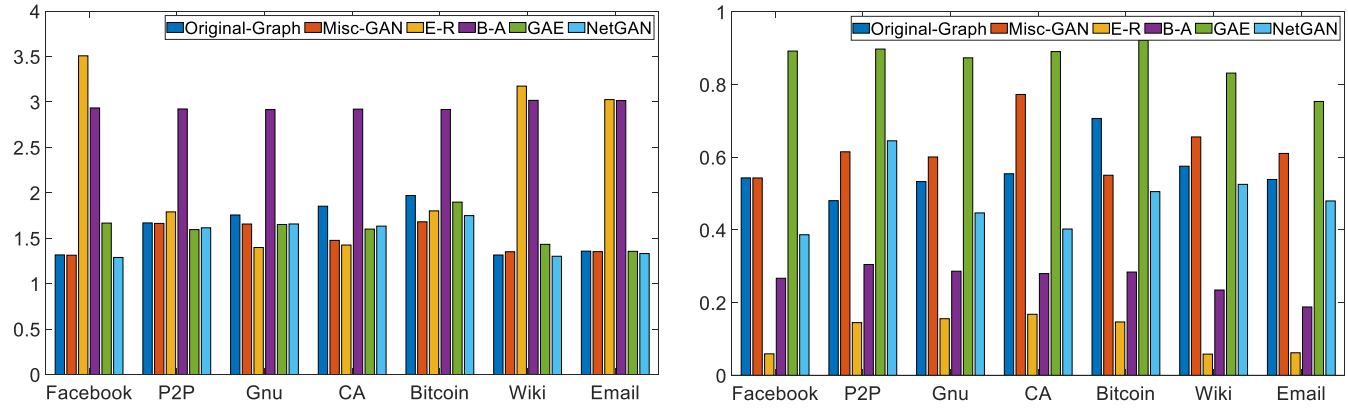
## 5.2 Quantitative Evaluation

The comparison results in terms of effectiveness across a diverse set of real networks are shown in Fig. 3. In particular, we present the results regarding the following metrics: (1) AD: the average degree of all nodes in a graph; (2) LCC: the size of the largest connected component of the graph; (3) EPL: the exponent of the power law distribution of the graph; (4) GC: the Gini coefficient of the degree distribution of the graph; (5) KL: the symmetric Kullback-Leibler (KL) divergence Moreno et al. (2004) between the local clustering coefficient distributions of the original graphs and the generated graphs; (6) Graph Kernel: the similarity between the original graph and the generated one by using the random-walk based graph kernel Kang et al. (2012). From these figures, the x-axis of each figure represents a data set, and the y-axis is the value of metrics. From Fig. 3 (a) to Fig. 3 (d), we mainly compare various graph statistics between the original graph and the generated ones using baseline methods. If the value of the metric of the generated graph is close to that of the original graph, it means the generated graph is much more similar to the original graph. We observe that the AD of our proposed algorithm is almost identical to the AD of the original graph for all data sets; for the other three metrics, our proposed algorithm also outperforms the others in most cases. In Fig. 3 (e) and Fig. 3 (f), we present the divergence and similarity score between the original graphs and the generated graphs. Note that, for presentation purposes, all the results in Fig. 3 (e) and Fig. 3 (f) are presented using a negative log function, i.e., $f(x) = -\log(x)$. In general, we observe that (1) our proposed *Misc-GAN* outperforms the comparison methods across most of the datasets and evaluation metrics in most cases. For example, in the Email data, *Misc-GAN* is $66\%$ smaller on the clustering coefficient distribution



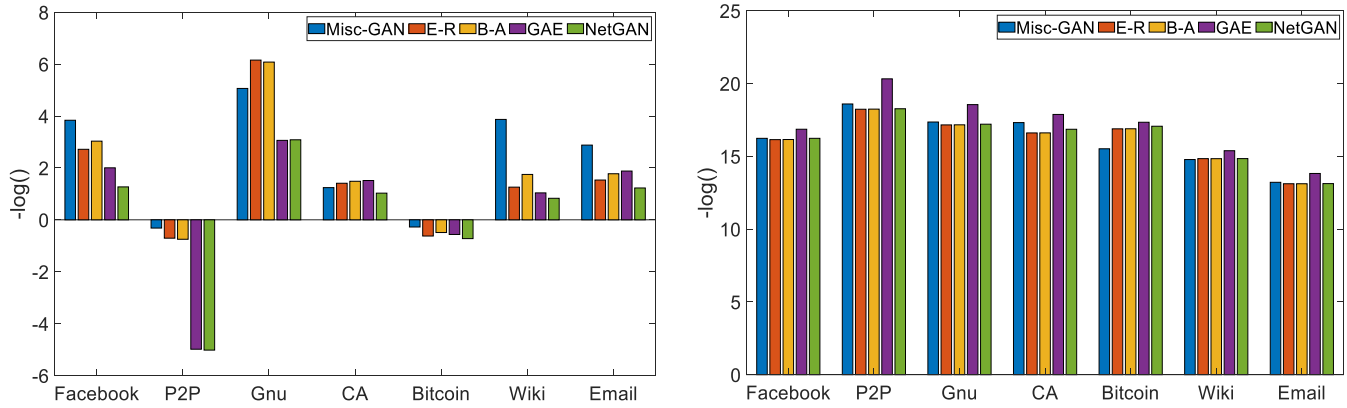

(a) AD (More similar to original graph is better)

(b) LCC (More similar to original graph is better)




(c) EPL (More similar to original graph is better)

(d) GC (More similar to original graph is better)




(a) KL (Higher is better)

(b) Graph Kernel (Lower is better)

**Figure 3.** Effectiveness analysis.

evaluation; (2) our proposed *Misc-GAN* framework better preserves the local topological features (e.g., the largest connected component and local clustering coefficient) and the global features (e.g., mean degree, the power law coefficients of the degree distribution of graphs) than other deep generative models (e.g., GAE). It is because our method explores the network structures at multiple resolutions and automatically learns the weights regarding the importance of topological features at different levels, while the existing deep generative models may fail to model such fine-grained topological features.

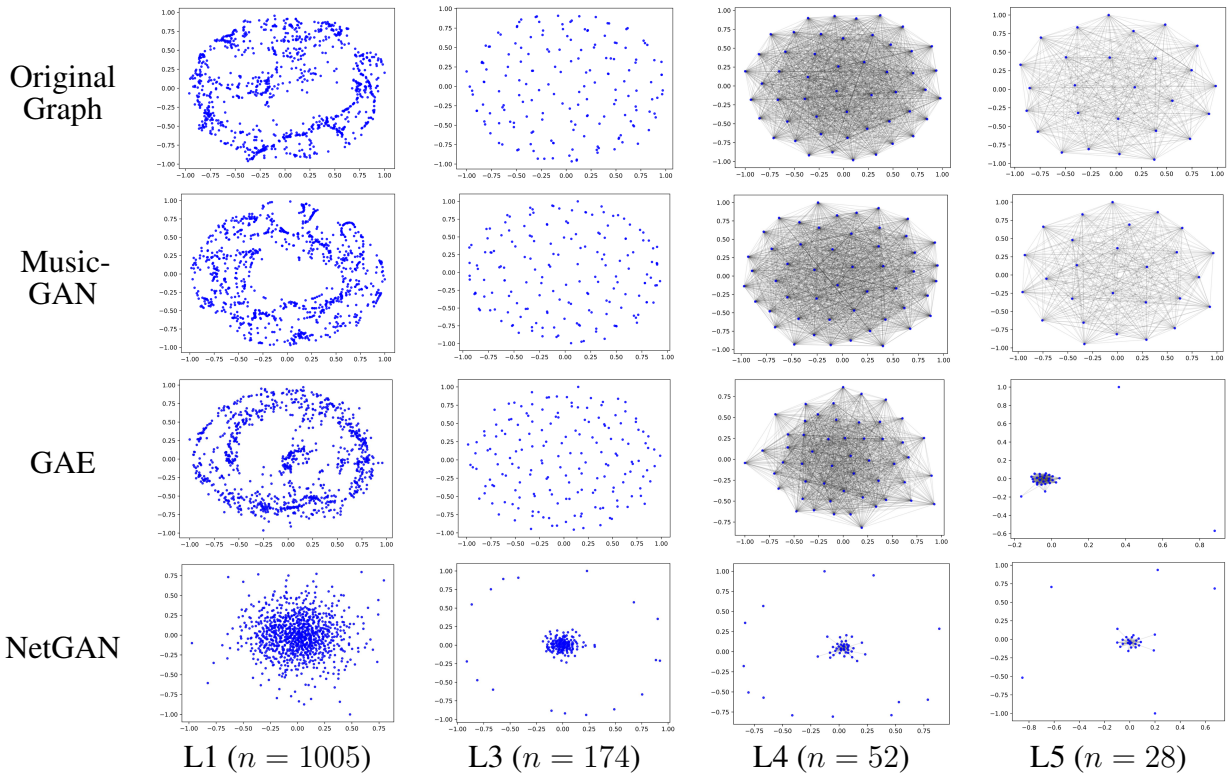



**Figure 4.** Graph reconstruction at multiple scales.

### 5.3 A Case Study with respect to the Impact of Multi-Scale Analysis

A simple but intuitive way to evaluate the generated graphs is to visualize the network layout in a two-dimensional space. In Fig. 4, we compare the multi-scale network representations of the original graph (i.e., Email) and the generated graphs. In particular, we select the deep generative model GAE and NetGAN as our baseline methods and construct coarse graphs at four different scales based on Eq. 4. In general, we find that (1) our framework well preserves the graph structures at multiple levels of granularity; (2) NetGAN only preserves the lower-level connectivity patterns (e.g., clusters within a loop pattern) in Layer 1, but fails in capturing the higher-level connectivity patterns (e.g., the cluster of super-nodes) in Layer 3, Layer 4 and Layer 5. The reason for the preceding phenomenon is that NetGAN is trained at a single level (i.e., a single granularity of nodes), which results in the coarse reconstruction of high-level network structures. GAE also has the similar problem due to the failure to capture the higher-level connectivity patterns(i.e., in Layer 5).

## 6 CONCLUSION

We propose a multi-scale generative model named *Misc-GAN* for graph-structured data, which explores the network structures at multiple resolutions and automatically generates a unique graph representation that preserves such fine-grained topological features. The empirical studies show that *Misc-GAN* achieves significantly better performance compared to the state-of-the-art models on real networks. However, various challenges remain in this problem, such as how to make the deep generative model scale to massive graphs, and how to generate the domain-specific graph with complex connectivity patterns (e.g., modeling the online transaction networks with money laundering patterns)?

## ACKNOWLEDGMENT

This work is supported by the United States Air Force and DARPA under contract number FA8750-17-C-0153 , National Science Foundation under Grant No. IIS-1552654, and Grant No. CNS-1629888, the U.S. Department of Homeland Security under Grant Award Number 2017-ST-061-QA0001, and an IBM Faculty Award. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

## REFERENCES

Albert, R. and Barabási, A. (2002). Statistical mechanics of complex networks. *Reviews of modern physics*

Bojchevski, A., Shchur, O., Zügner, D., and Günnemann, S. (2018). Netgan: Generating graphs via random walks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 609–618

Chang, H., Lu, J., Yu, F., and Finkelstein, A. (2018). Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

Coifman, R. R. and Maggioni, M. (2006). Diffusion wavelets. *Applied and Computational Harmonic Analysis* 21, 53–94

Cour, T., Benezit, F., and Shi, J. (2005). Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (IEEE), vol. 2, 1124–1131

Erdös, P. and Rényi, A. (1959). On random graphs, i. *Publicationes Mathematicae (Debrecen)*

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., and etc. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*

Fich, E. M. and Shivdasani, A. (2007). Financial fraud, director reputation, and shareholder wealth. *Journal of Financial Economics*

Gao, Z.-K., Cai, Q., Yang, Y.-X., Dang, W.-D., and Zhang, S.-S. (2016). Multiscale limited penetrable horizontal visibility graph for analyzing nonlinear time series. *Scientific reports* 6, 35622

Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 6602–6611

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., et al. (2016). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *NIPS*

Gou, L., You, F., Guo, J., Wu, L., and Zhang, X. L. (2011). Sfviz: interest-based friends exploration and recommendation in social networks. In *Proceedings of the 2011 Visual Information Communication-International Symposium* (ACM), 15

[Dataset] Grandjean, M. (2016). How to display complex network data with information visualization. `https://www.interaction-design.org/literature/article/how-to-display-complex-network-data-with-information-visualization`. CC BY-SA 3.0

Grover, A., Zweig, A., and Ermon, S. (2018). Graphite: Iterative generative modeling of graphs. *arXiv preprint arXiv:1803.10459*

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 770–778

Huang, Q. and Guibas, L. J. (2013). Consistent shape maps via semidefinite programming. *Comput. Graph. Forum* 32, 177–186

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*

Kang, U., Tong, H., and Sun, J. (2012). Fast random walk graph kernel. In *SDM*

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR* abs/1412.6980

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*

Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*

Lee, J. D. and Maggioni, M. (2011). Multiscale analysis of time series of graphs. In *International Conference on Sampling Theory and Applications (SampTA)*

Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Ghahramani, Z. (2010). Kronecker graphs: An approach to modeling networks. *JMLR*

Leskovec, J. and Krevl, A. (2015). {SNAP Datasets}:{Stanford} large network dataset collection

Li, Y., Vinyals, O., Dyer, C., Pascanu, R., and Battaglia, P. (2018). Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*

Liu, W., Cooper, H., Oh, M. H., Yeung, S., Chen, P., Suzumura, T., et al. (2017). Learning graph topological features via gan. *arXiv preprint arXiv:1709.03545*

Moreno, P. J., Ho, P. P., and Vasconcelos, N. (2004). A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *NIPS*

Ravasz, E. and Barabási, A. (2003). Hierarchical organization in complex networks. *Physical Review E*

Ruge, J. W. and Stüben, K. (1987). Algebraic multigrid. In *Multigrid methods* (SIAM)

Safro, I. and Temkin, B. (2011). Multiscale approach for the network compression-friendly ordering. *Journal of Discrete Algorithms* 9, 190–202

Schaeffer, S. (2007). Graph clustering. *Computer science review*

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 2242–2251

Simonovsky, M. and Komodakis, N. (2018). Graphvae: Towards generation of small graphs using variational autoencoders. *arXiv preprint arXiv:1802.03480*

Stolte, C., Tang, D., and Hanrahan, P. (2003). Multiscale visualization using data cubes. *IEEE Transactions on Visualization and Computer Graphics* 9, 176–187

Wang, F., Huang, Q., and Guibas, L. J. (2013). Image co-segmentation via consistent functional maps. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 849–856

Wang, F., Huang, Q., Ovsjanikov, M., and Guibas, L. J. (2014). Unsupervised multi-class joint image segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. 3142–3149

Wilson, K. and Snavely, N. (2013). Network principles for sfm: Disambiguating repeated structures with local context. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*. 513–520

You, J., Ying, R., Ren, X., Hamilton, W. L., and Leskovec, J. (2018a). Graphrnn: A deep generative model for graphs. *arXiv preprint arXiv:1802.08773*

You, J., Ying, R., Ren, X., Hamilton, W. L., and Leskovec, J. (2018b). Graphrnn: Generating realistic graphs with deep auto-regressive models. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 5694–5703

Zach, C., Klopschitz, M., and Pollefeys, M. (2010). Disambiguating visual relations using loop constraints. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*. 1426–1433

Zhou, T., Krähenbühl, P., Aubry, M., Huang, Q., and Efros, A. A. (2016). Learning dense correspondence via 3d-guided cycle consistency. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 117–126

Zhou, T., Lee, Y. J., Yu, S. X., and Efros, A. A. (2015). Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 1191–1200

Zhu, J., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*