

Learning Graph Representation of Agent Diffusers

Youcef Djenouri
University of South-Eastern Norway
Vestfold, Norway
youcef.djenouri@usn.no
Norwegian Research Centre
Oslo, Norway
youcef.djenouri@usn.no

Nassim Belmecheri
Simula Laboratory Research
Oslo, Norway
nassim@simula.no

Tomasz Michalak
University of Warsaw
Warsaw, Poland
tpm@mimuw.edu.pl

Jan Dubiński
IDEAS NCBR
Warsaw, Poland
jan.dubinski@ideas-ncbr.pl

Ahmed Nabil Belbachir
Norwegian Research Centre
Grimstad, Norway
nabe@norcersearch.no

Anis Yazidi
University of Oslo
Oslo, Norway
anisy@ifi.uio.no

ABSTRACT

Diffusion-based generative models have significantly advanced text-to-image synthesis, demonstrating impressive text comprehension and zero-shot generalization. These models refine images from random noise based on textual prompts, with initial reliance on text input shifting towards enhanced visual fidelity over time. This transition suggests that static model parameters might not optimally address the distinct phases of generation. We introduce LGR-AD (Learning Graph Representation of Agent Diffusers), a novel multi-agent system designed to improve adaptability in dynamic computer vision tasks. LGR-AD models the generation process as a distributed system of interacting agents, each representing an expert sub-model. These agents dynamically adapt to varying conditions and collaborate through a graph neural network that encodes their relationships and performance metrics. Our approach employs a coordination mechanism based on top-k maximum spanning trees, optimizing the generation process. Each agent's decision-making is guided by a meta-model that minimizes a novel loss function, balancing accuracy and diversity. Theoretical analysis and extensive empirical evaluations show that LGR-AD outperforms traditional diffusion models across various benchmarks, highlighting its potential for scalable and flexible solutions in complex image generation tasks. Code can be found at: https://github.com/YousIA/LGR_AD.

KEYWORDS

Collaborative Frameworks; Graph Representations of Agents; Text to Image Generation.

ACM Reference Format:

Youcef Djenouri, Nassim Belmecheri, Tomasz Michalak, Jan Dubiński, Ahmed Nabil Belbachir, and Anis Yazidi. 2025. Learning Graph Representation of Agent Diffusers. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*, Detroit, Michigan, USA, May 19 – 23, 2025, IFAAMAS, 10 pages.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Diffusion models, or diffusers, have revolutionized the computer vision domain and are the cornerstone of modern text-to-image systems. These models demonstrate impressive capabilities in converting complex textual prompts into photorealistic images, including those involving novel and previously unseen concepts [11, 13, 17]. In addition, diffusers have paved the way for a range of interactive applications, significantly accelerating the democratization of content creation. However, the performance of diffusion models is not consistent across all datasets; certain architectures excel under specific conditions while others underperform. To address these performance discrepancies, researchers have increasingly explored ensemble methods and expert diffusers, which aggregate the outputs of multiple specialized models to improve results [2, 20, 40, 48]. Expert diffusers, in particular, have shown superior capabilities by generating high-quality images from both simple and complex text prompts [2]. Despite these advantages, the adoption of expert diffusers faces two major challenges within the framework of multi-agent systems. The first challenge stems from their resource-intensive nature. Since expert diffusers require loading and executing all constituent models in the expert pool during inference, they impose significant computational demands in terms of time and memory, leading to latency issues. This is especially problematic in multi-agent environments where efficiency and scalability are crucial for real-time applications. The second challenge lies in the collaborative dynamics among models in the expert pool. The inherent interactions between models, analogous to multi-agent coordination, can lead to suboptimal results when certain models exert a counterproductive influence on the collective process. This interference can impede the overall performance gains that expert diffusers are intended to provide, highlighting the need for improved agent communication and coordination strategies to prevent such disruptions. Addressing these challenges requires further research into optimizing the coordination mechanisms and resource management strategies within multi-agent systems to ensure that the full potential of expert diffusers can be realized in diverse real-world applications.

Motivations. We hypothesize that analyzing the diverse interactions among models within the agent diffusers pool can yield valuable insights to enhance the overall text-to-image generation process.

Our hypothesis suggests that by carefully analyzing and modeling the interactions between agents (i.e., the models in the agent diffusers pool), we can identify the optimal subset of models that positively contribute to the inference process. This approach aligns with the core principles of multi-agent systems, where not all agents contribute equally to a given task, and by selecting the most effective agents, we can significantly improve the system’s overall performance. In this research, we explore the dependencies and synergies among the models using a graph-based multi-agent representation. A graph structure, consisting of vertices and edges, allows us to model the complex relationships between individual models and analyze their interactions. Graph-based representations are powerful tools for capturing interdependencies among entities, enabling us to derive patterns that enhance collective decision-making [28]. In the context of agent diffusers, the goal is to obtain robust representations that offer meaningful insights into how various models collaborate within the system.

Contributions. This study provides the first comprehensive investigation of graph-based representation in the context of agent diffusers, specifically designed to address the complex coordination challenges posed by modern diffusion models in multi-agent systems. We introduce the LGR-AD framework (Learning Graph Representation of Agent Diffusers), which establishes a novel paradigm for building adaptive diffusion systems that surpass the limitations of both single-task and multi-task models. The key contributions of this paper are as follows:

- (1) We propose a graph-based representation of agent diffusers to capture the diverse features and performance metrics of each model within the pool. To this end, we introduce several strategies for constructing the graph of interacting models, facilitating more efficient selection of the optimal subset of agents.
- (2) We develop a **Graph Convolutional Neural Network (GCNN)** as a meta-model to learn optimal task execution by leveraging the structural properties of the graph. The GCNN minimizes a novel composite loss function that accounts for model diversity, prediction accuracy, and the hierarchical structure of the expert agents. We also provide a theoretical analysis of this loss function, highlighting its effectiveness in multi-agent coordination.
- (3) We conduct extensive experiments to evaluate the performance of **LGR-ED** on well-established benchmarks. Our results demonstrate that LGR-ED consistently outperforms previous state-of-the-art solutions, achieving superior output quality across a range of computer vision tasks.

1 RELATED WORK

Diffusion Models. Diffusion models [15, 27, 47] represent a significant advancement within the family of generative models. Previously, Generative Adversarial Networks (GANs) [3] were predominantly utilized for generative tasks. However, diffusion and score-based generative models have demonstrated notable improvements, particularly in the domain of image synthesis [5, 16, 48]. Several initiatives in the computer vision community attempt to improve the learning of diffusion models by updating the architectures used in image encoding and decoding [7, 18, 39]. Despite the

significant success of these models in generating high-quality images across diverse prompt contexts, they may still produce images that do not align with the specified user prompt.

Model Ensembling and Merging. Model ensembling and merging are effective techniques for enhancing model performance and have been extensively used in various computer vision tasks [1, 36, 52]. MagicFusion [51] is a notable approach in this domain, focusing on diffusion models by fusing the predicted noises from two expert U-Net denoisers to achieve applications such as style transfer and object binding. Additionally, several intuitive merging-based methods have been developed. A popular approach is Weighted Merging [24], which involves manually assigning weights to merge each U-Net parameter across multiple shared models. Despite its simplicity, Weighted Merging tends to coarsely distribute weights across all U-Net blocks. These ensemble and merging models face limitations due to decoder performance bottlenecks and the extensive time required for enumerative selection to identify optimal settings.

Discussion. In this work, we build on recent advancements in diffusion models by integrating model ensembling and multi-agent coordination strategies. Our approach aims to enhance the capability of diffusion models to generate high-quality images from user prompts by leveraging the collaborative strengths of multiple model ensembles. Specifically, we treat each model in the ensemble as an individual agent within a multi-agent system, where their coordinated interactions improve image fidelity and coherence. To achieve this, we employ advanced merging methods based on graph-based representations to optimally blend the parameters and outputs of these agents. This graph structure captures the interdependencies between the models, enabling more effective communication and coordination. The diffusion process is thus guided by the collective intelligence of the multi-agent system, addressing the limitations of individual models and resulting in superior image generation that meets user requirements with greater precision and detail.

2 BACKGROUND

Text-to-Image Diffusion-Based Generation Problem. Given a text description T , the goal is to generate a corresponding image I that visually represents the content of T . This is achieved using a diffusion-based generative model. The diffusion process involves a forward process that progressively adds noise to the image and a reverse process that denoises it to generate a realistic image from random noise, guided by the text description T [23, 43, 44].

Forward Diffusion Process. The forward diffusion process is defined as a Markov chain that gradually adds Gaussian noise to the image. Let x_0 be the initial image (which is the final output the model aims to learn to generate), and x_t be the image at time step t [6, 11, 45]. The forward process can be expressed as: $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)\mathbf{I})$. α_t is a variance schedule that controls the amount of noise added at each step.

Reverse Diffusion Process. The reverse diffusion process aims to recover x_0 from x_T (pure noise) by denoising [10, 14, 22]. This process is guided by a model p_θ parameterized by θ , conditioned on the text T :

$$p_\theta(x_{t-1}|x_t, T) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t, T), \Sigma_\theta(x_t, t)) \quad (1)$$

μ_θ and Σ_θ are the predicted mean and covariance of the Gaussian distribution at step t , conditioned on the noisy image \mathbf{x}_t and the text T .

Training Objective. The model is trained to minimize the variational bound on the negative log-likelihood of the data, leading to the following loss function:

$$L(\theta) = \mathbb{E}_q \left[\sum_{t=1}^T D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \right] \quad (2)$$

where D_{KL} denotes the Kullback-Leibler divergence [37].

Generation Process. To generate an image from a text description T , the process starts with a sample from a Gaussian distribution $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively applies the reverse diffusion steps, $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, T)$, until \mathbf{x}_0 is obtained, which is the final generated image.

3 LGR-AD: LEARNING GRAPH REPRESENTATION FOR AGENT DIFFUSERS

3.1 Principle

In this section, we introduce a novel approach called Learning Graph Representation for Agent Diffusers (LGR-AD), which integrates principles from diffusion models with graph-based representation to enhance the text-to-image generation process. The conceptual framework of LGR-AD is illustrated in Figure 1. This approach comprises several distinct stages, each contributing to the creation of integrated expert diffusers ensemble. Initially, a diverse set of diffusers is trained using appropriate datasets and algorithms, encompassing various architectures and learning strategies. Upon completing the training phase, the resultant outputs and specifications of each individual model are stored within a knowledge base. This repository of model-specific information serves as the foundation for constructing a graph representation. The process of creating the model graph begins with utilizing the accumulated knowledge base. Each trained model is represented as a node in the graph, and relationships between these nodes are established based on observed interconnections and similarities among the models. This results in a graph where nodes denote the trained models, and edges signify the associations between these models, reflecting their shared characteristics and behaviors. Subsequently, a Graph Convolutional Neural Network (GCNN) is deployed to extract insights from the constructed model graph. The primary objective of this step is to capture the correlations and dependencies that exist between the various models. The GCNN transforms the information embedded within the model graph into a lower-dimensional vector space, effectively capturing nuanced relationships that might not be apparent in the original high-dimensional space. The GCNN phase produces an embedded vector that captures the relationships among the trained models. This vector, a condensed representation of the ensemble’s collective behavior, is then processed by a fully connected layer to generate a final output image tailored to the user’s objective. The subsequent sections provide detailed descriptions of LGR-AD’s core components.

3.2 Graph Representation for Models

Definition 3.1 (Model Output). We define the set of outputs for model \mathcal{M}_i of the agent \mathcal{A}_i as the union of all its outputs on the training dataset D . Formally, we write:

$$\mathcal{Y}_i^* = \left\{ \bigcup_{D_j \in D} y_{ij}^* \right\} \quad (3)$$

where, y_{ij}^* is the predicted value of D_j by the model \mathcal{M}_i of the agent \mathcal{A}_i .

Definition 3.2 (Model Specification). We define the set of model specification \mathcal{S}_i of the model \mathcal{M}_i by the set of the representative layers of the model \mathcal{M}_i . For instance, $\mathcal{S}_i = \{\text{conv} = 1, \text{pool} = 1, \text{att} = 0, \text{bn} = 0, \text{dr} = 1\}$ to represent that the model \mathcal{M}_i has a convolution layer, a pooling layer, no attention layer, no batch normalization layer, and has a dropout layer.

Definition 3.3 (Connectivity Function). Consider a set of n models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_n\}$, each model \mathcal{M}_i represents the behaviour of the agent \mathcal{A}_i , we define f_{ij} a function that connects the two models \mathcal{M}_i , and \mathcal{M}_j , and we write:

$$f : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$$

Definition 3.4 (Characteristic Connectivity Function). We define the Characteristic Connectivity Function (CCF) as a connectivity function that compute the similarity between two models specification, and we write:

$$\text{CCF}(\mathcal{M}_i, \mathcal{M}_j) = |\mathcal{S}_i \cap \mathcal{S}_j| \quad (4)$$

Definition 3.5 (Performance Connectivity Function). We define the Performance Connectivity Function (PCF) as a connectivity function that computes the similarity between two models output, and we write:

$$\text{PCF}(\mathcal{M}_i, \mathcal{M}_j) = |\mathcal{Y}_i^* \cap \mathcal{Y}_j^*| \quad (5)$$

Definition 3.6 (Graph of Models). We define the graph of models $\mathcal{G}_M = (V, E)$ by the set of nodes V represented the set of models \mathcal{M} , and a set of edges, $E \subseteq V \times V$. $(\mathcal{M}_i, \mathcal{M}_j) \in E$, if and only if, $f_{ij} \neq 0$. f_{ij} is calculated as $\text{CCF}(\mathcal{M}_i, \mathcal{M}_j)$ or $\text{PCF}(\mathcal{M}_i, \mathcal{M}_j)$ depending to the connectivity function chosen by the user.

3.3 Maximum Spanning Tree for Agent Diffusers

Let $G = (\mathcal{A}, E)$ be a graph, where \mathcal{A} is the set of agents and E represents the edges (relationships) between these agents. Each edge $e_{ij} \in E$ between agents \mathcal{A}_i and \mathcal{A}_j is weighted by a similarity function $w(e_{ij})$ that quantifies the shared properties between the two agents, based on their performance metrics or model characteristics. The objective of the Maximum Spanning Tree (MST) algorithm is to select a subset of edges $E' \subseteq E$ such that, $T = (\mathcal{A}, E')$ is a tree (i.e., an acyclic, connected graph), and the total weight of the tree is maximized:

$$\max_T \sum_{e_{ij} \in E'} w(e_{ij}) \quad (6)$$

The following formal description captures the distinct roles of the MST in LGR-AD:

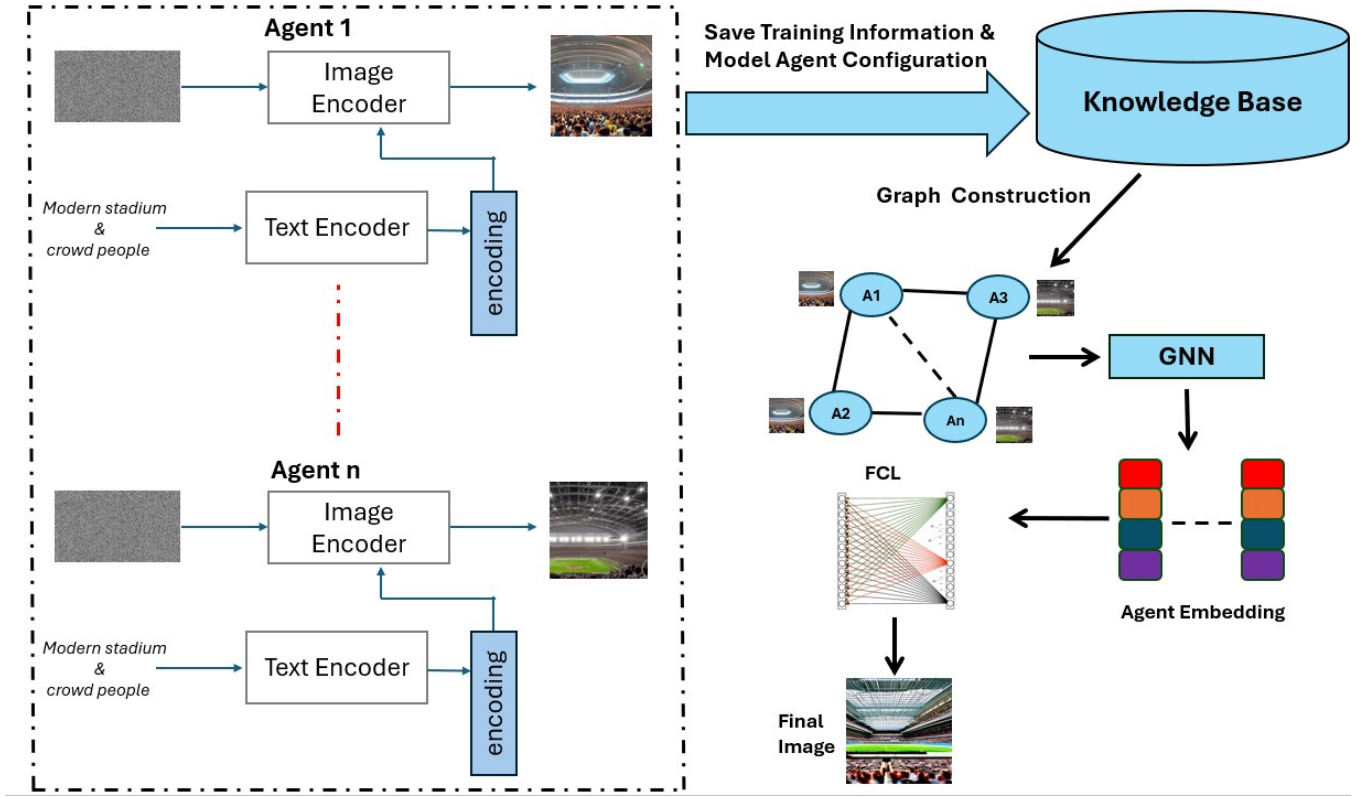


Figure 1: LGR-AD begins by training diffusion-based models for text-to-image generation, treating each model as an agent in a multi-agent system. After training each agent diffuser, we use the agents’ outputs and specifications to construct a graph, where nodes represent agents and edges capture their interactions. GCNN is then applied to learn an optimal representation of the graph, leveraging agent collaboration. The learned features are processed through a fully connected layer to guide image generation, enabling the system to adapt and refine the diffusion process for improved fidelity and coherence.

1. Relationships Capture: The MST emphasizes the most significant relationships among agents, capturing the most relevant shared characteristics. For each pair of agents $\mathcal{A}_i, \mathcal{A}_j \in \mathcal{A}$, we define a weight function $w(e_{ij})$ that assigns a scalar value based on the similarity of their outputs. The MST maximizes the sum of these weights:

$$T^* = \arg \max_T \sum_{e_{ij} \in E'} w(e_{ij}) \quad (7)$$

where T^* is the optimal tree structure that spans all agents with the most significant relationships. By focusing on the highest weights, the MST filters out weaker or less relevant connections, providing a concise representation of the most critical relationships.

2. Reduced Redundancy: The MST algorithm prevents cycles, ensuring that redundant connections between agents are removed. This is achieved by enforcing the acyclic property of the tree. Let $T = (\mathcal{A}, E')$ represent the maximum spanning tree, where $E' \subseteq E$ is the set of selected edges. The acyclic property ensures:

$$\forall \mathcal{A}_i, \mathcal{A}_j \in T, \text{ there exists at most one path from } \mathcal{A}_i \text{ to } \mathcal{A}_j.$$

This property reduces redundancy, capturing only the most informative relationships among agents.

3. Hierarchical Information: The hierarchical structure of the MST reflects the importance of relationships between agents. The tree defines a hierarchy of shared properties, starting with general properties and refining down to more specific ones. For any pair of agents \mathcal{A}_i and \mathcal{A}_j , their depth in the tree reflects the degree of shared characteristics. Formally, the hierarchy can be captured by defining the depth $d(\mathcal{A}_i)$ of each node $\mathcal{A}_i \in \mathcal{A}$ in the tree:

$$d(\mathcal{A}_i) = \min_{\text{paths}} \sum_{e_{ij} \in T} w(e_{ij}) \quad (8)$$

where deeper nodes in the tree represent agents with more specialized relationships. The hierarchical nature of the MST provides insights into how closely the agents are related, both in general and in specific terms.

4. Ensemble Composition: By selecting only the most significant edges in the MST, we form a focused and diverse ensemble of models. The ensemble \mathcal{E} consists of agents connected by the edges of the MST. This selective process can be formulated as:

$$\mathcal{E} = \{\mathcal{A}_i \mid \mathcal{A}_i \in T^*\} \quad (9)$$

The collective performance of this ensemble is enhanced by emphasizing relationships that provide complementary strengths

across agents. By maximizing the relevant shared properties, the ensemble is designed to leverage the diversity of the agents, leading to improved performance across a variety of tasks.

3.4 Intuition of the loss function in Agent Diffusers

An important contribution of the LGR-AD algorithm is the introduction of a novel loss function, $L(x)$ used to train Graph Convolutional Neural Network.

Definition 3.7 (Loss Function). We define $L(x)$ as:

$$L(x) = C(x) + \lambda D(x) + \gamma L_{\text{laplace}} \quad (10)$$

where x is the input image text, $C(x)$ is the Cross-Entropy loss, $D(x)$ is the Kullback-Leibler Divergence loss, λ and γ are regularization parameters. The L_{laplace} is a modified version of the Laplacian loss to ensure the model's predictions align with the underlying graph structure [21].

To promote diversity, the Kullback-Leibler (KL) Divergence loss needs to be maximized. To incorporate this into the overall loss function, which requires minimization, we introduce the term $D(x)$. This adjustment allows us to integrate the KL Divergence effectively into the global loss function. The resulting loss function combines the benefits of accuracy, diversity, and structural alignment, providing a comprehensive and versatile optimization objective.

Diversity and Accuracy. The Cross-Entropy loss, calculated between the final generated image and the ground truth image, ensures that the agent of diffusers is accurate, while the Kullback-Leibler Divergence enforces diversity among the models.

Regularization. The λ term acts as a regularization parameter. A higher λ gives more importance to the Kullback-Leibler Divergence, encouraging the ensemble to pay more attention to the diversity of selected agent diffusers.

Laplacian Loss. The Laplacian loss, derived from the Laplacian matrix of the graph, encourages the model to respect the graph's structure. In the context of Maximum Spanning Tree, this loss ensures that the model's predictions align with the underlying tree structure, promoting a more structured prediction.

Trade-off. The hybrid loss allows for a balance between text-to-image generation accuracy, distributional similarity, and graph structure adherence, offering the 'best of both worlds'.

Robustness. The hybrid loss can make the ensemble more robust. While Cross-Entropy loss ensures text-to-image generation accuracy, the Kullback-Leibler part can make the ensemble robust to variations in the input distribution.

Optimal Weighting. The hybrid loss can provide a good performance metric, allowing for potentially more effective weighting of individual models in the ensemble.

3.5 Graph Convolutional Neural Network (GCNN) for Agent Diffusers

In this section, we propose the use of a GCNN to effectively model the relationships between agents in the LGR-AD framework. MST,

which captures the most significant relationships among the agents, is utilized as the underlying structure of the graph. The GCNN learns a representation of each agent's state, allowing the agents to communicate and collaborate, thus improving their collective performance. The weights of the graph G on the edges are derived from the MST, with each edge weight $w(e_{ij})$ representing the strength of the relationship between agents v_i and v_j . GCNN layer updates the feature vector of each agent by aggregating information from its neighbors in the graph. Formally, the update rule for each GCNN layer is given by:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{M}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \quad (11)$$

where M is the adjacency matrix between the different agents. $\mathbf{H}^{(l)} \in \mathbb{R}^{|V| \times d^{(l)}}$ is the feature matrix at layer l , with $d^{(l)}$ being the dimensionality of the feature space at layer l . $\tilde{\mathbf{M}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{M}\mathbf{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix, where \mathbf{D} is the diagonal degree matrix with $D_{ii} = \sum_j M_{ij}$. $\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$ is the learnable weight matrix at layer l . $\sigma(\cdot)$ is a non-linear activation function, such as ReLU.

The GCNN processes multiple layers of graph convolutions, allowing each agent to aggregate information from its local neighborhood in the graph. After L layers, the output matrix $\mathbf{H}^{(L)} \in \mathbb{R}^{|V| \times d^{(L)}}$ represents the learned embeddings for each agent, capturing both their individual properties and the relationships with other agents as encoded by the MST. For the final prediction, a fully connected layer is applied to each node's embedding to generate the output for each agent:

$$y_i = \text{softmax}(\mathbf{W}^{\text{out}}\mathbf{h}_i^{(L)} + \mathbf{b}^{\text{out}}) \quad (12)$$

where $\mathbf{W}^{\text{out}} \in \mathbb{R}^{d^{(L)} \times c}$ is the weight matrix for the final classification or regression task, \mathbf{b}^{out} is the bias term, and c is the number of output classes or the dimensionality of the regression target.

MST provides the structural backbone for the GCNN, ensuring that the most relevant and informative relationships among agents are emphasized. By using the MST-based adjacency matrix, the GCNN learns to propagate information primarily along the most critical edges, leading to a more efficient and focused learning process. This hierarchical and non-redundant structure enables the system to leverage the complementary strengths of the agents, improving the overall performance of the model.

3.6 Pseudo-Code

The LGR-AD algorithm enhances the performance of a set of "agent diffusers" for text-to-image generation by explicitly modeling the relationships between them. It begins by constructing a graph where each node represents a diffuser from the set (\mathcal{M}) . The edges between these nodes are weighted based on a function f that assesses the relationship or similarity between the outputs of the two connected diffusers (using training data \mathcal{D}). To focus on the most important connections, Maximum Spanning Tree (MST) is extracted from this graph, the number of MSTs can be too many, and thus increasing the computational cost since one MST is built in $O(m \log n)$, when all of the MSTs are considered it will be $O(d \cdot m \log n)$ where d is the all the number of trees and m is the number of edges and n is

Algorithm 1: LGR-AD Algorithm

Data: Set of diffusers \mathcal{M} , Training data \mathcal{D}
Result: Trained LGR-AD model

```

1 Initialize GCNN model with parameters  $\theta$ 
2 for each epoch do
3   for each  $x \in \mathcal{D}$  do
4      $G \leftarrow \emptyset$  // Initialize the graph
5     foreach  $(m_i, m_j) \in \mathcal{M}^2$  s.t.  $i < j$  do
6        $w_{ij} = f(m_i(x), m_j(x))$  // Compute weight between models
7       using  $f$ 
8        $G \leftarrow G \cup \{(m_i, m_j, w_{ij})\}$  // Add edge with weight to the
9       graph
10    MST  $\leftarrow$  Extract  $k$  Maximum Spanning Trees from  $G$ 
11     $A_{MST}, \mathbf{h} \leftarrow GCNN(MST)$ 
12     $L_{\text{laplace}} \leftarrow \frac{1}{2} \sum_{i,j} A_{MST_{ij}} \|\mathbf{h}_i - \mathbf{h}_j\|^2$ 
13     $L(x) = C(x) + \lambda D(x) + \gamma L_{\text{laplace}}$  Compute the loss  $L(x)$ 
14    Compute gradients  $\nabla L(\theta)$ 
15    Update model parameters  $\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t)$ 
16 return Trained GCNN model with parameters  $\theta$ 

```

the number of vertices. To overcome this only a small number k of MSTs is sampled thus resulting in $O(k.m.\log n)$ where k is a bounded number of MSTs (in our case it is 1). A Graph Convolutional Neural Network (GCNN) is then employed to learn from this MST. LGR-AD introduces a "Laplacian loss" ($L_{\text{laplace}} = \frac{1}{2} \sum_{i,j} A_{MST_{ij}} \|\mathbf{h}_i - \mathbf{h}_j\|^2$) that encourages the GCNN to learn similar embeddings (\mathbf{h}_i) for diffusers that are strongly connected in the MST. Here, $A_{MST_{ij}}$ is an element in the adjacency matrix of the MST, indicating the connection strength. This loss, in essence, promotes smoothness in the embedding space, ensuring that closely related diffusers have similar representations.

The GCNN is trained using a hybrid loss function that combines this Laplacian loss with the primary task loss (e.g., image reconstruction error) and a regularization term. This combined loss ensures that the model achieves good performance on the main task while also respecting the relationships between the diffusers captured in the MST. By leveraging these relationships, LGR-AD facilitates a more focused and effective learning process, leading to improved performance in text-to-image generation.

4 EXPERIMENTAL STUDIES

4.1 Experimental Settings

In this study, we employed an ensemble of agent diffusion models to achieve high-quality image synthesis. Each model in the ensemble was carefully selected to cover a diverse range of capabilities and specialties. Below, we outline the configuration and hyperparameters for each of the expert models used in our ensemble.

Model Configurations. Table 1 provides an overview of the expert models used in our ensemble, including their model IDs, the number of parameters, and the datasets on which they were pre-trained.

Inference Settings. Table 2 summarizes the inference settings, including the number of inference steps and the guidance scale used during image generation.

The performance of the ensemble was evaluated based on well-known metrics including Fréchet Inception Distance (FID) [19], Inception Score (IS) [9], and CLIP score [49].

Table 1: Configurations of Agent Diffusers. All datasets are large image-text pairs collections.

Model	Param.(B)	Pre-training Dataset
DALL-E 2 [30]	12	Internal OpenAI
Stable Diffusion v2 [31]	1.5	Filtered subset of LAION-5B [33]
LDM [31]	0.4	LAION-400M [34]
Imagen [32]	1	COYO-700M [4]; internal Google

Table 2: Inference Settings

Model	Inference Settings	
	Num Inference Steps	Guidance Scale
DALL-E 2 [30]	100	10.0
Stable Diffusion v2 [31]	50	7.5
LDM [31]	75	9.0
Imagen [32]	60	8.5

We evaluated our approach using a suite of datasets that are widely recognized and frequently utilized in the literature [42, 50, 53]. The selected datasets encompass a diverse range of images and include (MSCOCO (Microsoft Common Objects in Context) [8], CUB (Caltech-UCSD Birds-200-2011) [38], LN-COCO (Large-scale Noisy-COCO) [29], Multi-modal CelebA-HQ (MM CelebA-HQ) [41]). A detailed description of the datasets can be found in the Appendix.

4.2 Numerical Results

Models	FID ↓	IS ↑	CLIP Score ↑
StableDiffusion [31]	9.91	85	0.31
LatteGAN [25]	11.05	74	0.28
DALL-E 2 [30]	9.91	112	0.30
Stable Diffusion v2 [31]	9.78	89	0.33
LDM [31]	10.32	83	0.28
Imagen [32]	9.95	116	0.35
MagicFusion [51]	10.94	92	0.29
LGR-AD (Our)	9.52	129	0.36

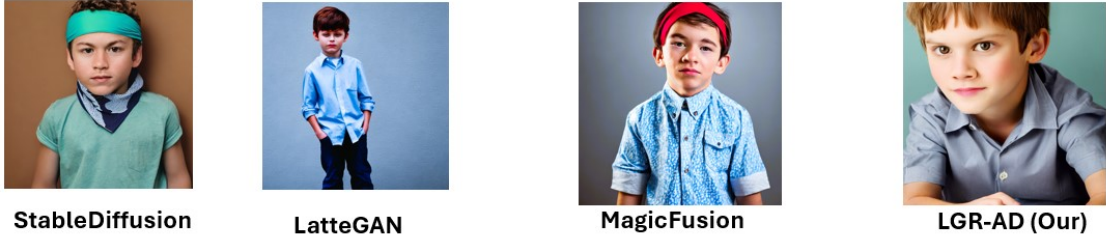
Table 3: Comparison of the SOTA text-to-image generation models on MSCOCO.

Models	FID ↓	IS ↑	CLIP Score ↑
StableDiffusion [31]	10.15	82	0.39
LatteGAN [25]	10.89	75	0.42
DALL-E 2	9.95	79	0.44
Stable Diffusion v2 [31]	9.74	84	0.41
LDM [31]	10.33	79	0.34
Imagen [32]	10.15	93	0.44
MagicFusion [51]	9.91	77	0.40
LGR-AD (Our)	9.55	105	0.48

Table 4: Comparison of the SOTA text-to-image generation models on CUB.

We compare the image generation capabilities of our model against existing baselines, including both stand-alone models and state-of-the-art model ensemble method. The results are presented in Tables 3, 4, 5, and 6. Our findings consistently show that LGR-AD achieves superior performance across all datasets and evaluation metrics. Lower FID scores indicate that LGR-AD generates images with a distribution closer to the real images, suggesting high quality and

Boy is wearing a grey shirt with green eyes and yellow hair.



Photorealistic image of a big stadium with high light, and crowd people.

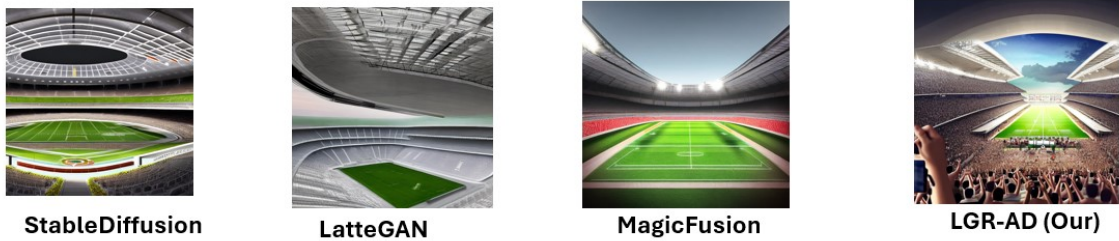


Figure 2: Selected qualitative results of LGR-AD compared to the baseline solutions.

Models	FID ↓	IS ↑	CLIP Score ↑
StableDiffusion [31]	9.66	115	0.39
LatteGAN [25]	10.85	109	0.37
DALL-E 2 [30]	9.87	118	0.40
Stable Diffusion v2 [31]	9.50	123	0.42
LDM [31]	10.25	107	0.36
Imagen [32]	9.89	135	0.43
MagicFusion [51]	10.71	111	0.38
LGR-AD (Our)	9.44	149	0.45

Table 5: Comparison of the SOTA text-to-image generation models on LN-COCO.

Models	FID ↓	IS ↑	CLIP Score ↑
StableDiffusion [31]	9.70	154	0.49
LatteGAN [25]	10.13	141	0.48
DALL-E 2 [30]	9.91	144	0.54
Stable Diffusion v2 [31]	9.57	172	0.51
LDM [31]	10.15	123	0.51
Imagen [32]	9.96	185	0.53
MagicFusion [51]	9.83	129	0.52
LGR-AD (Our)	9.50	190	0.57

Table 6: Comparison of the SOTA text-to-image generation models on MM CelebA-HQ.

Method	Backbone	Dataset	Diversity↑
InstantBooth [35]	Stable Diffusion	PPR10K	0.45
DM-GAN [53]	ResNet-101	CUB-200	0.55
LAFITE [46]	ViT-B	FashionGen	0.58
Stable Diffusion [31]	ViT-L	FashionGen	0.65
VQGAN-CLIP [12]	ViT-B	Oxford Pets	0.52
T2I-Adapter [26]	ViT-B	Emogen	0.63
Ours	ViT-L	COCO	0.67

Table 7: Comparison of Methods on Diversity Metric

realism. Higher IS values reflect better diversity and perceptual quality of the generated images. Finally, higher CLIP scores indicate a better alignment between the generated images and their corresponding text descriptions, demonstrating the model’s effectiveness in capturing semantic content. These results underscore the usefulness of graph-based representations in diffusion models ensembles. The graph-based approach employed by LGR-AD allows for structured and context-aware representations, leading to an enhanced image quality and a better alignment with textual descriptions. Our proposed methodological innovation not only improves image generation performance on standard benchmarks but also highlights the potential for broader applicability in various computer vision tasks. The superior performance of LGR-AD across multiple datasets and evaluation metrics establishes it as a robust and versatile model for image generation, setting a new standard for future research in this domain. Table 7 compares several text-to-image generation methods based on the diversity metric across various datasets and backbone architectures. The diversity metric is defined as the average pairwise distance between generated image embeddings, with higher values indicating greater diversity among generated samples. The datasets range from domain-specific collections like CUB-200 (birds) and FashionGen (fashion) to general-purpose datasets like COCO, offering a broad evaluation scope. Ours achieves the highest diversity score of 0.67 using the ViT-L backbone and COCO dataset, indicating the method’s ability to produce a wide variety of images while preserving coherence to the input text. Classical methods such as DM-GAN, which employs a ResNet-101 backbone, yield relatively lower diversity (0.55) due to limitations in capturing nuanced text-to-image mappings. This achievement stems from the seamless integration of multi-expert

collaboration and dynamic coordination mechanisms facilitated by graph neural networks.

4.3 Qualitative Results

As depicted in Figure 2, we conduct a qualitative comparison between the outputs of the baseline methods, and our proposed LGR-AD approach. This analysis yields several significant observations. Firstly, LGR-AD exhibits a remarkable ability to generate images with enhanced aesthetic quality. Specifically, LGR-AD consistently produces images with superior composition and finer details compared to established methods such as StableDiffusion [31], LatteGAN [25], and MagicFusion [51]. Secondly, LGR-AD demonstrates a significant advantage in achieving superior context alignment. For instance, all baseline methods generate images depicting an empty stadium, failing to capture the context accurately. In contrast, our LGR-AD method uniquely generates the image of a stadium filled with a crowd, accurately reflecting the provided context and thereby demonstrating its superior contextual understanding. These superior results are attributed to the advanced learning representation mechanisms employed by LGR-AD. The model effectively leverages the correlations among various diffusers, which enhances the learning process and leads to a significant improvement in the quality of the generated images. This efficient representation learning enables LGR-AD to outperform existing methods in both aesthetic quality and contextual alignment, as evidenced by the comparative analysis.

4.4 Ablation Study

Models	FID ↓	IS ↑	CLIP Score ↑
(DALL-E 2, Stable Diff. v2)	10.38	119	0.38
(DALL-E 2, LDM)	10.51	115	0.36
(DALL-E 2, Imagen)	9.91	125	0.38
(Stable Diff. v2, LDM)	10.31	121	0.35
(Stable Diffusion v2, Imagen)	9.95	131	0.39
(Imagen, LDM)	9.96	128	0.35
(DALL-E 2, Stable Diff. v2, Imagen)	9.69	139	0.43
(DALL-E 2, Stable Diff. v2, LDM)	9.75	130	0.40
(DALL-E 2, Imagen, LDM)	9.80	129	0.40
(Stable Diff. v2, Imagen, LDM)	9.81	133	0.42
LGR-AD (All models)	9.50	143	0.46

Table 8: Varying the Number of Models of LGR-AD.

Models	FID ↓	IS ↑	CLIP Score ↑
CCF	9.55	139	0.44
PCF	9.53	140	0.44
Hybrid	9.50	143	0.46

Table 9: Varying the Connectivity Function of LGR-AD.

In our ablation study, we investigate various configurations of LGR-AD to identify the optimal setup for high-quality image generation. Specifically, we experiment with the number of models used in the expert diffusers, ranging from two to four models, and examined different connectivity functions utilized in the graph construction. The models included in the four-model configuration were Stable Diffusion v2, Imagen, DALL-E 2, and LDM. The connectivity functions used are: CCF, PCF, and a Hybrid function that combines

both. The results presented in Table 8 demonstrate a substantial advantage in utilizing all four models (Stable Diffusion v2, Imagen, DALL-E 2, and LDM) compared to using only a subset of these models. This is explained by the fact that each model contributes unique strengths, and their combined use results in a more robust generation process. The results shown in Table 9 also highlight the importance of using both connectivity functions (CCF and PCF) to better capture the various correlations and dependencies among the agent diffusers in LGR-ED. Indeed, incorporating both CCF and PCF provides a more nuanced understanding of the relationships between different agent diffusers, leading to superior image generation performance. These findings highlight the importance of exploring diverse model architectures and advanced connectivity functions in LGR-AD, pushing the boundaries of image generation quality in computer vision.

5 CONCLUSION

In this paper, we introduced, LGR-AD, a novel multi-agent system that integrates diffusion-based generative models with graph representation techniques for text-to-image generation. Our approach models the generation process as a distributed system of interacting agents, where each agent represents an expert sub-model specializing in different aspects of the task. The outputs of these agents are stored in a knowledge base, which is then used to construct a graph that encodes the relationships and dependencies among the agents. GCNN learns the complex interactions between agents from this graph, while a fully connected layer synthesizes the final image based on the user’s input prompt. Each agent operates with a degree of autonomy, and their collective decision-making is optimized through a meta-model that minimizes a novel loss function. This loss function balances diversity and accuracy across the agents and incorporates a maximum spanning tree approach to enhance coordinated optimization. Our theoretical analysis sheds light on how the graph structure and loss function promote effective collaboration between agents, leading to emergent behaviors that improve overall system performance. Empirical results demonstrate that LGR-AD outperforms traditional diffusion models across various computer vision benchmarks by leveraging multiple specialized agents. This adaptable and scalable multi-agent framework addresses complex tasks effectively. Future work will focus on enabling agents to specialize in subtasks while sharing knowledge, enhancing robustness and adaptability. Additionally, incorporating game-theoretic strategies for agent collaboration could further improve contextual text-to-image generation, highlighting LGR-AD’s potential to advance adaptive image synthesis.

ACKNOWLEDGMENTS

This work is funded by the Research Council of Norway under the project entitled "Next Generation 3D Machine Vision with Embedded Visual Computing" under grant number 325748. It is also partially funded by the European Commission through the AI4CCAM project (Trustworthy AI for Connected, Cooperative Automated Mobility) under grant agreement number 101076911, and the National Science Centre in Poland, under grant agreement number 2020/39/O/ST6/01478.

REFERENCES

- [1] Rouqia Al-Refai and Karthik Nandakumar. 2023. A unified model for face matching and presentation attack detection using an ensemble of vision transformer features. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 662–671.
- [2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. 2022. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324* (2022).
- [3] Matyáš Boháček and Hany Farid. 2023. A geometric and photometric exploration of gan and diffusion synthesized faces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 874–883.
- [4] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. 2022. COYO-700M: Image-Text Pair Dataset. <https://labelbox.com/datasets/coyo-700m-image-text-pair-dataset/>
- [5] Chenjie Cao, Yunuo Cai, Qiaole Dong, Yikai Wang, and Yanwei Fu. 2024. LeftRefill: Filling Right Canvas based on Left Reference through Generalized Text-to-Image Diffusion Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7705–7715.
- [6] Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. 2024. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [7] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. 2023. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19830–19843.
- [8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
- [9] Min Jin Chong and David Forsyth. 2020. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6070–6079.
- [10] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. 2023. Parallel diffusion models of operator and image for blind inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6059–6069.
- [11] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
- [12] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. 2022. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *European Conference on Computer Vision*. Springer, 88–105.
- [13] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2426–2436.
- [14] Guy Gilboa, Nir Sochen, and Yehoshua Y Zeevi. 2002. Forward-and-backward diffusion processes for adaptive image enhancement and denoising. *IEEE transactions on image processing* 11, 7 (2002), 689–703.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- [16] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research* 23, 47 (2022), 1–33.
- [17] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. 2024. Soda: Bottleneck diffusion models for representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 23115–23127.
- [18] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. 2023. Ddp: Diffusion model for dense visual prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 21741–21752.
- [19] Steffen Jung and Margret Keuper. 2021. Internalized biases in fr chet inception distance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*.
- [20] Subhadeep Koley, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. 2024. Text-to-Image Diffusion Models are Great Sketch-Photo Matchmakers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16826–16837.
- [21] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2017. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 624–632.
- [22] Charles Laroche, Andr s Almansa, and Eva Coup te. 2024. Fast diffusion em: a diffusion model for blind inverse problems with application to deconvolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 5271–5281.
- [23] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. 2023. Instaflo: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*.
- [24] Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems* 35 (2022), 17703–17716.
- [25] Shoya Matsumori, Yuki Abe, Kosuke Shingyouchi, Komei Sugiyura, and Michita Imai. 2021. LatteGAN: Visually Guided Language Attention for Multi-Turn Text-Conditioned Image Manipulation. *IEEE Access* 9 (2021), 160521–160532. <https://doi.org/10.1109/access.2021.3129215>
- [26] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 4296–4304.
- [27] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*. PMLR, 8162–8171.
- [28] Liang Peng, Yujie Mo, Jie Xu, Jialie Shen, Xiaoshuang Shi, Xiaoxiao Li, Heng Tao Shen, and Xiaofeng Zhu. 2023. GRLC: Graph representation learning with constraints. *IEEE Transactions on Neural Networks and Learning Systems* (2023).
- [29] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 647–664.
- [30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.
- [31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj rn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- [32] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karag l Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.
- [33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5B: An open large-scale dataset for training next generation image-text models. *arXiv:2210.08402 [cs.CV]* <https://arxiv.org/abs/2210.08402>
- [34] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. *arXiv:2111.02114 [cs.CV]* <https://arxiv.org/abs/2111.02114>
- [35] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. 2024. Instantbooth: Personalized text-to-image generation without test-time finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8543–8552.
- [36] Haoru Tan, Chuang Wang, Sitong Wu, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2024. Ensemble quadratic assignment network for graph matching. *International Journal of Computer Vision* (2024), 1–23.
- [37] Tim Van Erven and Peter Harremo s. 2014. R nyi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* 60, 7 (2014), 3797–3820.
- [38] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. (2011).
- [39] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8228–8238.
- [40] Cong Wang, Kuan Tian, Yonghang Guan, Jun Zhang, Zhiwei Jiang, Fei Shen, Xiao Han, Qing Gu, and Wei Yang. 2024. Ensembling Diffusion Models via Adaptive Feature Aggregation. *arXiv preprint arXiv:2405.17082* (2024).
- [41] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. 2021. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2256–2265.
- [42] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1316–1324.
- [43] Yifei Xu, Xiaolong Xu, Honghao Gao, and Fu Xiao. 2024. SGDM: An Adaptive Style-Guided Diffusion Model for Personalized Text to Image Generation. *IEEE Transactions on Multimedia* (2024).
- [44] Ling Yang, Jingwei Liu, Shenda Hong, Zhilong Zhang, Zhilin Huang, Zheming Cai, Wentao Zhang, and Bin Cui. 2024. Improving diffusion-based image synthesis with context prediction. *Advances in Neural Information Processing Systems* 36 (2024).

- [45] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *Comput. Surveys* 56, 4 (2023), 1–39.
- [46] Haonan Yin, Guanlong Jiao, Qianhui Wu, Borje F Karlsson, Biqing Huang, and Chin Yew Lin. 2023. Lafite: Latent diffusion model with feature editing for unsupervised multi-class anomaly detection. *arXiv preprint arXiv:2307.08059* (2023).
- [47] Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. 2023. Video probabilistic diffusion models in projected latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18456–18466.
- [48] Yu Zeng, Vishal M Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. 2024. JeDi: Joint-Image Diffusion Models for Finetuning-Free Personalized Text-to-Image Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6786–6795.
- [49] Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023. Text-to-image diffusion models in generative ai: A survey. *arXiv preprint arXiv:2303.07909* (2023).
- [50] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. 2021. Cross-modal contrastive learning for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 833–842.
- [51] Jing Zhao, Heliang Zheng, Chaoyue Wang, Long Lan, and Wenjing Yang. 2023. MagicFusion: Boosting Text-to-Image Generation Performance by Fusing Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22592–22602.
- [52] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. 2024. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 23554–23564.
- [53] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. 2019. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5802–5810.