DA-DFGAS:Differentiable Federated Graph Neural Architecture Search with Distribution-Aware Attentive Aggregation

Zhaowei Liu[1*], Yihao Jiang[1], Rufei Gao[1], Jinglei Liu[1], Dong Yang[2]

[1]School of Computer and Control Engineering, Yantai University, Shandong, China
[2]School of the Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
lzw@ytu.edu.cn, yihaojiang2023@s.ytu.edu.cn, gaorufei@yitsd.edu.cn, Jinglei_liu@sina.com, dyang26@student.gsu.edu

Abstract

Graph Neural Networks (GNNs) have demonstrated superior performance in processing centralized graph-structured data. However, real-world privacy and security concerns hinder data centralization and shareing, leading to severe data isolation (data silos). While Federated Learning (FL) offers a distributed solution to mitigate these obstacles, existing Federated Graph Neural Network (FedGNN) frameworks struggle to effectively address data heterogeneity. To address this, this paper proposes DA-DFGAS, a federated graph neural architecture search algorithm. Specifically, DA-DFGAS facilitates model personalization via a directed tree topology and path constraint mechanisms, while simultaneously employing a joint self-attention mechanism based on predicted probability distributions to capture distributional variations across multiple clients. Furthermore, it integrates a bi-level global-local objective optimization strategy to ensure global model consistency while preserving local adaptability. Experimental results on multiple datasets demonstrate that DA-DFGAS outperforms state-of-the-art methods, achieving 0.5–3.0% accuracy improvements over centralized baselines and 0.5–5.0% over federated counterparts.

Introduction

Non-Euclidean data (Wu et al. 2020), characterized by its irregular structure and lack of translation invariance, encompasses a wide range of real-world information, from protein molecular structures to urban traffic networks. In the deep learning community, such data is generally represented as graphs. Over the past few years, Graph Neural Networks (GNNs) (Shao et al. 2024) have emerged as the dominant paradigm for machine learning on graph data (Hamilton, Ying, and Leskovec 2017). By iteratively propagating information layer-by-layer, GNNs can effectively learn node representations and capture structural dependencies from neighbors. This capability enables GNNs (Meng et al. 2025b,a) to handle diverse tasks such as recommendation systems (Huang et al. 2024; Sang et al. 2025), node classification (Zhang et al. 2023a; Xia et al. 2024), and traffic prediction (Khaled et al. 2024; Wang et al. 2023).

*Corresponding author.

However, in real-world scenarios, valuable graph data is often distributed across multiple edge devices and cannot be centrally shared due to strict privacy regulations and commercial barriers. Furthermore, since data on these edge devices typically originates from diverse domains, it inherently exhibits significant statistical heterogeneity, often referred to as Non-Independent and Identically Distributed (Non-IID) data. This heterogeneity makes designing a single GNN architecture that generalizes effectively across all clients an extremely challenging task.

FL, typified by the seminal FedAvg framework (McMahan et al. 2017), has emerged as a promising solution to address these privacy concerns by enabling collaborative training on decentralized data. Despite its potential, existing Federated GNN (FedGNN) frameworks have not effectively resolved the challenges posed by data heterogeneity. Although auxiliary techniques such as knowledge distillation (Zhang et al. 2024; Tan et al. 2024), meta-learning (Serhani et al. 2025; Wang et al. 2022), and regularization (Li et al. 2020) have been explored, they are often superimposed upon generic frameworks (e.g., FedAvg) that suffer from inherent limitations. These limitations primarily manifest in two critical aspects. First, the reliance on simplistic aggregation algorithms (Xu et al. 2025) fails to adequately balance and integrate contributions from diverse clients, leading to compromised convergence and robustness. Second, existing methods typically enforce a unified global model architecture across all clients. This "one-size-fits-all" approach cannot adapt to the unique data characteristics and task requirements of individual clients (Ju et al. 2024; Zhang et al. 2022), resulting in suboptimal generalization. Consequently, developing a federated framework capable of creating personalized, high-performance GNN architectures to navigate data heterogeneity remains an open challenge.

To address these challenges, we propose DA-DFGAS, a novel FedGNN framework based on Neural Architecture Search (NAS) (Lu et al. 2023; Xie et al. 2024). To mitigate the impact of Non-IID data, DA-DFGAS employs a self-attention federated aggregation mechanism that dynamically quantifies the importance of each client's update to optimize the global model. Furthermore, to overcome the limitations of fixed architectures, DA-DFGAS utilizes differentiable NAS to automatically discover personalized GNN architectures tailored to the local data features of each client.

A bi-level objective function is integrated to simultaneously optimize global model consistency and local personalized performance. Experimental results demonstrate that DA-DFGAS significantly outperforms existing methods in federated graph learning settings.

- This paper proposes DA-DFGAS, a novel framework that integrates a self-attention federated aggregation mechanism with a bi-level global-local objective optimization strategy. This combination effectively balances global consistency with personalized adaptability based on predicted probability distributions.
- DA-DFGAS introduces a federated differentiable NAS strategy that leverages a directed tree topology and path constraint mechanisms. This design enables the dynamic construction of architectures adapted to individual client data characteristics, effectively circumventing the inherent limitations of a static unified global architecture.
- Extensive experiments across six datasets demonstrate that DA-DFGAS consistently outperforms state-of-the-art FedGNN baselines in terms of accuracy and robustness.

Related works

Federated Graph Neural Network

FedGNN extends the capability of modeling graph-structured data to distributed scenarios. Depending on the overlap of feature and sample spaces among client nodes, FedGNNs are generally categorized into horizontal and vertical settings. This paper primarily focuses on horizontal FedGNN, where the key challenge lies in addressing Non-IID graph data. Existing methods can be broadly classified into three categories: model-based, data-based, and aggregation-based methods.

Model-based methods emphasize enhancing the adaptability of local models to local data. For instance, FedEgo (Zhang et al. 2023b) applies GraphSAGE over ego-graphs to maximize structural information utilization while employing Mixup strategies for privacy preservation. Similarly, InfoFedSage (Guo, Li, and Zhang 2023) adopts an information bottleneck-driven approach for federated subgraph learning. Data-based methods aim to mitigate statistical heterogeneity in client data distributions through techniques such as sample reweighting, clustering, and manifold learning. FedACS (Xu et al. 2025) introduces a multi-armed bandit algorithm to select participating clients based on resource budgets and training performance. FedAMP (Huang et al. 2021) utilizes a federated attentive message-passing mechanism to explicitly encourage collaboration among clients with similar data distributions. Aggregation-based methods alleviate the Non-IID issue by recalibrating the importance of each client during the aggregation phase. FedPub (Baek et al. 2023) calculates aggregation weights based on the functional similarity between local model outputs and randomly generated common graphs. Furthermore, FedNor (Xu et al. 2024) integrates robust statistical methods with personalized FL strategies to enhance resilience against malicious attacks while preserving model generalization capabilities.

Neural Architecture Search

NAS treats the design of neural networks as a learnable process, typically formulated as a bi-level optimization problem: the outer loop searches for optimal architectures, while the inner loop optimizes the corresponding model weights. This bi-level paradigm enables the automatic discovery of optimal structural configurations, surpassing the limitations of manual design. When applied to GNNs, NAS necessitates a predefined search space that encompasses micro- and macro-architectures, pooling strategies, and hyperparameters.

Recent works have focused on addressing specific challenges in GNNs through NAS. To tackle distribution shifts, OMG-NAS (Cai et al. 2024) optimizes the architecture using a multimodal graph representation decorrelation strategy and a global sample weight estimator. Similarly, DC-GAS (Yao et al. 2024) incorporates an embedding-guided data generator and a dual-factor uncertainty curriculum weighting strategy to bolster generalization capabilities. Regarding adversarial robustness, G-RNA (Xie et al. 2023) introduces graph structure masking operations into the search space, while LRNAS (Feng et al. 2024) proposes a differentiable search method for robust lightweight architectures. Additionally, RACL (Dong et al. 2025) explores the statistical relationship between the Lipschitz constant and architecture parameters to ensure stability.

NAS search strategies primarily include reinforcement learning, evolutionary algorithms, and differentiable methods (Cai et al. 2021). To further enhance search performance, GASSIP (Xie et al. 2024) proposes a lightweight search framework incorporating graph sparsification and network pruning. More recently, LLM4GNAS (Gao et al. 2025) has explored the use of Large Language Models to enhance the search process. Despite these advancements in automated GNN modeling, existing NAS methods generally assume centralized data availability and are not directly applicable to distributed, privacy-preserving federated scenarios.

DA-DFGAS Algorithm

Problem Definition

This paper defines a graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A, X)$, comprising n nodes with features $X \in \mathbb{R}^{n \times f}$ and adjacency matrix A. In the federated setting with m clients $\{C_1, \ldots, C_m\}$, each client C_i trains a local GNN (parameterized by weights w_i and architecture α_i) on its private subgraph.

We define the following two key attributes for client C_i:

Prediction Distribution $P_i \in \mathbb{R}^C$: Let C denote the number of classes. P_i represents the predictive probability distribution over the classes, where $P_i(c)$ is the probability assigned to class c, satisfying $\sum_{c=1}^{C} P_i(c) = 1$.

Accuracy Correction Factor $\mathrm{ACC}(w_i, \alpha_i) \in (0, 1]$: This scalar quantifies the model performance, defined as the classification accuracy of the model (w_i, α_i) on the validation set of client C_i.

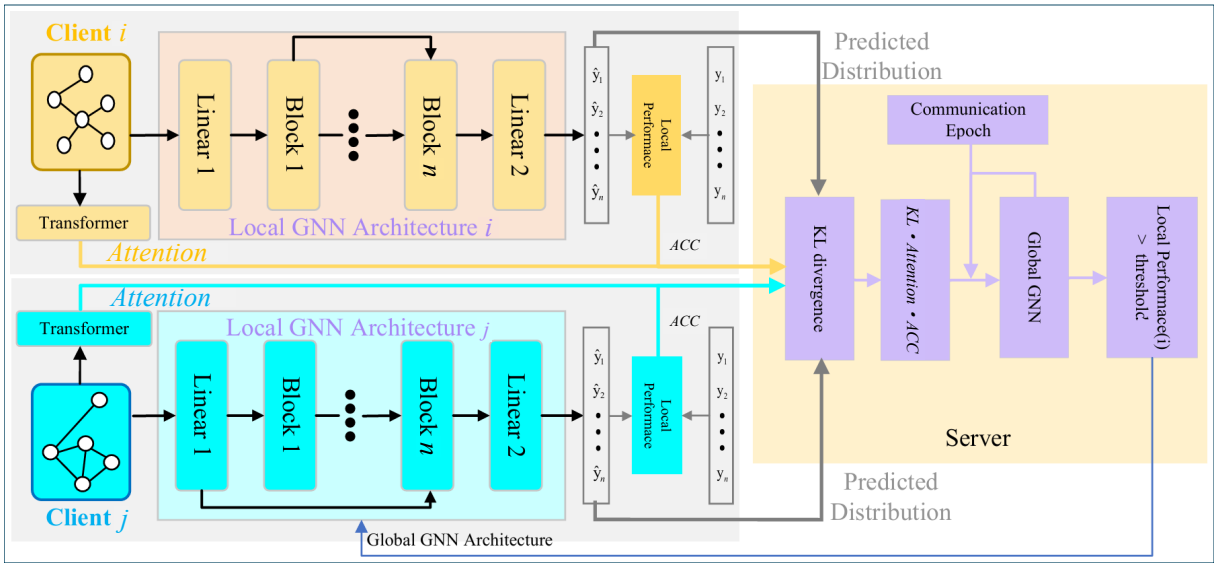

Figure 1. Overall framework of DA-DFGAS. Each client performs bi-level local optimization for architecture and weights, then uploads local model parameters and distribution statistics (e.g., P_i, ACC) to the server. The server executes distribution-aware aggregation to update the global architecture and weights, subsequently broadcasting them along with global distribution references to guide local personalization.

	Operations					
filter	$\mathcal{N}$	$\mathcal{I}$	$\mathcal{F}_s$	$\mathcal{F}_d$		
aggregate	$\mathcal{I}$	$\mathcal{L}_{sum}$	$\mathcal{L}_{max}$	$\mathcal{L}_{mean}$	$\mathcal{N}$	
activate	$\mathcal{I}$	LeakyReLU	ReLU	Tanh	ELU	Sigmoid

Table 1. The operations for the search space of DA-DFGAS

Architecture Design

To strike a balance between model performance and search efficiency, we constrain the search space to three core functional components: Feature Filtering, Neighborhood Aggregation and Activation Functions. The search space is structured as a directed tree topology, encompassing all candidate operators and their potential connections (see Figure 2, left).

We define a comprehensive set of operations, as detailed in Table 1. To enable flexible connectivity search, we introduce two topological operations: the Zero operation $\mathcal{N}$, indicating no connection between nodes, and the Identity operation $\mathcal{I}$, representing a direct skip connection.

Traditional GNNs often lack mechanism to adaptively rescale features based on their importance. To address this, we design two filtering operations: Coarse-grained Scaling $\mathcal{F}_s$ and Fine-grained Scaling $\mathcal{F}_d$. These operations act as gating mechanisms to control information propagation. Let $H^{(l)} \in \mathbb{R}^{n\times d}$ denote the node feature matrix at layer l. The operations are formally defined as:

$$\mathcal{F}_s(H_j^l) = QH_j^l \tag{1}$$

$$\mathcal{F}_d(H_j^l) = Z \odot H_j^l \tag{2}$$

where $\odot$ denotes the Hadamard product. $Q \in \mathbb{R}^{n\times n}$ is a diagonal scaling matrix (node-wise), and $Z \in \mathbb{R}^{n\times d}$ is a

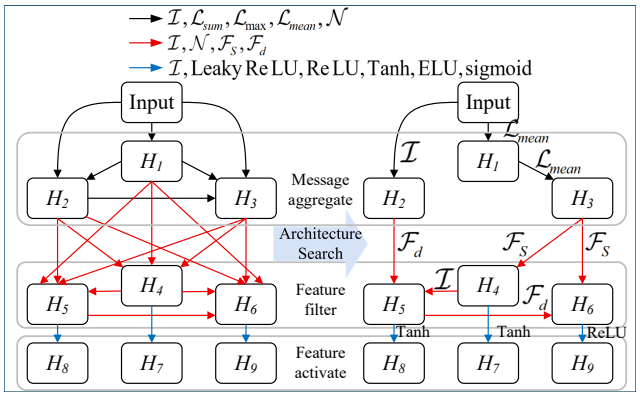

Figure 2. Illustration of the search space and architecture derivation. Nodes H_i represent hidden features and edges represent candidate operations with learnable architectural parameters. Right is a discretized GNN architecture formulated by retaining the optimal paths and operations.

feature-wise scaling matrix. They are computed via a learnable gating network:

$$Q = \mathrm{diag}(W_Q([H_j^l, X^{l-1}])) \tag{3}$$

$$Z = W_Z(W_Q([H_j^l, X^{l-1}])) \tag{4}$$

where $[\cdot]$ denotes the concatenation operation along the feature dimension. $H^{(l-1)}$ represents the output of the previous layer. $W_Q : \mathbb{R}^{2d} \to \mathbb{R}$ and $W_Z : \mathbb{R}^{2d} \to \mathbb{R}^d$ are learnable projection weights (implemented as Multi-Layer Perceptrons). $\sigma(\cdot)$ is the activation function (e.g., Sigmoid) ensuring the scaling factors are within a valid range. Figure 1 is the framework diagram of DA-DFGAS.

Architecture Discretization

Instead of optimizing a discrete search space directly, DA-DFGAS adopts a continuous relaxation strategy. it assigns learnable architecture parameters α to each candidate operation, effectively relaxing the categorical choice into a continuous probability distribution (e.g., via Softmax). During the search phase, both the network weights and the architecture parameters are jointly optimized via gradient descent, allowing the framework to explore the search space in a differentiable manner.

Upon convergence of the search phase, we map the continuous architecture parameters back to a discrete graph structure based on the learned probability distributions. For edge selection, we employ a hierarchical pruning strategy. Intermediate nodes in Layers one and three retain only the single incoming edge with the maximal probability, ensuring a streamlined flow. Intermediate nodes in Layer two follow an adaptive path preservation mechanism. Let p_1 and p_2 denote the highest and second-highest probabilities of incoming edges, respectively. If the probability gap $(p_1 - p_2)$ is below a pre-defined threshold δ, both edges are retained to encourage feature diversity; otherwise, only the top-ranked edge is kept. This process yields a compact, discrete Directed Acyclic Graph. The final node representation X is obtained by concatenating the features from the terminal nodes of all retained paths and projecting them into the latent space.

In the federated setting, the constructed Supernet serves as a unified parameter space. This weight-sharing mechanism ensures that while clients may sample and train personalized subnetworks, they maintain aligned parameter indices and dimensionality, thereby facilitating effective aggregation. The overall process comprises two stages: First, the search stage jointly optimizes network parameters and architectural parameters until a stable probability distribution is achieved. Subsequently, the discretization stage fixes the final architecture and focuses solely on training model parameters under this fixed architecture.

Model Aggregation Strategy

To effectively handle Non-IID data, DA-DFGAS integrates a self-attention-based data quality assessment with a distribution alignment mechanism. For each client i, we project the local feature matrix $X_i \in \mathbb{R}^{|S_i| \times d}$ into a latent space using linear transformations to generate query Q_i, key K_i, and value V_i matrices:

$$Q_i = X_i W_Q, K_i = X_i W_K, V_i = X_i W_V \tag{5}$$

where $W_Q, W_K \in \mathbb{R}^{d \times d_k}$ and $W_V \in \mathbb{R}^{d \times d}$ are learnable projection matrices, with $d_k \ll d$ to reduce computational complexity. We then compute a self-attention matrix to capture feature correlations and compress it into a scalar metric via the Frobenius norm $\| \cdot \|_F$. To comprehensively evaluate client reliability, we combine this feature-based metric with the model's validation performance, defining the local weight $\mathcal{W}_i$ as:

$$\mathcal{W}_i = \|\text{softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}}) V_i\|_F \cdot ACC(w_i, \alpha_i) \tag{6}$$

where $ACC(w_i, \alpha_i)$ is the accuracy correction factor.

Under conditions where the local model is sufficiently trained and possesses probabilistic calibration, its predicted probabilities can approximate the true conditional distribution (Guo et al. 2017). Let $P_i(c)$ denote the average predicted probability of class c over all samples in client i. We estimate the Global Reference Distribution P_{global} as a weighted average of local distributions:

$$P_{\text{global}}(c) = \sum_{i=1}^{m} \frac{|S_i| \mathcal{W}_i}{\sum_{j=1}^{m} |S_j| \mathcal{W}_j} P_i(c) \tag{7}$$

To quantify the statistical heterogeneity, we employ the Kullback-Leibler (KL) divergence between the local distribution P_i and the global reference P_{global}. To ensure numerical stability, we apply ϵ-smoothing to the probabilities. The unbounded KL divergence is then mapped to a bounded similarity score $Sim(\cdot) \in (0, 1]$:

$$KL(P_i \| P_{\text{global}}) = \sum_{c=1}^{C} P_i(c) \log \frac{P_i(c)}{P_{\text{global}}(c)} \tag{8}$$

$$Sim(P_i, P_{\text{global}}) = 1/(1 + \tau \cdot KL(P_i | P_{global})) \tag{9}$$

where τ is a scaling hyperparameter. The final aggregation weight $\overline{\mathcal{W}}_i$ is dynamically adjusted to prioritize clients that are statistically aligned with the global distribution:

$$\overline{\mathcal{W}}_i = \frac{Sim(P_i, P_{\text{global}}) \cdot \mathcal{W}_i}{\sum_{j=1}^{m} Sim(P_j, P_{\text{global}}) \cdot \mathcal{W}_j} \tag{10}$$

In the early stages of NAS, diverse architectural exploration is beneficial. However, as training progresses, the search process must converge to a stable optimal structure. To ensure this stability and prevent oscillation, we introduce a temporal smoothing mechanism for the global architecture update:

$$\alpha^t = \gamma_t \alpha^{t-1} + (1 - \gamma_t) \sum_{i=1}^{m} \overline{\mathcal{W}}_i \alpha_i^{t-1} \tag{11}$$

where α^t represents the global architecture parameters at round t. We employ a time-varying coefficient γ_t that monotonically increases from an initial value γ_0 to 1:

$$\gamma_t = 1 - (1 - \gamma_0) \exp(-t/T) \tag{12}$$

where T controls the rate of saturation. As $t \to \infty$, $\gamma_t \to 1$, causing the global architecture to gradually freeze ($\alpha^t \approx \alpha^{t-1}$), thereby ensuring the convergence of the architecture search phase.

Normalization for System Heterogeneity

To accommodate heterogeneous computing resources and varying model convergence speeds, DA-DFGAS incorporates an adaptive client training strategy. Specifically, during each communication round t, if the validation accuracy of client i falls below a predefined threshold ϵ, the system identifies it as an underperforming client. To assist convergence, these clients are permitted to perform additional local training steps or retain their local model states without resetting

to the global model immediately. While this flexibility improves local adaptability, it inevitably induces heterogeneity in the number of local updates, denoted as τ_i, across different clients.

In a standard federated average, directly aggregating updates from clients with different τ_i introduces a bias, causing the global optimization objective to disproportionately favor clients with larger τ_i rather than those with higher data quality or importance. To eliminate this objective inconsistency, we adopt the FedNova normalization strategy on top of our distribution-aware weights $\overline{\mathcal{W}}_i$. The core idea is to normalize the accumulated local updates by the number of local steps to approximate the true gradient direction. The calibrated global update rule is defined as:

$$w^{t+1} = w^t + \eta \cdot \tau_{\text{eff}} \cdot \sum_{i=1}^{m} \overline{\mathcal{W}}_i \cdot \frac{\Delta_i^t}{\tau_i}, \quad \tau_{\text{eff}} = \sum_{i=1}^{m} \overline{\mathcal{W}}_i \tau_i \quad (13)$$

where $\Delta_i^t = w_i^t - w^t$ represents the accumulated local update, η is the server-side learning rate, and τ_{eff} is the effective total steps weighted by client importance. By incorporating the term Δ_i^t/τ_i, we effectively treat the update as a normalized gradient expectation. This mechanism ensures that the global model convergence is driven by the distribution-aware significance $\overline{\mathcal{W}}_i$ rather than the spurious scale effects caused by heterogeneous local training steps.

Model Optimization

In Non-IID federated scenarios, a single unified objective often fails to balance the trade-off between local adaptability and global statistical consistency. To address this, we formulate a proximal bi-level optimization objective. For each client i, the local optimization aims to minimize the empirical loss while constraining the model deviation to ensure stable convergence. The local objective function is formulated as:

$$F_i(w_i, \alpha_i) = \mathcal{L}_{\text{train},i}(w_i, \alpha_i) + \frac{\mu}{2}\|w_i - w\|^2 \quad (14)$$

where $\mathcal{L}_{\text{train},i}$ denotes the training loss on the local subgraph. The proximal term $\frac{\mu}{2}\|w_i - w\|^2$ serves as a regularization constraint, effectively mitigating local model drift caused by data heterogeneity. Architecture Search phase fixs weights w_i and updates architecture parameters α_i by minimizing the validation loss. Weight Update phase fixs α_i and update w_i via gradient descent on Eq. 14. This decoupled approach ensures that the proximal term specifically stabilizes the weight space, reducing the variance introduced by heterogeneous local updates.

The server aims to minimize a global surrogate objective, defined as the distribution-aware weighted average of local losses:

$$F_{\text{global}}(w, \alpha) = \sum_{i=1}^{m} \overline{\mathcal{W}}_i \, F_i(w, \alpha) \quad (15)$$

where $\overline{\mathcal{W}}_i$ is the aggregation weight derived in Eq. 10. The server aggregates the uploaded local model parameters w_i and architecture parameters α_i using the normalization rules described in Eq. 11–13.

To strictly safeguard user privacy, the Transformer projection parameters $\{Q_i, K_i, V_i\}$ are treated as local personalization modules. They are updated locally to align the local prediction distribution P_i with the global reference distribution P_{global}.

Datasets	Nodes	Classes	Training/Validation/ Test Nodes
Cora	2,708	7	140/500/1,000
CiteSeer	3,327	6	120/500/1,000
DBLP	22,081	4	17,000/2,400/2,400
ACM	10,924	4	9,000/1,100/1,100
SBM_PATTERN	1,664,491	2	1,400,957/123,040/124,358
SBM_CLUSTER	1,401,803	6	1,168,087/116,896/116,820
CIFAR10	7,058,005	10	5,293,503/705,800/705,840
MNIST	4,939,668	10	3,704,750/493,970/493,920

Table 2. Specific details of the six datasets

Experiments

Experimental Settings

Datasets and Baselines. We evaluate DA-DFGAS on eight diverse datasets, including traditional, heterogeneous, large-scale, and image-based graphs (summarized in Table 2). Image datasets (MNIST, CIFAR10) are converted into graph structures via superpixel segmentation. For comparison, we select representative centralized GNNs (e.g., GIN (Xu et al. 2019), GraphSAGE (Hamilton, Ying, and Leskovec 2017), GAT (Meng et al. 2025b), GatedGCN (Yang et al. 2021), MoNet (Monti et al. 2017)) and seven state-of-the-art FedGNN frameworks to validate the superiority of our federated optimization strategy.

Implementation Details. To simulate realistic Non-IID scenarios with cross-client dependencies, we employ Metis clustering (Karypis and Kumar 1998) to partition graphs into $m \in \{10, 20, 50\}$ subgraphs with varying overlap rates of $\{0.1, 0.2, 0.3\}$. Experiments utilize the missing neighbor generator from FedSage+ to synthesize pseudo-neighbors based on feature similarity. In the training phase, each client performs 10 local epochs per communication round. We optimize model weights w using SGD with momentum and architecture parameters α using Adam. A cosine annealing scheduler is applied to adjust the learning rate dynamically.

Comparison with Existing GNN Architectures

Table 3 demonstrates that DA-DFGAS consistently establishes a new state-of-the-art across diverse graph benchmarks, validating the efficacy of our distribution-aware search strategy. On datasets with relatively homogeneous structures (e.g., MNIST, Cora), DA-DFGAS outperforms the strongest baselines by over 1.0%. More notably, on the highly heterogeneous DBLP and large-scale SBM_CLUSTER datasets, our method achieves accuracy of 59.36% and 61.37% respectively, surpassing the competitive FL-BlockQNN and Fed-Pub by substantial margins of 1.5-3.0%. These results confirm that DA-DFGAS possesses superior robustness and generalization capabilities compared to both manually designed GNNs and existing federated

Model	MNIST		Cora		DBLP		SBM_CLUSTER	
	ACC(%)	Params(M)	ACC(%)	Params(M)	ACC(%)	Params(M)	ACC(%)	Params(M)
GCN	90.71 ± 0.22	0.12	77.10 ± 0.46	0.09	53.46 ± 1.35	0.14	53.45 ± 2.03	0.14
GIN	96.49 ± 0.25	0.15	78.22 ± 0.17	0.12	55.62 ± 0.74	0.14	58.38 ± 0.24	0.17
GraphSage	96.31 ± 0.10	0.11	75.55 ± 0.55	0.08	54.22 ± 0.65	0.10	50.45 ± 0.15	0.13
GAT	95.54 ± 0.21	0.18	77.61 ± 0.31	0.11	56.64 ± 0.42	0.13	57.73 ± 0.32	0.20
GatedGCN	97.34 ± 0.14	0.16	80.13 ± 0.44	0.10	55.97 ± 0.51	0.12	60.40 ± 0.42	0.18
MoNet	90.81 ± 0.03	0.14	79.34 ± 0.13	0.09	57.04 ± 0.63	0.11	58.06 ± 0.13	0.16
FedGCN	97.22 ± 0.06	0.14	78.56 ± 0.16	0.13	56.07 ± 0.37	0.19	58.77 ± 0.57	0.17
FL-BlockQNN	96.97 ± 0.26	0.48	78.17 ± 0.26	0.38	56.33 ± 0.64	0.39	59.89 ± 0.34	0.50
FL-GraphNAS	95.80 ± 0.10	0.47	77.94 ± 0.61	0.37	55.97 ± 0.55	0.37	57.33 ± 0.43	0.49
FedNAS	96.62 ± 0.06	0.43	77.64 ± 0.26	0.36	57.11 ± 0.44	0.39	60.12 ± 0.57	0.47
FL-AGNNS	97.24 ± 0.32	0.52	80.23 ± 0.41	0.41	56.97 ± 0.29	0.52	60.52 ± 0.29	0.52
FedNova	96.04 ± 0.41	0.17	78.67 ± 0.39	0.12	56.82 ± 0.47	0.18	58.97 ± 0.52	0.14
FedProx	96.55 ± 0.28	0.18	80.11 ± 0.22	0.13	57.22 ± 0.34	0.18	59.81 ± 0.44	0.16
FED-PUB	96.79 ± 0.06	0.10	80.00 ± 0.11	0.10	58.32 ± 0.79	0.12	59.12 ± 0.57	0.11
FGSSL	97.11 ± 0.32	0.18	79.95 ± 0.32	0.14	58.64 ± 0.59	0.15	59.77 ± 0.29	0.18
FedSSP	96.55 ± 0.32	0.16	80.14 ± 0.32	0.13	57.11 ± 0.29	0.17	60.02 ± 0.29	0.17
BI-FedGNN	96.97 ± 0.32	0.21	79.21 ± 0.54	0.19	56.83 ± 0.88	0.42	59.76 ± 0.29	0.22
DA-DFGAS	**98.15 ± 0.17**	0.44	**81.21 ± 0.17**	0.39	**59.36 ± 0.32**	0.41	**61.37 ± 0.34**	0.46

Table 3. Comparison of model performance on four datasets

Model	SBM_PATTERN			CiteSeer			DBLP		
	ACC(%)	Params(M)	Time(h)	ACC(%)	Params(M)	Time(h)	ACC(%)	Params(M)	Time(h)
FL-BlockQNN	86.23 ± 0.26	0.43	1.8	67.24 ± 0.89	0.39	1.5	57.21 ± 0.78	0.52	1.7
FL-GraphNAS	85.77 ± 0.54	0.47	1.6	67.97 ± 0.53	0.44	1.8	57.39 ± 0.55	0.49	1.6
FedNAS	85.43 ± 0.77	0.32	1.3	66.89 ± 0.81	0.37	1.0	56.47 ± 0.66	0.37	0.9
FL-AGNNS	86.11 ± 0.41	0.48	1.9	67.62 ± 0.59	0.42	1.7	58.28 ± 0.48	0.51	1.6
DA-DFGAS	**87.35 ± 0.17**	0.37	1.2	**68.79 ± 0.34**	0.40	1.3	**59.36 ± 0.32**	0.41	1.4

Table 4. Comparison of four FedGNAS frameworks on five datasets

frameworks, effectively handling complex structural variations in Non-IID settings.

Comparison with Existing FedGNAS

We evaluate efficiency and convergence under a unified search space, as detailed in Table 4 and Figure 3. DA-DFGAS achieves an optimal balance between performance and cost: while incurring a marginal parameter increase (about 5-10%) due to the attention mechanism, it reduces search time by 30-40% compared to RL-based FL-BlockQNN (1.2h and 1.8h on SBM_PATTERN). This efficiency stems from our differentiable strategy, which avoids the expensive population evaluation inherent in evolutionary or RL methods. Furthermore, Figure 3 illustrates that DA-DFGAS maintains a smooth and fast ascending trajectory with minimal fluctuation, confirming that our distribution-aware aggregation effectively mitigates gradient noise and ensures stable convergence compared to the volatile training curves of baselines.

Comparison with Existing FedGNN

To isolate the impact of our aggregation strategy, we benchmark against seven FedGNN frameworks using a fixed GCN backbone (Table 5). DA-DFGAS consistently delivers the highest accuracy across all five datasets. Unlike optimization-centric methods (e.g., FedProx, FedNova) that mitigate drift solely through regularization or gradient normalization, our approach explicitly exploits distributional information to guide aggregation. This allows DA-DFGAS to achieve gains of 1.0-2.5% on complex tasks like SBM_CLUSTER, effectively bridging the gap between local personalization and global statistical consistency where similarity-based methods (e.g., FED-PUB) and standard baselines fall short.

Datasets	ACM	DBLP	MNIST	CIFAR10	SBM-CLUSTER
FedGCN	87.34	56.07	97.22	66.89	58.77
FedNova	88.67	56.44	97.35	67.11	58.96
FedProx	89.11	56.98	97.11	66.55	59.67
FED-PUB	89.65	57.28	97.24	67.79	58.64
FGSSL	90.13	57.86	96.79	67.57	59.44
BI-FedGNN	88.23	56.75	97.34	65.72	59.43
FedSSP	89.22	57.23	96.31	67.11	59.81
DA-DFGAS	90.89	58.61	97.94	68.46	60.56

Table 5. Comparison of six FedGNN frameworks on five datasets

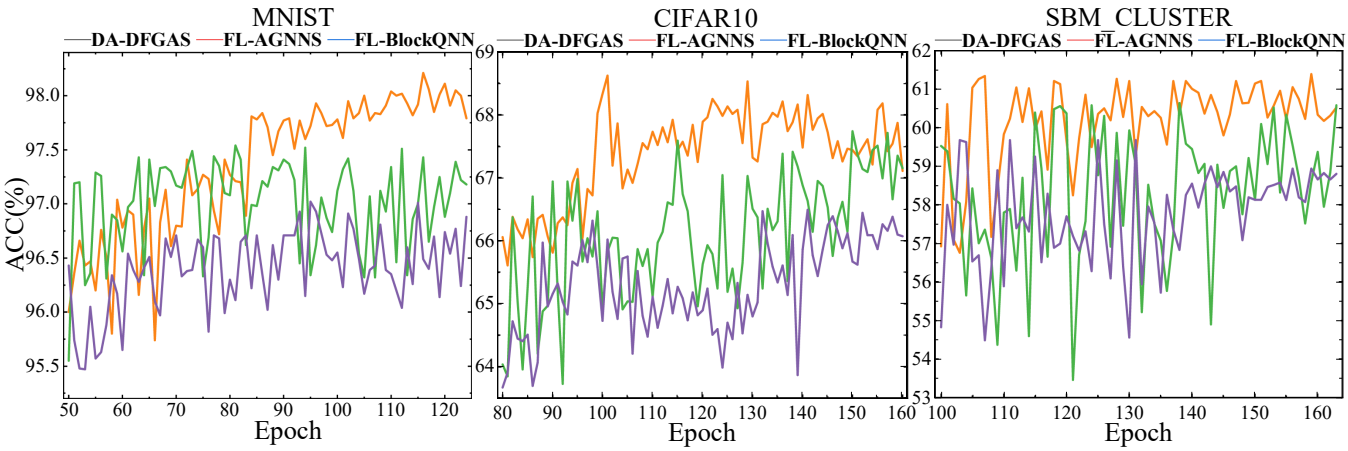

Figure 3. Comparison of DA-DFGAS, FL-AGNNS and FL-BlockQNN on three datasets.

Model	SBM_PATTERN		DBLP	
	ACC(%)	Round	ACC(%)	Round
DA-DFGAS(w/o DA)	86.77 ± 0.54	126	58.67 ± 0.33	164
DA-DFGAS(w/o Sel)	86.89 ± 0.77	144	59.07 ± 0.38	127
DA-DFGAS(w/o Prox)	86.31 ± 0.41	172	58.95 ± 0.44	172
DA-DFGAS(full)	**87.35 ± 0.17**	118	**59.36 ± 0.32**	97

Table 6. Ablation experiments on two datasets

Ablation Experiments

To investigate the contribution of each component in DA-DFGAS, we conduct ablation studies on the SBM_PATTERN and DBLP datasets by removing key modules. The variants are defined as follows:

- DA-DFGAS(w/o DA): Removes the Distribution-Aware aggregation, reverting to standard data-volume based weighting.
- DA-DFGAS(w/o Sel): Removes the System Heterogeneity handling module (i.e., FedNova normalization and performance-based selection).
- DA-DFGAS(w/o Prox): Removes the proximal regularization term from the local objective.

As presented in Table 6, the full DA-DFGAS framework consistently achieves the highest accuracy and fastest convergence. Removing the DA module causes a noticeable accuracy drop (e.g., from 59.36% to 58.67% on DBLP), confirming that ignoring statistical heterogeneity leads to suboptimal global aggregation. Moreover, DA-DFGAS(w/o Sel) variant exhibits unstable performance and increased training time (144 rounds on SBM), demonstrating that correcting for system heterogeneity is crucial for efficient federated optimization. The removal of proximal regularization results in the most severe convergence delay. As shown in the Round column, the required communication rounds for DBLP skyrocket from 97 to 172. This validates that the proximal term effectively suppresses local drift and accelerates consensus.

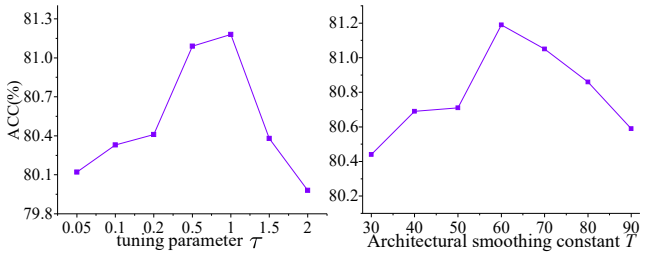

Figure 4. Sensitivity analysis of two critical hyperparameters on the Cora dataset

Parameter Analysis

Figure 4 illustrates the sensitivity of DA-DFGAS to two critical hyperparameters on the Cora dataset: the similarity scaling factor τ and the architectural smoothing constant T. Scaling Factor τ (Eq. 9) controls the sensitivity of the distribution alignment whose performance peaks at $\tau = 1.0$. A value that is too small ($\tau < 0.2$) fails to discriminate between heterogeneous distributions, while a value that is too large ($\tau > 1.5$) over-penalizes differences, making the similarity scores too sparse. Both extremes hinder effective aggregation. Smoothing Constant T (Eq. 12) governs the annealing rate of the global architecture update. The right plot shows an optimal range around $T = 60$. A small T causes the global architecture to freeze too early (before sufficient exploration), whereas an excessively large T delays the stabilization of the architecture, prolonging the search phase.

Conclusion

This paper proposes a unified framework called DA-DFGAS, which aims to address the statistical heterogeneity problem in federated graph learning through personalized neural architecture search and distribution-aware self-attention mechanisms. Moving forward, we plan to broaden the applicability of DA-DFGAS to heterogeneous device environments and investigate its potential in large-scale graph pre-training tasks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62572420, the Shandong Province Key R&D Program under Grant 2024CXGC010801, Yantai Science and Technology Plan Project under Grant 2023ZDCX001, Yantai Smart City Innovation Lab Project, School and Locality Integration Development Project of Yantai City(2022).

References

Baek, J.; Jeong, W.; Jin, J.; Yoon, J.; and Hwang, S. J. 2023. Personalized Subgraph Federated Learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, 1396–1415. PMLR.

Cai, J.; Wang, X.; Li, H.; Zhang, Z.; and Zhu, W. 2024. Multimodal Graph Neural Architecture Search under Distribution Shifts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8227–8235.

Cai, S.; Li, L.; Deng, J.; Zhang, B.; Zha, Z.-J.; Su, L.; and Huang, Q. 2021. Rethinking Graph Neural Architecture Search from Message-Passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6657–6666.

Dong, M.; Li, Y.; Wang, Y.; and Xu, C. 2025. Adversarially Robust Neural Architectures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5): 4183–4197.

Feng, Y.; Lv, Z.; Chen, H.; Gao, S.; An, F.; and Sun, Y. 2024. LRNAS: Differentiable Searching for Adversarially Robust Lightweight Neural Architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3): 5629–5643.

Gao, Y.; Yang, H.; Chen, Y.; Wu, J.; Zhang, P.; and Wang, H. 2025. LLM4GNAS: A Large Language Model Based Toolkit for Graph Neural Architecture Search. *arXiv preprint arXiv:2502.10459*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 1321–1330. PMLR.

Guo, J.; Li, S.; and Zhang, Y. 2023. An Information Theoretic Perspective for Heterogeneous Subgraph Federated Learning. In *Database Systems for Advanced Applications*, 745–760. Springer Nature Switzerland.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30, 1058–1068.

Huang, L.; Huang, X.-D.; Zou, H.; Gao, Y.; Wang, C.-D.; and Yu, P. S. 2024. Knowledge-Reinforced Cross-Domain Recommendation. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7): 12880–12894.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized Cross-Silo Federated Learning on Non-IID Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7865–7873.

Ju, W.; Fang, Z.; Gu, Y.; Liu, Z.; Long, Q.; Qiao, Z.; Qin, Y.; Shen, J.; Sun, F.; Xiao, Z.; et al. 2024. A Comprehensive Survey on Deep Graph Representation Learning. *Neural Networks*, 173: 106207.

Karypis, G.; and Kumar, V. 1998. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*, 20(1): 359–392.

Khaled, A.; Elsir, A. M. T.; Wang, P.; Shen, Y.; and Zhang, Q. 2024. A Graph-Based Approach for Traffic Prediction Using Similarity and Causal Relations Between Nodes. *Knowledge-Based Systems*, 296: 111913.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. *Proceedings of Machine Learning and Systems*, 2: 429–450.

Lu, Z.; Cheng, R.; Jin, Y.; Tan, K. C.; and Deb, K. 2023. Neural Architecture Search as Multiobjective Optimization Benchmarks: Problem Formulation and Performance Assessment. *IEEE Transactions on Evolutionary Computation*, 28(2): 323–337.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, volume 54, 1273–1282. PMLR.

Meng, C.; Tang, M.; Setayesh, M.; and Wong, V. W. S. 2025a. Tackling Resource Allocation for Decentralized Federated Learning: A GNN-Based Approach. *IEEE Transactions on Mobile Computing*, 24(10): 9554–9569.

Meng, T.; Shan, S.; Shao, H.; Shou, Y.; Ai, W.; and Li, K. 2025b. SE-GNN: Seed Expanded-Aware Graph Neural Network with Iterative Optimization for Semi-Supervised Entity Alignment. *IEEE Transactions on Knowledge and Data Engineering*, 37(6): 3700–3713.

Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5117–5124.

Sang, L.; Wang, Y.; Zhang, Y.; Zhang, Y.; and Wu, X. 2025. Intent-Guided Heterogeneous Graph Contrastive Learning for Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 37(4): 1915–1929.

Serhani, M. A.; Tariq, A.; Qayyum, T.; Taleb, I.; Din, I.; and Trabelsi, Z. 2025. Meta-XPFL: An Explainable and Personalized Federated Meta-Learning Framework for Privacy-Aware IoMT. *IEEE Internet of Things Journal*, 12(10): 13790–13805.

Shao, Y.; Li, H.; Gu, X.; Yin, H.; Li, Y.; Miao, X.; Zhang, W.; Cui, B.; and Chen, L. 2024. Distributed Graph Neural Network Training: A Survey. *ACM Computing Surveys*, 56(8): 1–39.

Tan, Z.; Wan, G.; Huang, W.; and Ye, M. 2024. FedSSP: Federated Graph Learning with Spectral Knowledge and Personalized Preference. In *Advances in Neural Information Processing Systems*, volume 37, 34561–34581.

Wang, B.; Li, A.; Pang, M.; Li, H.; and Chen, Y. 2022. GraphFL: A Federated Learning Framework for Semi-Supervised Node Classification on Graphs. In *2022 IEEE International Conference on Data Mining (ICDM)*, 498–507.

Wang, Q.; Liu, W.; Wang, X.; Chen, X.; Chen, G.; and Wu, Q. 2023. GMHANN: A Novel Traffic Flow Prediction Method for Transportation Management Based on Spatial-Temporal Graph Modeling. *IEEE Transactions on Intelligent Transportation Systems*, 25(1): 386–401.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.

Xia, R.; Zhang, C.; Zhang, Y.; Liu, X.; and Yang, B. 2024. A Novel Graph Oversampling Framework for Node Classification in Class-Imbalanced Graphs. *Science China Information Sciences*, 67(6): 1–16.

Xie, B.; Chang, H.; Zhang, Z.; Wang, X.; Wang, D.; Zhang, Z.; Ying, R.; and Zhu, W. 2023. Adversarially Robust Neural Architecture Search for Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8143–8152.

Xie, B.; Chang, H.; Zhang, Z.; Zhang, Z.; Wu, S.; Wang, X.; Meng, Y.; and Zhu, W. 2024. Towards Lightweight Graph Neural Network Search with Curriculum Graph Sparsification. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3563–3573.

Xu, H.; Gao, X.; Liu, J.; Ma, Q.; and Huang, L. 2025. FedACS: An Adaptive Client Selection Framework for Communication-Efficient Federated Graph Learning. *IEEE Transactions on Mobile Computing*, 24(10): 9760–9773.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful Are Graph Neural Networks? In *International Conference on Learning Representations*.

Xu, S.; Xia, H.; Zhang, R.; Liu, P.; and Fu, Y. 2024. FedNor: A Robust Training Framework for Federated Learning Based on Normal Aggregation. *Information Sciences*, 684: 121274.

Yang, B.; Kang, Y.; Zhang, L.; and Li, H. 2021. GGAC: Multi-Relational Image Gated GCN with Attention Convolutional Binary Neural Tree for Identifying Disease with Chest X-Rays. *Pattern Recognition*, 120: 108113.

Yao, Y.; Wang, X.; Qin, Y.; Zhang, Z.; Zhu, W.; and Mei, H. 2024. Data-Augmented Curriculum Graph Neural Architecture Search under Distribution Shifts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16433–16441.

Zhang, C.; Xie, Y.; Chen, T.; Mao, W.; and Yu, B. 2024. Prototype Similarity Distillation for Communication-Efficient Federated Unsupervised Representation Learning. *IEEE Transactions on Knowledge and Data Engineering*, 36(11): 6865–6876.

Zhang, P.; Chen, J.; Che, C.; Zhang, L.; Jin, B.; and Zhu, Y. 2023a. IEA-GNN: Anchor-Aware Graph Neural Network Fused with Information Entropy for Node Classification and Link Prediction. *Information Sciences*, 634: 665–676.

Zhang, T.; Mai, C.; Chang, Y.; Chen, C.; Shu, L.; and Zheng, Z. 2023b. Fedego: Privacy-Preserving Personalized Federated Graph Learning with Ego-Graphs. *ACM Transactions on Knowledge Discovery from Data*, 18(2): 1–27.

Zhang, W.; Lin, Z.; Shen, Y.; Li, Y.; Yang, Z.; and Cui, B. 2022. Deep and Flexible Graph Neural Architecture Search. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, 26362–26374.