Bridging the Domain Gap: Finding reliable Fine-Tuning Paths for GNNs

Jiayi Que[1,†], Lian Shen[1,†], Yue Hong[1], Yuan Lin[3], Juan Liu[2], Xiangrong Liu[1*]
[1] Department of Computer Science and Technology, Xiamen University
[2] Pen-Tung Sah Institute of Micro-Nano Science and Technology, Xiamen University
[3] School of Economics, Ennovation, and Technology, Kristiania University College
quejiayi@stu.xmu.edu.cn, shenlian@stu.xmu.edu.cn, hongy@stu.xmu.edu.cn, yuan.lin@kristiania.no,
cecyliu@xmu.edu.cn, *xrliu@xmu.edu.cn

***Abstract*—The pre-training and fine-tuning paradigms of Graph Neural Networks exhibit poor performance in out-of-distribution generalization. This limitation stems from negative transfer caused by distributional differences between source and target data. A promising solution is to sequentially fine-tune across auxiliary datasets, which act as "knowledge bridges" to mitigate the distribution shift. However, this strategy faces the challenge of finding the reliable auxiliary dataset sequence within a vast combinatorial space. To navigate this search problem, we develop Knowledge Bridge Tuning (KBT), which significantly reduces search complexity by using pre-computed gradients and a logistic regression approximation to efficiently estimate the effectiveness of different transfer paths, making the search for an reliable sequence computationally feasible. Extensive experiments demonstrate that our framework efficiently identifies transfer paths that achieve superior downstream performance, establishing new state-of-the-art results for both in-distribution and out-of-distribution generalization across multiple graph datasets.**

***Index Terms*—negative transfer, OOD generalization, knowledge bridges, sequential fine-tuning, graph classification**

I. Introduction

Graph Neural Networks (GNNs) [1] have achieved remarkable success in modeling graph-structured data. However, their performance is often constrained in scenarios with scarce supervised data. To address this challenge, the pre-training and fine-tuning paradigm [2]–[4] has become a mainstream solution. This paradigm, however, relies on a critical assumption: that the distribution of the pre-training data must align with the distribution of the target data. When a significant distributional shift exists, the model's ability to adapt to the new task is impeded, leading to "negative transfer" [5], [6]. In real-world applications, the prevalent distribution mismatch poses a fundamental challenge, not only hindering the effective transfer of pre-trained knowledge but also strictly limiting the paradigm's performance ceiling.

To mitigate negative transfer, existing research has explored various strategies. For instance, BSS [7] penalizes small-magnitude features to preserve salient information during fine-tuning. Meanwhile, GTOT-Tuning [8] applies feature regularization based on reliable transport to maintain graph structure coherence. More recently, Bridge-Tune [9] introduced an optimization step to enhance representation consistency between the source and target tasks. G-Tuning [10], on the other hand, achieves adaptation by adding the objective of reconstructing the Graphon [11] of the downstream graph, which forces the model to learn and fit the structural characteristics of the downstream graph during fine-tuning. However, these methods are fundamentally constrained by their single-step transfer framework, which exclusively optimizes the mapping between a single source and target domain. When the domain gap is substantial, such a "one-step" transfer path is often insufficient to bridge the vast knowledge gap, leading to diminished performance.

To overcome the limitations of single-step methods, we propose a multi-stage fine-tuning strategy that leverages a sequence of intermediate auxiliary datasets as "knowledge bridges" for sequential adaptation. By constructing a smoother path from the source to the target domain, this approach theoretically aims to enhance the model's adaptability and final performance. However, the success of this strategy requires solutions to the fundamental challenges: how to efficiently search for the reliable sequence in a combinatorially explosive space. To address these challenges, we tackle the search problem by proposing **Knowledge Bridge Tuning (KBT)**, a computationally efficient method to navigate the combinatorial search space. By pre-computing initial task gradients and using logistic regression to approximate the full fine-tuning process, KBT can rapidly estimate the effectiveness of numerous potential sequences, enabling the efficient identification of a highly promising candidate sequence from an otherwise intractable search space.

In summary, our contributions are as follows:

- We propose a novel, multi-stage fine-tuning framework that leverages auxiliary dataset sequences to bridge the distributional gap, effectively mitigating negative transfer.
- We develop Knowledge Bridge Tuning (KBT), a gradient-based method to make the search for an reliable dataset sequence computationally feasible.
- Experiments show our framework achieves state-of-the-art results on both in-distribution and out-of-distribution tasks.

† These authors contributed equally to this work.
* is the corresponding author.

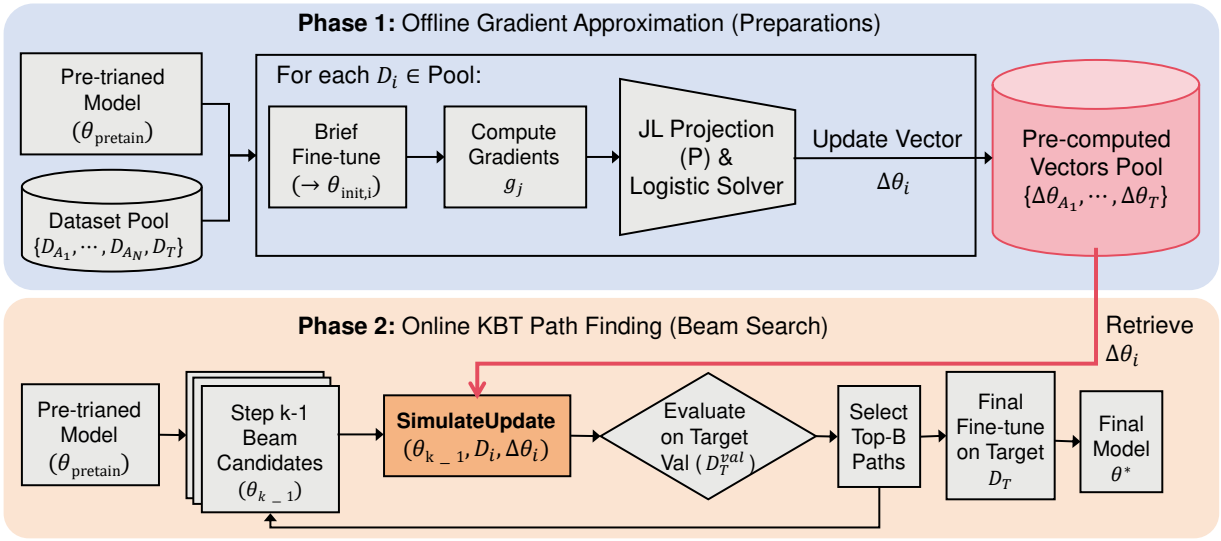

Fig. 1: Overview of the Knowledge Bridge Tuning (KBT) Framework. **(A) Phase 1 (Offline):** This phase efficiently pre-computes an **Update Vector** ($\Delta\theta_i$) for each candidate dataset via gradient projection and a logistic regression solver. **(B) Phase 2 (Online):** This phase navigates the combinatorial search space using a Beam Search strategy. Expanding from the **top-B candidates** (θ_{k-1}) retained from the preceding step, the **SimulateUpdate** mechanism approximates the next model state θ_{new}. This is achieved by aggregating the retrieved vector $\Delta\theta_i$ with a brief fine-tuning update on dataset D_i, thereby circumventing computationally prohibitive full fine-tuning loops.

II. Related Work

Various strategies have been proposed to improve the standard fine-tuning process. Methods like L2_SP [12], BSS [7], and StochNorm [13] focus on preserving pre-trained knowledge and improving model robustness. Other approaches, such as DELTA [14], AUX-TS [15], and Bridge-Tune [9], improve task adaptation through feature selection or auxiliary information. Furthermore, GTOT-Tuning [8] and G-Tuning [10] are specifically designed to preserve structural patterns during graph knowledge transfer. However, these methods rely on a direct, single-step transfer. This approach is fragile under large distribution gaps due to the lack of a gradual adaptation mechanism.

III. Preliminaries and Problem Formulation

Let $\mathcal{G} = (V, E)$ denote a graph. We consider Graph Neural Networks (GNNs), represented as h_θ, parameterized by θ. The expected risk on a data distribution $\mathcal{D}$ is defined as $R_{\mathcal{D}}(h_\theta) = \mathbb{E}_{(g,y)\sim\mathcal{D}}[\mathcal{L}(h_\theta(g), y)]$, where $\mathcal{L}$ is the loss function.

Given a GNN h_{θ_0} pre-trained on a source dataset $\mathcal{D}_S$ and a target dataset $\mathcal{D}_T$, our objective is to identify an effective ordered sequence of K auxiliary datasets, $A^* = (\mathcal{D}_{A_1}, \ldots, \mathcal{D}_{A_K})$, selected from a candidate pool $\mathcal{P}$. This sequence functions as a "knowledge bridge," facilitating the model's adaptation from the source to the target domain through sequential fine-tuning:

$$h_{\theta_i} = \text{FineTune}(h_{\theta_{i-1}}, \mathcal{D}_{A_i}), \quad \text{for } i = 1, \ldots, K. \tag{1}$$

The final model, h_{θ_K}, is subsequently fine-tuned and evaluated on the target dataset $\mathcal{D}_T$. The optimization goal is to determine the sequence A^* that maximizes generalization performance on $\mathcal{D}_T$.

IV. Methodology: Knowledge Bridge Tuning

Identifying the effective auxiliary sequence A^* presents a significant challenge due to the combinatorial nature of the search space, which grows exponentially with the sequence length. A brute-force evaluation of all possible paths is computationally intractable. To address this, we propose Knowledge Bridge Tuning (KBT), a streamlined framework that reformulates sequential fine-tuning as an efficient path-finding problem. As illustrated in Figure 1, KBT decouples the transfer process into two distinct phases: an offline phase for efficient gradient estimation and an online phase for heuristic path search.

A. Phase 1: Offline Effective Vector Computation

The cornerstone of the KBT framework is the efficient estimation of transfer directions. As depicted in Figure 1(A), we pre-compute an Update Vector, $\Delta\theta_i$, for each candidate dataset $\mathcal{D}_i$ in the pool. This vector serves as a computationally inexpensive proxy for the ideal parameter shift $\theta - \theta_{\text{init},i}$ that would typically require a full fine-tuning session. Crucially, this computation is performed once for each dataset, incurring a constant one-time cost regardless of the search depth.

1) Linearization of the Loss Function: Our approximation is grounded in the observation that during fine-tuning, the model parameters θ typically evolve within a local trust region around the initialization $\theta_{\text{init},i}$ [16]. Within this neighborhood, the complex, non-linear loss landscape can be effectively approximated via a first-order Taylor expansion:

$$\mathcal{L}(h_\theta, (x_j, y_j)) \approx \mathcal{L}(h_{\theta_{\text{init},i}}, (x_j, y_j)) + g_j^\top(\theta - \theta_{\text{init},i}), \tag{2}$$

where $g_j = \nabla_\theta \mathcal{L}(h_{\theta_{\text{init},i}}, (x_j, y_j))$ represents the gradient evaluated at the task-specific starting point. This linearization

allows us to transform the expensive non-convex optimization problem into a simplified convex problem.

2) Dimensionality Reduction and Optimization: Directly optimizing in the high-dimensional parameter space (where dimension p can exceed millions) remains computationally prohibitive. To mitigate this, we employ a Johnson-Lindenstrauss (JL) random projection matrix $P \in \mathbb{R}^{p \times d}$ ($d \ll p$) to project the high-dimensional gradients G_i into a lower-dimensional subspace $G'_i = G_i P$:

$$G'_i = G_i P. \tag{3}$$

In this reduced space, we seek a consensus update vector $\Delta\theta'_i \in \mathbb{R}^d$ that minimizes the aggregate linearized loss across the training set. We formulate this as a generalized regression problem. For general classification tasks (including binary, multi-class, and multi-task settings), the objective is to minimize the Cross-Entropy loss applied to the linearized logits:

$$\Delta\theta'_i = \arg\min_{\Delta\theta' \in \mathbb{R}^d} \sum_{j=1}^{|\mathcal{D}_i^{\text{train}}|} \mathcal{L}_{\text{CE}}\left(y_j, \sigma(g_j^\top P \Delta\theta')\right), \tag{4}$$

where y_j is the label vector and σ denotes the activation function (e.g., sigmoid or softmax). Finally, the high-dimensional update vector is reconstructed via $\Delta\theta_i = P\Delta\theta'_i$, providing a precise directional guide for the subsequent search phase.

B. Phase 2: Online Path Finding via Beam Search

Equipped with the pre-computed update vectors, the second phase navigates the combinatorial search space to identify an effective transfer path (Figure 1(B)). We model sequential fine-tuning as a discrete decision process and employ a Beam Search strategy to balance exploration breadth with computational efficiency.

1) The Search Strategy: Instead of exhaustively evaluating all permutations, KBT maintains a beam of the top-B most promising paths at each step k. For every candidate path in the beam, we expand it by appending unused auxiliary datasets from the pool. The critical challenge here is to evaluate the quality of these expanded paths without incurring the prohibitive cost of full fine-tuning for every candidate.

2) The SimulateUpdate Mechanism: To address this challenge, we introduce the SimulateUpdate mechanism (Algorithm 2). This function provides a rapid yet accurate estimate of the model state θ_{new} after adapting to a new dataset $\mathcal{D}_i$. It synergizes two complementary forces:

- Local Adaptation: The model undergoes a brief fine-tuning phase (e.g., $N_{\text{brief}} = 3$ epochs). This step allows the model parameters to adjust to the specific local curvature of the new dataset's loss landscape, capturing fine-grained task information.
- Global Guidance: We inject the pre-computed effective update vector $\Delta\theta_i$ directly into the parameters. This component acts as a global momentum, guiding the optimization along the theoretically derived optimal direction and preventing the model from getting trapped in poor local minima due to the brevity of the fine-tuning.

Formally, the hybrid update rule is defined as:

$$\theta_{\text{new}} \leftarrow \text{FineTune}(\theta_{\text{curr}}, \mathcal{D}_i, N_{\text{brief}}) + \Delta\theta_i. \tag{5}$$

This strategy reduces the verification cost from $O(C_{\text{full}})$ to $O(C_{\text{brief}})$ per step, rendering the search process computationally feasible while preserving high fidelity to the true fine-tuning trajectory. The candidate paths are then scored based on their performance on the target validation set, and the top-B candidates are propagated to the next iteration.

Algorithm 1 Effective Path Finding via KBT Beam Search

Input: Pre-trained model θ_{pretrain}; Auxiliary pool $\mathcal{P} = \{\mathcal{D}_A\}$; Target dataset $\mathcal{D}_T$; Max length K; Beam width B.
Output: Effective auxiliary sequence A^*; parameters θ^*.
// Phase 1: Offline Preparation
For each $\mathcal{D}_i \in \mathcal{P} \cup \{\mathcal{D}_T\}$, pre-compute $\Delta\theta_i$ using Eq. (4).
// Phase 2: Online Search
Initialize Beam $\mathcal{B} = \{([\,], \theta_{\text{pretrain}})\}$.
for $k = 1, \ldots, K$ **do**
  Candidate Pool $\mathcal{C} \leftarrow \emptyset$.
  for each (A_j, θ_j) in $\mathcal{B}$ **do**
    for each $\mathcal{D}_i \in \mathcal{P} \setminus A_j$ **do**
      $A_{\text{new}} \leftarrow A_j \oplus (\mathcal{D}_i)$.
      $\theta_{\text{new}} \leftarrow \texttt{SimulateUpdate}(\theta_j, \mathcal{D}_i, \Delta\theta_i)$.
      Add $(A_{\text{new}}, \theta_{\text{new}})$ to $\mathcal{C}$.
    end for
  end for
  Score candidates in $\mathcal{C}$ on $\mathcal{D}_T^{\text{val}}$.
  $\mathcal{B} \leftarrow$ Top B candidates from $\mathcal{C}$.
end for
return Best (A^*, θ^*) from $\mathcal{B}$.

Algorithm 2 The SimulateUpdate Function

Input: Params θ_{in}; Dataset $\mathcal{D}_k$; Vector $\Delta\theta_k$; Epochs N_{brief}.
Output: Approximated parameters θ_{out}.
$\theta_{\text{local}} \leftarrow \text{FineTune}(\theta_{\text{in}}, \mathcal{D}_k^{\text{train}}, N_{\text{brief}})$.
$\theta_{\text{out}} \leftarrow \theta_{\text{local}} + \Delta\theta_k$.
return θ_{out}.

V. Experiments and Results

In this section, we answer the following questions:

Q1. (Effectiveness) Can **KBT** enhance the performance on downstream target datasets?

Q2. (OOD generalization) Can **KBT** enhance the model's out-of-distribution generalization performance?

Q3. (Efficiency) Can **KBT** efficiently identify reliable paths within a feasible timeframe and scale to longer sequences?

Q4. (Ablation Study) How do different components and hyperparameters of KBT affect its performance?

A. Fine-tuning GNNs In Domains

Setting. To answer Q1, we evaluate the effectiveness of KBT on molecular property prediction tasks through a comparative experiment. For this test, all methods use a GIN model

TABLE I: Mean and standard deviation of ROC-AUC (%) for fine-tuning strategies on the same domains from pre-training datasets. **Bold** denotes the best performance and underline denotes the second best.

Dataset	BBBP	Tox21	ToxCast	Sider	ClinTox	MUV	Bace	Avg.
Supervised	65.84±4.51	74.02±0.89	61.45±0.64	57.29±1.65	58.06±4.42	71.81±2.56	70.13±5.41	65.5
Vanilla-Tune	68.99±3.13	75.39±0.94	63.33±0.74	59.71±1.08	65.92±3.78	75.78±1.73	79.22±1.20	69.8
StochNorm	69.27±1.67	74.97±0.77	62.69±0.69	60.35±0.98	65.53±4.25	76.05±1.61	81.48±2.10	70.0
Feature Map	60.42±0.78	70.58±0.28	61.50±0.24	61.66±0.45	64.05±3.40	78.36±1.11	76.32±1.15	67.6
DELTA	67.79±0.76	75.22±0.54	63.34±0.58	61.43±0.68	72.11±3.06	80.18±1.13	81.83±1.17	71.7
L2_SP	67.70±3.21	73.55±0.82	62.43±0.34	61.08±0.73	68.12±3.77	76.73±0.92	82.21±2.44	70.3
BSS	68.02±2.76	75.01±0.71	63.11±0.49	59.60±0.92	70.75±4.98	77.92±2.01	82.48±2.10	71.0
GTOT-Tune	70.04±1.48	75.20±0.94	62.89±0.46	63.04±0.50	71.77±5.38	79.82±1.78	82.48±2.18	72.2
G-Tuning	72.59±0.32	75.80±0.29	64.25±0.27	61.40±0.71	74.64±4.30	75.84±1.97	84.79±1.39	72.8
KBT	**72.70±0.26**	**76.32±0.72**	**65.92±0.57**	**63.55±0.37**	**77.66±1.71**	**81.64±1.70**	**86.79±0.34**	**74.9**

TABLE II: Mean and standard deviation of Accuracy (%) for fine-tuning strategies on the different domains from pre-training datasets. **Bold** denotes the best performance and underline denotes the second best.

Dataset	IMDB-M	IMDB-B	MUTAG	PROTEINS	ENZYMES	MSRC_21	Avg.
Supervised	36.67±6.67	52.40±7.20	82.89±6.16	63.51±3.60	20.50±4.02	7.45±3.52	43.9
Vanilla-Tune	50.20±2.72	72.10±3.65	83.45±5.71	64.42±4.91	21.33±5.62	7.99±2.39	49.9
StochNorm	49.87±3.11	72.20±2.99	82.40±7.95	64.08±3.61	23.33±4.25	9.08±3.21	50.2
Feature Map	50.87±2.89	72.90±2.70	82.95±7.80	63.06±4.80	22.67±4.78	9.77±4.30	50.4
DELTA	50.67±2.81	71.80±3.99	82.98±6.12	63.96±5.35	22.33±5.01	10.48±3.84	50.4
L2_SP	51.07±2.11	71.90±2.59	82.98±3.91	65.95±4.57	21.50±4.50	10.29±3.91	50.6
BSS	47.35±1.76	73.20±3.25	84.56±5.52	65.58±6.79	23.41±4.79	10.47±3.29	50.8
GTOT-Tune	51.13±2.72	72.30±2.93	87.50±6.94	62.89±3.88	20.67±5.49	8.32±3.92	50.5
G-Tuning	51.80±2.31	74.30±3.29	86.14±5.50	**72.05±3.80**	26.70±4.28	11.01±2.08	53.7
KBT	**51.93±3.20**	**75.20±3.46**	**89.98±5.82**	71.34±3.33	**26.83±4.62**	**12.99±2.95**	**54.7**

pretrained on the ZINC15 dataset via a self-supervised context prediction task as the shared backbone. We fine-tune and evaluate the models on seven graph classification datasets from MoleculeNet: BBBP, Tox21, ToxCast, Sider, ClinTox, MUV, and Bace. The data is split using a scaffold-based 8:1:1 ratio. All experiments are repeated over 10 independent runs with different random seeds. We report the mean ROC-AUC and standard deviation. When applying our KBT method, a "leave-one-out" strategy is employed where one dataset is designated as the target, and the remaining six (from the list above) are treated as the candidate pool of auxiliary datasets.

Results. The results in Table I show that when the pre-training and downstream task data distributions are aligned, KBT significantly enhances model performance. Our method surpasses all baseline fine-tuning strategies, achieving the best performance on 7 out of 7 datasets. Overall, KBT achieves an average ROC-AUC of 74.9%, an improvement of 1.7% over the strongest baseline, G-Tuning. These results validate KBT's effectiveness in leveraging auxiliary data for model transfer.

B. Fine-tuning GNNs Across Domains

Setting. To answer Q2, we assess the model's capability for out-of-distribution (OOD) generalization under significant domain shifts. We utilize the GCC framework [17] as the backbone, which is pre-trained on large-scale academic and social networks via subgraph discrimination to extract generic structural features. We evaluate performance on six diverse downstream datasets: IMDB-M, IMDB-B, MUTAG, PROTEINS, ENZYMES, and MSRC_21. These datasets encompass social, bioinformatics, and computer vision domains, possessing distinct structural patterns and semantic meanings compared to the pre-training source, thereby creating a challenging OOD scenario. We adopt a leave-one-out transfer strategy where, for each designated target dataset, the remaining five datasets form the candidate pool of auxiliary tasks. We report the average accuracy and standard deviation across 10 independent runs with different random seeds to ensure robust evaluation.

Results. As shown in Table II, cross-domain transfer poses significant challenges, often leading to negative transfer in conventional methods. For instance, DELTA underperforms on IMDB-B and MUTAG due to unstable feature correlations, while BSS degrades performance on IMDB-M by 2.9% due to restrictive regularization. In contrast, KBT effectively bridges these gaps through progressive adaptation. By efficiently identifying a reliable adaptation path, KBT mitigates abrupt distribution shifts and achieves state-of-the-art performance, securing the highest average accuracy of 54.7% and top scores on 5 out of 6 datasets. This validates the effectiveness of transforming fine-tuning into a systematic search for reliable intermediate bridges.

C. Efficiency and Scalability Analysis

To answer Q3, we evaluate the computational feasibility of KBT by analyzing the acceleration of single-step adaptation, the overall search efficiency compared to baselines, and the scalability of the method with respect to path length.

1) Single-Step Acceleration: The foundation of KBT's efficiency lies in its gradient-based approximation. By replacing

TABLE III: Efficiency comparison on MUTAG ($N = 5$). KBT finds an effective path ($K = 3$) with the best trade-off between runtime and accuracy, significantly outperforming Brute-Force (intractable) and Multi-task (slow) baselines.

Category	Method	Update Cost	Total Time (min)	Accuracy (%)
Search Baselines	Brute-Force	Full FT	$\sim 1,440$ (24 h)	N/A
	Random Search	Full FT	19.0 (Fixed)	$86.50_{\pm 2.1}$
Multi-task	AUX-TS	Joint Train	~ 118.0	$87.63_{\pm 1.5}$
Ours	**KBT** ($K = 3$)	**Proxy** ($\Delta\theta$)	**19.0**	$\mathbf{89.98_{\pm 1.9}}$

the computationally expensive full fine-tuning with a projected gradient proxy ($d = 1000$), KBT significantly reduces the overhead per adaptation step. Specifically, on the MUTAG dataset, the runtime for a single update decreases from 8.00 minutes (Vanilla Fine-tuning) to 0.73 minutes (KBT Proxy). This represents an $11.0\times$ speedup, confirming that our linearization strategy effectively circumvents the heavy computational burden of iterative back-propagation during the search phase while maintaining comparable predictive accuracy.

2) Search Efficiency Comparison: Building on single-step acceleration, we further assess KBT's ability to navigate the combinatorial search space. We compare KBT against search-based and multi-task baselines on the MUTAG dataset. Consistent with our cross-domain leave-one-out protocol, the auxiliary candidate pool size is $N = 5$. The results are summarized in Table III.

A comprehensive brute-force search would require evaluating 60 permutation paths, taking an estimated 24 hours, which is computationally intractable. In contrast, KBT identifies an effective path in just 19.0 minutes, achieving a speedup of over $75\times$. This dramatic reduction makes effective path-finding computationally feasible. Furthermore, Random Search fails to find effective solutions when constrained to the same time budget as KBT, as it cannot complete even a single full-path evaluation. In comparison, the multi-task baseline AUX-TS requires joint training on all datasets, resulting in a runtime of 118 minutes. This is approximately $6\times$ slower than KBT, primarily due to the slow convergence caused by gradient conflicts among diverse tasks. KBT outperforms all baselines, achieving state-of-the-art accuracy with significantly lower runtime.

3) Scalability and Convergence Analysis: A potential concern in sequential methods is whether the computational cost becomes prohibitive as the sequence length K increases. We addressed this by investigating the trade-off between search depth and performance gain.

As illustrated in Figure 2, KBT exhibits a consistent "rise-and-plateau" pattern across both in-domain and cross-domain tasks. Theoretically, the search time scales linearly with the sequence length K (adding approx. 5 minutes per step for $N = 5$). However, empirical results show that performance saturates at a shallow depth ($K = 3$). Extending the path further ($K > 3$) yields diminishing returns or even degradation, likely due to semantic drift or overfitting. This early convergence property is operationally significant: it effectively bounds the necessary search depth, ensuring that KBT delivers maximal performance gains within a minimal computational budget rather than requiring deep, expensive searches.

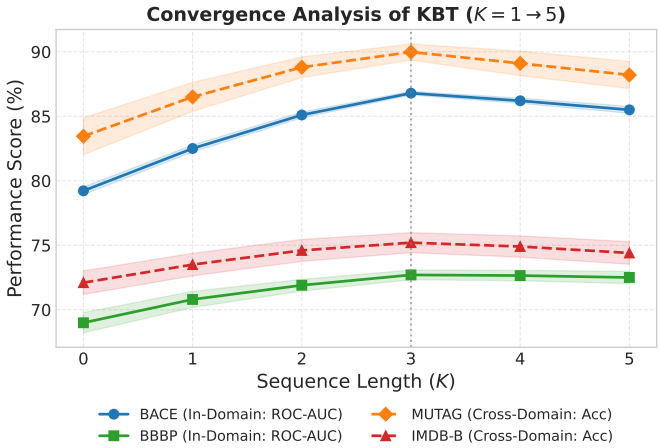

Fig. 2: Convergence analysis of KBT with increasing sequence length K. Across diverse domains (In-Domain: BACE, BBBP; Cross-Domain: MUTAG, IMDB-B), performance consistently saturates at $K = 3$. This demonstrates that KBT achieves early convergence, avoiding the need for deep, expensive searches.

TABLE IV: Ablation study of the update mechanism. We report ROC-AUC (%) for BACE and Accuracy (%) for MUTAG. The hybrid strategy (KBT) yields the best performance with the lowest standard deviation.

Variant	Mechanism	BACE	MUTAG
Vector-Only	$\theta + \Delta\theta_i$	$82.4_{\pm 1.2}$	$85.1_{\pm 2.5}$
Brief-FT Only	FineTune(1 ep)	$83.5_{\pm 1.5}$	$86.8_{\pm 3.1}$
KBT (Ours)	**Hybrid**	$\mathbf{86.1_{\pm 0.6}}$	$\mathbf{89.5_{\pm 1.9}}$

D. Ablation Study and Hyperparameter Sensitivity

To answer Q4, we conduct a rigorous ablation study to validate the contribution of the core components of KBT and analyze the sensitivity of the model to key hyperparameters.

1) Effectiveness of the Hybrid Update Mechanism: The SimulateUpdate mechanism is designed to balance global gradient guidance with local task adaptation. To verify the necessity of this synergistic design, we compare KBT against two degenerate variants: Vector-Only (which removes local fine-tuning) and Brief-FT Only (which removes the pre-computed vector).

As shown in Table IV, the full KBT framework consistently outperforms both variants on the BACE and MUTAG datasets. The Vector-Only approach yields lower accuracy, indicating that a fixed global update vector lacks the flexibility to adapt to the specific curvature of the current loss landscape. Con-

TABLE V: Sensitivity analysis on the BACE dataset (ROC-AUC %). We report Mean ± Std over 10 seeds. The robust default setting ($B = 3, K = 3, N = 3$) is highlighted in bold.

Beam Width (B) *(Fix $K = 3, N = 3$)*		**Seq. Length (K)** *(Fix $B = 3, N = 3$)*		**Brief Epochs (N_{brief})** *(Fix $B = 3, K = 3$)*	
Value	Score	Value	Score	Value	Score
1	$84.2_{\pm 0.9}$	1	$83.5_{\pm 1.1}$	0	$82.4_{\pm 1.2}$
3	$\mathbf{86.1}_{\pm 0.6}$	2	$85.8_{\pm 0.7}$	1	$85.3_{\pm 0.9}$
5	$86.2_{\pm 0.5}$	**3**	$\mathbf{86.1}_{\pm 0.6}$	**3**	$\mathbf{86.1}_{\pm 0.6}$
10	$86.1_{\pm 0.5}$	4	$84.9_{\pm 1.1}$	5	$86.0_{\pm 0.7}$

TABLE VI: Performance with varying projection dimensions (d). $d = 1000$ suffices for high-performance transfer.

Dimension	**ToxCast**	**Sider**	**ENZYMES**
10	60.83	58.04	16.66
100	61.29	58.32	16.83
500	63.06	59.16	18.83
1000	**63.35**	**59.83**	**21.30**
Vanilla-FT	63.33	59.71	21.33

versely, the Brief-FT Only approach exhibits higher variance, suggesting that gradients derived from few-shot fine-tuning are noisy and prone to local optima. The hybrid mechanism successfully integrates the stability of the global vector with the flexibility of local fine-tuning, achieving the best trade-off between performance and stability.

2) Hyperparameter Sensitivity Analysis: We examine the impact of three critical hyperparameters: beam width B, sequence length K, and brief fine-tuning epochs N_{brief}. The results on the BACE dataset are summarized in Table V.

Search Parameters. Performance improves as beam width B increases to 3 and then saturates, indicating that a small beam width efficiently locates effective paths. Similarly, accuracy peaks at sequence length $K = 3$; extending paths further ($K \geq 4$) degrades performance and stability due to error accumulation and potential catastrophic forgetting.

Update Efficiency. The choice of N_{brief} represents a trade-off between adaptation quality and gradient approximation validity. Performance improves as N_{brief} increases to 3, enabling the model to adapt to the local geometry of auxiliary tasks. However, larger values yield no further gain, as excessive fine-tuning moves parameters beyond the trust region where the linearized gradient approximation holds. Consequently, we select $N_{\text{brief}} = 3$ as the optimal setting to balance efficiency and accuracy.

Impact of Projection Dimension (d). Finally, we evaluated our dimensionality reduction strategy (Table VI). Reducing the parameter dimension to $d = 1000$ (from $> 10^6$) yields performance comparable to full-dimensional fine-tuning. This confirms that random projection efficiently preserves essential gradient information for effective transfer while significantly reducing memory overhead.

VI. Conclusion

In this work, we introduced Knowledge Bridge Tuning (KBT), a novel framework that mitigates negative transfer by reframing fine-tuning as a multi-stage path-finding problem. By efficiently searching for an reliable sequence of auxiliary datasets, KBT constructs "knowledge bridges" to smoothly adapt models across distributional shifts. Our experiments show that KBT establishes new state-of-the-art results, particularly in challenging out-of-distribution scenarios. This work provides a search-based paradigm for robust and scalable transfer learning and domain adaptation.

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[2] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "Gpt-gnn: Generative pre-training of graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1857–1867.

[3] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," in *International Conference on Learning Representations (ICLR)*, 2020.

[4] Y. Lu, X. Jiang, Y. Fang, and C. Shi, "Learning to pre-train graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 4276–4284.

[5] Z. Wang, Z. Dai, B. Poczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," in *International Conference on Learning Representations*, 2022.

[7] X. Chen, S. Wang, B. Fu, M. Long, and J. Wang, "Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning," *Advances in neural information processing systems*, vol. 32, 2019.

[8] J. Zhang, X. Xiao, L.-K. Huang, Y. Rong, and Y. Bian, "Fine-tuning graph neural networks via graph topology induced optimal transport," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 3730–3736, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2022/518

[9] R. Huang, J. Xu, X. Jiang, C. Pan, Z. Yang, C. Wang, and Y. Yang, "Measuring task similarity and its implication in fine-tuning graph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, pp. 12 617–12 625, Mar. 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/29156

[10] Y. Sun, Q. Zhu, Y. Yang, C. Wang, T. Fan, J. Zhu, and L. Chen, "Fine-tuning graph neural networks by preserving graph generative patterns," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, 2024, pp. 9053–9061.

[11] L. Lovász, *Large networks and graph limits*. American Mathematical Soc., 2012, vol. 60.

[12] L. Xuhong, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *International conference on machine learning*. PMLR, 2018, pp. 2825–2834.

[13] Z. Kou, K. You, M. Long, and J. Wang, "Stochastic normalization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 304–16 314, 2020.

[14] X. Li, H. Xiong, H. Wang, Y. Rao, L. Liu, and J. Huan, "DELTA: DEEP LEARNING TRANSFER USING FEATURE MAP WITH ATTENTION FOR CONVOLUTIONAL NETWORKS," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rkgbwsAcYm

[15] X. Han, Z. Huang, B. An, and J. Bai, "Adaptive transfer learning on graph neural networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 565–574.

[16] D. Li, A. Sharma, and H. R. Zhang, "Scalable multitask learning using gradient-based estimation of task affinity," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2024.

[17] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1150–1160.