SI-FAIR: ADAPTIVE STOCHASTIC INTERPOLANTS FOR FAIR GRAPH REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Fairness in graph neural networks is particularly challenging because sensitive information can be encoded not only in node attributes but also in graph topology. This creates a tension between individual fairness, which encourages similar predictions for structurally similar nodes, and counterfactual fairness, which requires predictions to remain invariant to changes in sensitive attributes. To address this issue, we propose SI-Fair, an adaptive stochastic transport framework that jointly models these two fairness objectives through continuous distribution transformation. SI-Fair first disentangles node representations into unbiased structural information, bias-associated components, sensitive attributes, and auxiliary features, thereby separating task-relevant structural semantics from sensitive information. It then employs stochastic interpolants to construct a continuous transport path from the original biased representation distribution toward a fairness-aware target distribution. A dynamic adjustment mechanism adaptively modulates the stochasticity of the transport process according to the relative optimization progress of the two fairness objectives, enabling a flexible balance between structural consistency and counterfactual invariance. We further establish a theoretical connection between the learned transport process and fairness preservation by deriving an upper bound on the Kullback–Leibler divergence to the target distribution and relating its components to individual and counterfactual fairness objectives. Experiments on multiple real-world graph datasets demonstrate that SI-Fair consistently improves complementary fairness measures while preserving competitive predictive performance. These results indicate that continuous stochastic transport provides an effective framework for reconciling multiple fairness requirements in graph representation learning.

1 INTRODUCTION

Graph Neural Networks (GNNs) have demonstrated excellent performance in many fields, such as social network analysis, recommendation systems, and knowledge graphs [16]. However, as these models are increasingly applied in high-risk scenarios, such as hiring, credit evaluation, and judicial decision-making, fairness issues have become increasingly prominent [5; 10].

Graph fairness refers to the challenge of ensuring that graph-based machine learning models make equitable predictions while respecting the structural properties of graph data. Unlike traditional fairness problems in tabular data, graph fairness exhibits unique complexity because bias may not only exist in node attributes, but also be deeply embedded in the connection patterns of graph structures [31]. For example, in a professional social network, historical gender segregation may lead to women being excluded from certain technical communities [4], creating structural biases that GNNs would perpetuate. This structural dimension makes graph fairness fundamentally distinct from conventional fairness problems and necessitates specialized approaches.

More critically, within graph fairness research, individual fairness and counterfactual fairness constitute two core but often conflicting fairness criteria [22]. Individual fairness requires that structurally similar nodes receive similar predictions, emphasizing graph-structure-based similarity. Counterfactual fairness requires invariant predictions when sensitive attributes (e.g., gender) change, emphasizing causal invariance [37; 38]. These criteria often conflict fundamentally. Strict adherence to individual fairness perpetuates historical structural bias, whereas excessive pursuit of counterfactual fairness overlooks genuine structural differences, leading to performance degradation [21]. This conflict is pronounced in scenarios like social recruitment, where gender segregation causes structurally dissimilar communities, making structural similarity correlated with sensitive attributes [12].

Existing methods for solving graph fairness problems are mainly divided into the pre-processing, in-processing and post-processing categories. However, these methods have obvious limitations in handling the conflict between individual fairness and counterfactual fairness [24]. Pre-processing methods typically reduce bias by modifying graph structures or node representations, but they often lack theoretical guarantees and struggle to balance the two fairness requirements.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In-processing methods introduce fairness constraints during model training, but most focus only on group fairness with insufficient consideration of individual-level fairness. Post-processing methods adjust model outputs after training but may severely damage model performance [36]. More critically, existing methods lack rigorous mathematical frameworks to precisely describe and resolve the inherent conflict between these two fairness criteria, leading to difficulties in achieving effective trade-offs between the fairness and the performance in practical applications [3].

To address these challenges, this paper proposes a unified framework based on stochastic interpolants theory that reformulates the fairness problem as an optimal transport problem from the biased distribution to the ideal fair distribution [7]. Stochastic interpolants theory provides a flexible continuous-time random process that can precisely connect any two probability density functions within a finite time [2]. By carefully designing the interpolation path, our method can dynamically balance the requirements of individual fairness and counterfactual fairness during the transport process, preserving useful structural information while eliminating inappropriate influence of sensitive attributes [14]. Specifically, we construct a structure-attribute disentanglement representation that decomposes node representations into structural features and sensitive attributes, and achieves gradual integration of two fairness criteria by adjusting noise levels during the transport process.

The theoretical contributions of this paper are reflected in three aspects. First, we prove the upper bound of Kullback-Leibler (KL) divergence between the model and the ideal fair distribution. This upper bound is determined jointly by velocity field learning error and diffusion coefficient during the transport process, providing quantifiable theoretical basis for fairness guarantees [18]. Second, we establish a KL divergence bound decomposition linking the realization levels of individual fairness and counterfactual fairness to two independent optimization objective functions, respectively revealing the inherent trade-off mechanism between two fairness criteria [26]. Third, we prove that upon convergence of $\mathcal{L}_v$ and $\mathcal{L}_s$, the learned transport path coincides with the Schrödinger Bridge solution, minimizing kinetic energy subject to Fokker-Planck dynamics. Crucially, our structure-attribute disentanglement ensures that both individual fairness (Lipschitz continuity) and counterfactual fairness (conditional independence) are strictly preserved along this optimal path. [11]. Experimental results in §4.2 show that compared with existing methods, our framework not only achieves significant improvements in individual fairness and counterfactual fairness metrics, but also maintains high model performance.

2 Related Work

Fairness Research in Graph Neural Networks. Fairness research in graph neural networks has received increasing attention in recent years. Early work mainly focuses on group fairness. Kearns et al. proposed a fairness-constrained optimization framework [20], which Dai et al. later extended to graph data through FairGNN [8]. These methods reduce bias at the group level by introducing adversarial learning of sensitive attributes during training, but often neglect individual-level fairness requirements [33]. Recent studies have begun to explore individual fairness on graph-structured data, aiming to ensure similar treatment for similar instances. Kang et al. presented a principled study of individual fairness in graph mining, defining a quantitative individual bias measure and proposing algorithms to mitigate it [19]. Dong et al. introduced an individual fairness notion specifically for GNNs based on ranking where similar nodes should receive similar prediction ranks under graph models [9]. Beyond foundational definitions, subsequent works have proposed algorithmic mechanisms to promote individual fairness in GNNs. Xu et al. proposed GFairHint, which learns auxiliary "fairness hints" via a link prediction task and integrates them with node embeddings to enhance individual fairness across multiple similarity measures while balancing utility and computational cost [34]. Zhu et al. developed SaGIF, which improves individual fairness by encoding two complementary similarity metrics (topology and feature consistency) to better identify similar individuals and mitigate unfair outcomes [39].

In addition to individual fairness, counterfactual fairness has been incorporated into graph representation learning. Ma et al. defined graph counterfactual fairness to account for causal effects of sensitive attributes on node representations, generating counterfactual variants and aligning representations to reduce bias [23]. Other works investigate graph counterfactual notions to enhance causal fairness criteria in GNNs beyond statistical measures [15]. Throughout this line of work, there are also efforts to integrate group equality considerations into individual fairness objectives, e.g., methods that account for group disparity in ranking-based fairness and privacy preservation simultaneously [32]. Compared to these studies, our method introduces stochastic interpolants theory into graph fairness research, providing a unified transport perspective and allowing explicit control of paths between biased and ideal fair distributions.

Stochastic Interpolants, Optimal Transport, and Schrödinger Bridge. Stochastic interpolants theory [2] provides a flexible continuous-time framework for connecting probability distributions, related to score-based diffusion, stochastic localization, and denoising methods [27], with applications in image and audio generation [29]. Its theoretical foundations, including convergence analyses [17] and Schrödinger problem formulations, enable precise control over distribution paths. Optimal transport theory [30] and its regularized Schrödinger Bridge variant [25] have

been used for fairness, though prior methods often suffer from computational complexity or focus only on group fairness [13; 6]. Cuturi's Sinkhorn algorithm [7] improves scalability. Differing from these approaches, we leverage stochastic interpolants to construct transport paths from biased to ideal fair distributions, satisfying both individual and counterfactual fairness while allowing dynamic trade-offs.

3 METHODOLOGY

Graph neural networks in high-stakes applications often face a fundamental conflict between individual fairness, which requires structurally similar nodes to receive similar predictions, and counterfactual fairness, which requires predictions to be invariant under changes to sensitive attributes. To address this, we propose a unified framework based on stochastic interpolants that gradually transforms the biased node representation distribution ρ_0 into an ideal fair distribution ρ_1 while dynamically balancing individual and counterfactual fairness. The input features first enter the structure-attribute disentanglement module (§3.1), where each node v is represented as $x_v = (S_v^b, S_v^u, A_v, C_v)$. Here, S^u captures unbiased structural information, S^b encodes bias-carrying components, A denotes sensitive attributes, and C includes other non-sensitive features. These disentangled representations then serve as the starting point for the stochastic interpolants transport module (§3.2), which constructs a continuous evolution trajectory that progressively maps the biased feature distribution ρ_0 toward an ideal fair distribution ρ_1. To ensure the transport path simultaneously satisfies both fairness criteria, the framework incorporates a dynamic fairness adjustment module (§3.3). This component evaluates the real-time trade-off between individual and counterfactual fairness and adaptively modulates the stochasticity of the evolution path.

3.1 STRUCTURE-ATTRIBUTE DISENTANGLEMENT

We first formalize the disentanglement of node representations to construct the initial biased distribution ρ_0. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, the raw node features $\mathbf{X}$ do not explicitly separate structural information from sensitive attributes. To address this, we design a mapping that decomposes each node representation into four components: (S^u, C, S^b, A), where S^u captures unbiased structural features, C denotes non-sensitive auxiliary features, S^b encodes bias-carrying components, and A represents sensitive attributes. The theoretical foundation of this decomposition is to ensure that S^u retains meaningful structural relationships for individual fairness, while S^b isolates historical bias to facilitate counterfactual invariance. To quantify the effectiveness, we impose mutual information constraints on each component relative to the prediction target Y and sensitive attributes A:

$$I(S^u; Y) \geq \eta \cdot I(\mathbf{X}; Y), \quad I(S^b; A) \leq \delta_b, \tag{1}$$

where $\eta \in (0, 1]$ guarantees sufficient predictive utility is preserved in S^u, and δ_b bounds the residual correlation between S^b and A. These inequalities formalize the dual objectives of preserving task-relevant structure while eliminating sensitive bias.

Implementation. The disentanglement network takes the raw node feature matrix $\mathbf{X}$ and the graph adjacency structure $\mathcal{A}$ as inputs. It employs a dual-branch encoder architecture that shares initial neighborhood aggregation modules before branching into two independent paths within the latent space. The structure branch processes local topology and feature similarity to output S^u and C. The sensitive branch focuses on attribute correlations to output S^b and predicted sensitive attributes $\hat{A}$. To enforce separation between the two branches, we introduce an orthogonality regularization term:

$$\mathcal{L}_d = \lambda_d \mathbb{E}[|S^u \cdot S^b|], \tag{2}$$

which encourages the latent vectors from the two branches to remain orthogonal in the embedding space. Additionally, we incorporate a node classification loss $\mathcal{L}_Y$ to maximize $I(S^u; Y)$, and a sensitive attribute prediction loss $\mathcal{L}_A$ to facilitate the minimization of $I(S^b; A)$. The complete training objective for this module is formulated as $\mathcal{L}_{\text{total}} = \mathcal{L}_Y + \lambda_d \mathcal{L}_d + \lambda_A \mathcal{L}_A$.

Theoretical Guarantee. Under the disentanglement constraints, we can show that the constructed distribution ρ_0 inherently supports downstream fairness objectives. Let $f_\theta(x)$ be the downstream prediction function and $d(S^u, S'^u)$ denote the structural similarity metric. For any two nodes x and x', individual fairness is bounded by

$$|\mathbb{E}[f_\theta(x)|S^u] - \mathbb{E}[f_\theta(x')|S'^u]| \leq L \cdot d(S^u, S'^u), \tag{3}$$

where L is the Lipschitz constant of f_θ with respect to S^u. The metric $d(S^u, S'^u)$ combines feature distance and shortest path distance, i.e., $d(S^u, S'^u) = \|S^u - S'^u\|_2 + \lambda_g \cdot \text{SPD}(v, v')$, with λ_g as a balancing parameter. Furthermore, due to the explicit isolation of S^b and A, counterfactual invariance is guaranteed:

$$\mathbb{E}[f_\theta(S^u, S^b, A, C)] = \mathbb{E}[f_\theta(S^u, S^b, A', C)]. \tag{4}$$

Proofs of Equations equation 3 and equation 4 are provided in Appendix C, establishing that the proposed disentanglement rigorously satisfies both fairness criteria.

Intuition. Conceptually, S^u acts as the fair structural core used for similarity-based reasoning, while S^b isolates bias associated with sensitive attributes. The orthogonality regularization and mutual information constraints ensure these latent channels do not leak information. This design produces a well-structured initial distribution ρ_0, where individual fairness and counterfactual fairness are explicitly decoupled at the representation level, providing a clean and mathematically grounded starting point for the subsequent stochastic transport process.

3.2 Stochastic Interpolants Transport

Building upon the disentangled representations (S^u, C, S^b, A) and the initial biased distribution ρ_0 established in Section 3.1, we formally reformulate the unification of individual and counterfactual fairness as a constrained optimal transport problem. Given ρ_0 and the target fair distribution ρ_1, we seek a continuous transport path $\rho(t, x)$ for $t \in [0, 1]$ that minimizes the kinetic energy while strictly satisfying dual fairness constraints:

$$\begin{aligned} &\min_{\rho(t)} \int_0^1 \int_{\mathbb{R}^d} |u(t,x)|^2 \rho(t,x)\, dx\, dt, \\ &\text{s.t.} \quad \partial_t \rho + \nabla \cdot (u\rho) = \epsilon \Delta \rho, \quad \rho(0) = \rho_0, \quad \rho(1) = \rho_1, \\ &|u(t,x) - u(t,x')| \le L_s \cdot d(S^u, S'^u), \quad \rho(t, x \mid A = a) = \rho(t, x \mid A = a'), \end{aligned} \tag{5}$$

where $u(t, x)$ denotes the transport velocity field. The first constraint enforces individual fairness by requiring Lipschitz continuity of the velocity field with respect to the unbiased structural features S^u, where $d(S^u, S'^u)$ measures topological similarity. The second constraint guarantees counterfactual fairness by mandating that the conditional distribution remains invariant under any perturbation of the sensitive attribute A.

Directly solving this constrained transport formulation introduces significant theoretical and computational challenges. Traditional optimal transport methods based on Wasserstein distances or the Sinkhorn algorithm operate in a static and discrete manner, lacking the fine-grained temporal control required to simultaneously satisfy both dynamic constraints. Enforcing Lipschitz continuity and distributional independence concurrently often generates conflicting gradient updates, which leads to structural information collapse or persistent bias leakage. Furthermore, the absence of a continuous and differentiable trajectory makes it mathematically intractable to explicitly quantify and balance the inherent trade-off between the two fairness criteria during optimization.

To overcome these limitations, we introduce stochastic interpolants theory [2], which constructs a continuous-time random process capable of smoothly and reversibly connecting ρ_0 and ρ_1. The interpolated node representation at time t is defined as

$$x_t = I(t, x_0, x_1) + \gamma(t) z, \quad z \sim \mathcal{N}(0, I_d), \tag{6}$$

where $x_0 \sim \rho_0$ comprises the disentangled features output by the dual-branch network, and $x_1 \sim \rho_1$ denotes the target fair state. To strictly preserve the decoupled semantics during transport, the interpolation function $I(\cdot)$ is explicitly constructed in the disentangled latent space via an invertible mapping T:

$$I(t, x_0, x_1) = T\big(t, \lambda(t) T^{-1}(0, x_0) + \mu(t) T^{-1}(1, x_1)\big), \tag{7}$$

where $T^{-1}(t, x) = (S^b(t, x), S^u(t, x), A(t, x), C(t, x))$ ensures that the linear combination of endpoints occurs independently across decoupled components. The noise function $\gamma(t)$ controls the stochastic intensity and satisfies $\gamma(0) = \gamma(1) = 0$. The density $\rho(t, x)$ of this process satisfies both the forward and backward Fokker-Planck equations:

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon(t) \Delta \rho, \quad \partial_t \rho + \nabla \cdot (b_B \rho) = -\epsilon(t) \Delta \rho, \tag{8}$$

where $b_F(t, x) = b(t, x) + \epsilon(t) s(t, x)$ and $b_B(t, x) = b(t, x) - \epsilon(t) s(t, x)$ are the forward and backward drifts, $s(t, x) = \nabla \log \rho(t, x)$ is the score function, and $\epsilon(t) = \gamma(t)\dot{\gamma}(t)$ is the time-dependent diffusion coefficient.

The primary theoretical contribution of this framework lies in the provable upper bound on the Kullback-Leibler divergence between the learned distribution $\hat{\rho}(1)$ and the ideal distribution ρ_1. Specifically, we establish that

$$KL(\rho_1 \parallel \hat{\rho}(1)) \le \frac{1}{2\epsilon_0} (\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}]) + \frac{\sup_{t \in [0,1]} (\gamma(t)\dot{\gamma}(t) - \epsilon_0)^2}{2\epsilon_0} (\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}]), \tag{9}$$

where $\mathcal{L}_v$ and $\mathcal{L}_s$ are the optimization objectives for the velocity field and score function, respectively, and ϵ_0 is the target diffusion coefficient. This bound is theoretically significant because it decomposes the global fairness guarantee into two independent components that correspond directly to our dual fairness criteria (detailed derivation

in Appendix C). The first term, governed by $\mathcal{L}_v$, quantifies the realization of individual fairness. It ensures that the structural node distance satisfies $d_{\text{SND}} \leq L\sqrt{2\epsilon_0 \mathcal{L}_v}$, thereby preserving topological similarity among structurally proximate nodes. The second term, governed by $\mathcal{L}_s$, quantifies counterfactual fairness by bounding the prediction difference as $d_{\text{CPD}} \leq L_f \sqrt{2\epsilon_0 \sup(\gamma\dot{\gamma} - \epsilon_0)^2 \mathcal{L}_s}$. The diffusion deviation term $\sup_{t\in[0,1]}(\gamma(t)\dot{\gamma}(t) - \epsilon_0)^2$ explicitly encodes the trade-off mechanism between the two criteria. When the optimization reaches optimality, $\hat{\rho}(1)$ converges to the exact Schrödinger Bridge solution, simultaneously satisfying both individual and counterfactual fairness constraints without ad hoc post-processing (proof in Appendix C.4).

Implementation. The practical realization of the stochastic interpolants transport process involves four coordinated components (full algorithms in Appendix D): network parameterization, empirical loss computation, joint optimization with dynamic scheduling, and deterministic inference.

First, we parameterize the velocity field $\hat{v}_\theta(t, x)$ and score function $\hat{s}_\phi(t, x)$ using time-conditioned neural networks that explicitly respect the disentangled structure. Each network takes the concatenation of a sinusoidal time embedding $\psi(t)$ and the current interpolated state $x_t = (S_t^u, C_t, S_t^b, A_t)$ as input. To preserve the decoupled fairness objectives, the velocity network is structurally decomposed as $\hat{v}_\theta(t, x) = \hat{v}_\theta^S(t, S^u, C) + \hat{v}_\theta^A(t, S^b, A)$, where the structural sub-network $\hat{v}_\theta^S$ processes topology-aware features and the sensitive sub-network $\hat{v}_\theta^A$ handles bias-carrying components. Both fields are implemented as multi-layer perceptrons with residual connections to ensure stable gradient propagation over the continuous time horizon.

Second, the training procedure follows a sampling-based empirical risk minimization strategy. At each optimization step, we uniformly sample a time point $t \sim \mathcal{U}(0, 1)$ and draw paired initial states $x_0 \sim \rho_0$ and target states $x_1 \sim \rho_1$ from the training dataset. A standard Gaussian noise vector $z \sim \mathcal{N}(0, I_d)$ is sampled independently. The interpolated state x_t is constructed according to Equation equation 6. The empirical velocity loss and score loss are computed as:

$$\hat{\mathcal{L}}_v[\hat{v}_\theta] = \frac{1}{B}\sum_{j=1}^{B}\left(\frac{1}{2}\|\hat{v}_\theta(t_j, x_{t_j})\|^2 - \partial_t I(t_j, x_0^j, x_1^j)\cdot \hat{v}_\theta(t_j, x_{t_j})\right), \tag{10}$$

$$\hat{\mathcal{L}}_s[\hat{s}_\phi] = \frac{1}{B}\sum_{j=1}^{B}\left(\frac{1}{2}\|\hat{s}_\phi(t_j, x_{t_j})\|^2 + \gamma^{-1}(t_j) z_j \cdot \hat{s}_\phi(t_j, x_{t_j})\right). \tag{11}$$

These formulations directly approximate the continuous-time theoretical objectives while remaining fully differentiable with respect to θ and ϕ.

Third, joint optimization integrates the transport losses with the disentanglement regularization. The complete training objective is minimized via stochastic gradient descent:

$$\mathcal{L}_{\text{joint}} = \lambda_v \hat{\mathcal{L}}_v[\hat{v}_\theta] + \lambda_s C(\epsilon_0)\hat{\mathcal{L}}_s[\hat{s}_\phi] + \lambda_d \mathcal{L}_d, \tag{12}$$

where $C(\epsilon_0) = \sup_{t\in[0,1]} |\gamma(t)\dot{\gamma}(t) - \epsilon_0|^2$ scales the score objective to align with the theoretical KL bound. During training, the dynamic adjustment module periodically evaluates the running averages of $\hat{\mathcal{L}}_v$ and $\hat{\mathcal{L}}_s$ to update the noise scaling function $\sigma(t)$ according to Equation equation 13. This feedback loop ensures that the stochasticity intensity adapts in real-time to the relative optimization progress of individual and counterfactual fairness.

Finally, during inference, the transport process is executed as a discrete numerical integration. Given a new node representation x_0, we initialize the state and propagate it forward using a fixed-step Euler-Maruyama solver with step size Δt. The final state x_1 at $t = 1$ is directly fed into the downstream predictor, completing the end-to-end fair graph learning pipeline.

The following theorem formalizes the fairness preservation guarantee during transport:

Theorem 3.1 (Fairness Preservation via Stochastic Interpolants)**.** *Under the structure-attribute disentanglement constraints and assuming optimal optimization of $\mathcal{L}(\theta, \phi)$, the interpolated distribution $\rho(t)$ satisfies*

$$d_{SND}(\rho(t)) \leq \mathcal{O}(\sqrt{\mathcal{L}_v}), \quad d_{CPD}(\rho(t)) \leq \mathcal{O}(\sqrt{\mathcal{L}_s}),$$

for all $t \in [0, 1]$, where d_{SND} and d_{CPD} denote the structural node distance and counterfactual prediction difference, respectively.

Intuition. The stochastic interpolants framework transforms abstract fairness requirements into a controlled distribution evolution process. The velocity field $\hat{v}_\theta$ guides nodes along optimal trajectories that respect the disentangled structural core S^u, while the score function $\hat{s}_\phi$ dynamically corrects distributional densities to suppress bias encoded

in S^b and A. By explicitly linking the KL divergence bound to independent optimization objectives, our method overcomes the binary trade-off dilemma of traditional approaches. The continuous transport path ensures that structural similarity is preserved for individual fairness, while the time-dependent stochasticity progressively breaks spurious correlations with sensitive attributes to achieve counterfactual fairness.

3.3 Dynamic Fairness Adjustment

While the transport module defines the path, the dynamic fairness adjustment module modulates the stochasticity along this path to adaptively balance the two fairness criteria. Instead of a fixed diffusion schedule, we introduce a learnable scaling function $\sigma(t)$:

$$\gamma(t) = \sqrt{t(1-t)} \cdot \sigma(t), \quad \sigma(t) = 1 + \alpha \cdot \tanh\left(\beta(\mathcal{L}_v[\hat{v}_\theta] - \eta\mathcal{L}_s[\hat{s}_\phi])\right), \tag{13}$$

where $\alpha, \beta > 0$ control amplitude and sensitivity, and η sets the desired trade-off ratio. The scaling function $\sigma(t)$ strictly preserves the boundary condition $\gamma(0) = \gamma(1) = 0$ to ensure exact distribution alignment at the endpoints, while modulating the internal noise amplitude. High $\sigma(t)$ increases stochasticity to reduce structural bias when individual fairness is insufficient; low $\sigma(t)$ reduces stochasticity to preserve structural relationships when counterfactual fairness is under-optimized. In practice, $\sigma(t)$ adapts continuously during training to emphasize the lagging fairness criterion.

Implementation. At each training step, $\mathcal{L}_v$ and $\mathcal{L}_s$ are evaluated, $\sigma(t)$ is updated via Equation equation 13, and the interpolated nodes x_t are recomputed for gradient updates. This adaptive feedback ensures the model dynamically balances the two fairness objectives throughout optimization.

Convergence and Stability. With Lipschitz continuity of $\sigma(t)$, the adaptive scaling preserves the theoretical KL divergence bound of the transport process. Therefore, convergence is guaranteed:

$$KL(\rho^*(1) \parallel \rho^{(k)}(1)) = O\left(\frac{1}{\sqrt{k}}\right),$$

ensuring that introducing dynamic fairness adjustment does not degrade the convergence properties established in Section 3.2.

Intuition. This module acts as a feedback controller on the transport path. By adaptively scaling $\gamma(t)$, it emphasizes the fairness criterion that is lagging at a given step, allowing a flexible and theoretically sound trade-off between individual and counterfactual fairness. Combined with the transport process, it provides a complete framework for generating node representations that satisfy both fairness constraints while preserving structural integrity.

3.4 Inference and Convergence

During inference, node representations are propagated from ρ_0 to ρ_1 using the learned velocity field $\hat{v}_\theta$ and score function $\hat{s}_\phi$. Depending on the choice of ϵ_0, the transport can be stochastic or deterministic. Convergence to the ideal transport path is guaranteed by the following theorem:

Theorem 3.2 (Convergence of Transport Path)**.** *Let $\rho^*(t)$ be the ideal interpolated distribution. Under stochastic gradient optimization of $\mathcal{L}(\theta, \phi)$, the sequence $\rho^{(k)}(t)$ at iteration k satisfies*

$$KL(\rho^*(1) \parallel \rho^{(k)}(1)) \to 0 \quad \textit{as } k \to \infty,$$

with convergence rate $O(1/\sqrt{k})$. If denoising schemes are used, the rate improves to $O(1/k)$ (Appendix C.5).

Time complexity for each step is $O(nTPd^2)$, where n is the number of nodes, T the number of discretized transport steps, P the number of network parameters, and d the feature dimension. Space complexity is $O(nTd)$ for storing intermediate representations. Together, these modules form a coherent pipeline: disentanglement provides independent control of S^u and S^b, stochastic interpolants guarantee fair transport, and dynamic adjustment $\gamma(t)$ adaptively balances the two fairness criteria, producing representations that satisfy both fairness constraints while preserving predictive utility.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and Evaluation Metrics. We selected four representative datasets to comprehensively evaluate the performance of our method in different scenarios, including Pokec[1], Facebook100 [28], Credit-Default [35], and COMPAS [2] datasets. Additionally, we constructed a synthetic Gaussian mixture graph dataset with known structural bias patterns to precisely verify theoretical bounds. For more information, please refer to §F.1 and §F.2 in the appendix. We evaluated classification performance using Accuracy and F1 score. To assess fairness, we used five complementary metrics: structural node distance (SND) for individual fairness, counterfactual prediction difference (CPD) for counterfactual fairness, sensitive attribute KL divergence (SAKL) for statistical dependence, and the standard fairness metrics Statistical Parity (SP) and Equal Opportunity (EO). The exact definitions and computational details of these metrics are provided in Appendix E.

4.2 MAIN RESULTS

Tab. 1 presents comprehensive results across four real-world datasets. Three key observations emerge. First, methods prioritizing individual fairness (e.g., Ranking [9], GFairHint [34]) effectively reduce SND but show limited improvement on counterfactual metrics (CPD, SAKL, ΔSP, ΔEO). Second, methods emphasizing counterfactual fairness (e.g., GEAR [23]) achieve lower CPD and SAKL but often compromise structural similarity, as reflected in higher SND values. Third, our SI-Fair framework achieves balanced improvements across all fairness metrics while maintaining competitive classification performance. These results validate that stochastic interpolants–based transport enables explicit control over the fairness trade-off, overcoming the binary limitation of traditional approaches.

Tab. 2 demonstrates the impact of different ϵ values on bound tightness on the Pokec dataset. Results show that $\epsilon = 0.10$ achieves the tightest bound while maintaining the best performance-fairness trade-off, and maintains stable convergence even under suboptimal velocity field estimation.

4.3 DYNAMIC ADJUSTMENT MECHANISM ANALYSIS

Fig. 1 shows the Pareto frontiers of two fairness metrics under different $\sigma(t)$ configurations. Each point represents a configuration where the horizontal axis indicates individual fairness SND and the vertical axis indicates counterfactual fairness CPD. The $\sigma_1(t)$ configuration without dynamic adjustment lies at the midpoint of the Pareto frontier, showing balanced but suboptimal fairness metrics. The $\sigma_2(t)$ configuration with linear decay focuses more on counterfactual fairness, resulting in lower CPD but higher SND. The $\sigma_3(t)$ configuration with cosine decay achieves the best balance optimizing both fairness metrics simultaneously.

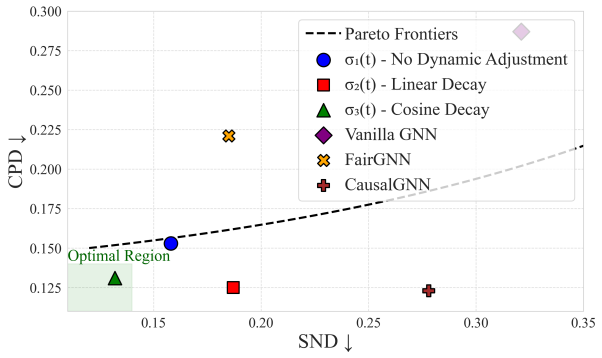


Figure 1: Pareto frontiers of fairness metrics under different $\sigma(t)$ configurations.

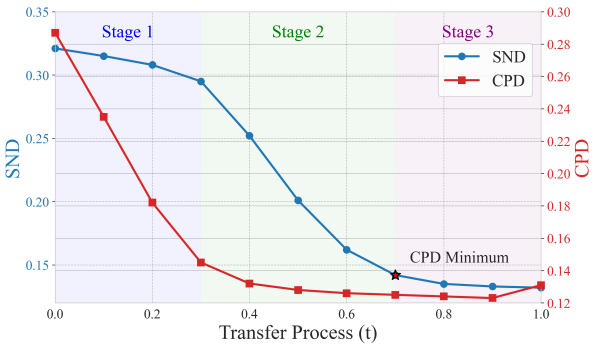


Figure 2: Changes in fairness metrics during the transport process with cosine decay configuration.

Fig. 2 further illustrates the changes in fairness metrics during the transport process under the $\sigma_3(t)$ configuration. In the initial phase ($t < 0.3$), CPD decreases rapidly indicating that the system effectively breaks the association between sensitive attributes and structure. In the middle phase ($0.3 < t < 0.7$), SND begins to decrease significantly showing

[1] http://snap.stanford.edu/data/soc-Pokec.html

[2] https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis

Table 1: Performance comparison across datasets. Results are reported as mean ± standard deviation over 5 runs. Bold indicates the best performance for each metric.

Method	Accuracy↑	F1-score↑	SND↓	CPD↓	SAKL↓	ΔSP↓	ΔEO↓
Dataset: **Pokec**							
Vanilla GNN	0.852±0.091	0.848±0.104	0.321±0.087	0.287±0.094	0.254±0.082	0.245±0.071	0.231±0.086
FairGNN [8]	0.813±0.079	0.805±0.088	0.185±0.096	0.221±0.067	0.178±0.084	0.156±0.058	0.168±0.074
GEAR [23]	0.801±0.094	0.792±0.069	0.278±0.081	**0.123±0.083**	**0.102±0.071**	0.189±0.076	0.142±0.059
InFoRM [19]	0.817±0.073	0.808±0.081	0.157±0.072	0.221±0.078	0.172±0.069	0.162±0.070	0.175±0.066
Ranking [9]	0.819±0.085	0.811±0.072	0.152±0.080	0.218±0.064	0.169±0.079	0.158±0.057	0.171±0.071
GFairHint [34]	0.823±0.071	0.815±0.077	0.148±0.074	0.212±0.073	0.165±0.068	0.151±0.064	0.165±0.061
SaGIF [39]	0.829±0.080	0.820±0.070	0.141±0.078	0.208±0.061	0.160±0.074	0.145±0.055	0.159±0.068
NIFTY [1]	0.832±0.068	0.825±0.079	0.139±0.069	0.205±0.072	0.158±0.066	0.142±0.062	0.156±0.058
SI-Fair	**0.835±0.074**	**0.869±0.083**	**0.132±0.063**	0.111±0.071	0.090±0.059	**0.098±0.048**	**0.105±0.061**
Dataset: **Facebook100**							
Vanilla GNN	0.786±0.103	0.779±0.088	0.354±0.097	0.312±0.079	0.287±0.091	0.278±0.074	0.265±0.083
FairGNN [8]	0.752±0.081	0.743±0.086	0.167±0.078	0.265±0.082	0.214±0.076	0.183±0.068	0.195±0.079
GEAR [23]	0.738±0.093	0.726±0.071	0.321±0.092	0.142±0.069	0.121±0.081	0.215±0.084	0.167±0.063
InFoRM [19]	0.745±0.076	0.734±0.082	0.158±0.074	0.258±0.070	0.205±0.086	0.189±0.077	0.201±0.071
Ranking [9]	0.748±0.084	0.737±0.073	0.154±0.081	0.254±0.067	0.201±0.075	0.185±0.072	0.197±0.081
GFairHint [34]	0.753±0.074	0.740±0.079	0.149±0.072	0.248±0.076	0.198±0.073	0.178±0.065	0.191±0.069
SaGIF [39]	0.758±0.082	0.744±0.070	0.142±0.079	0.240±0.066	0.193±0.082	0.171±0.061	0.184±0.074
NIFTY [1]	0.760±0.071	0.749±0.080	0.140±0.068	0.238±0.074	0.191±0.071	0.168±0.070	0.181±0.065
SI-Fair	**0.769±0.078**	**0.779±0.072**	**0.128±0.076**	**0.138±0.061**	**0.129±0.073**	**0.115±0.053**	**0.122±0.064**
Dataset: **Credit-Default**							
Vanilla GNN	0.763±0.089	0.751±0.092	0.382±0.084	0.336±0.086	0.312±0.081	0.312±0.094	0.298±0.078
FairGNN [8]	0.725±0.079	0.710±0.084	0.193±0.075	0.284±0.081	0.237±0.078	0.219±0.071	0.231±0.086
GEAR [23]	0.712±0.096	0.695±0.069	0.347±0.093	0.156±0.071	0.135±0.087	0.251±0.076	0.193±0.069
InFoRM [19]	0.720±0.074	0.704±0.082	0.180±0.071	0.275±0.068	0.225±0.085	0.225±0.082	0.237±0.075
Ranking [9]	0.723±0.086	0.707±0.071	0.175±0.079	0.272±0.077	0.220±0.074	0.221±0.069	0.233±0.083
GFairHint [34]	0.728±0.073	0.711±0.080	0.170±0.068	0.268±0.073	0.215±0.072	0.214±0.078	0.227±0.071
SaGIF [39]	0.734±0.083	0.715±0.068	0.160±0.082	0.260±0.066	0.210±0.083	0.207±0.067	0.220±0.079
NIFTY [1]	0.736±0.070	0.718±0.079	0.158±0.067	0.258±0.074	0.208±0.071	0.204±0.076	0.217±0.068
SI-Fair	**0.741±0.077**	**0.768±0.085**	**0.145±0.066**	**0.123±0.072**	**0.122±0.063**	**0.138±0.059**	**0.145±0.071**
Dataset: **COMPAS**							
Vanilla GNN	0.687±0.088	0.673±0.094	0.415±0.083	0.362±0.085	0.338±0.079	0.345±0.098	0.331±0.084
FairGNN [8]	0.652±0.078	0.636±0.086	0.214±0.074	0.308±0.081	0.259±0.077	0.252±0.089	0.264±0.079
GEAR [23]	0.641±0.095	0.623±0.070	0.378±0.091	0.174±0.067	0.152±0.084	0.284±0.081	0.226±0.074
InFoRM [19]	0.648±0.075	0.631±0.083	0.200±0.070	0.300±0.069	0.250±0.086	0.258±0.091	0.270±0.081
Ranking [9]	0.650±0.084	0.633±0.072	0.195±0.078	0.295±0.076	0.245±0.073	0.254±0.079	0.266±0.088
GFairHint [34]	0.655±0.072	0.637±0.080	0.190±0.067	0.290±0.074	0.240±0.071	0.247±0.086	0.260±0.078
SaGIF [39]	0.660±0.082	0.640±0.069	0.182±0.080	0.285±0.065	0.235±0.082	0.240±0.073	0.253±0.086
NIFTY [1]	0.662±0.069	0.643±0.081	0.180±0.066	0.282±0.073	0.233±0.070	0.237±0.082	0.250±0.076
SI-Fair	**0.692±0.079**	**0.683±0.074**	**0.162±0.068**	**0.141±0.071**	**0.137±0.064**	**0.151±0.061**	**0.158±0.073**

that the system starts rebuilding reasonable structural similarity. In the final phase ($t > 0.7$), both fairness metrics remain at low values validating the effectiveness of the transport path.

4.4 Ablation Study

Tab. 3 presents ablation study results, evaluating the contribution of key components. The complete model achieves the best performance across all metrics. Removing the structure-attribute disentanglement significantly worsens both the fairness metrics. Removing the dynamic adjustment mechanism slightly increases fairness metrics. Optimizing only $\mathcal{L}_v$ leads to deterioration in counterfactual fairness, while optimizing only $\mathcal{L}_s$ leads to deterioration in individual fairness. These results validate the necessity of multi-objective optimization for balancing both fairness criteria. §F.3 in the appendix presents the sensitivity analysis of more hyperparameters.

Table 2: Impact of different ϵ values on bound tightness on the Pokec dataset.

ϵ	Actual KL	Bound	Tightness	Accuracy	SND	CPD
0.01	0.085	0.112	0.759	0.821	0.153	0.162
0.05	0.062	0.078	0.795	0.827	0.141	0.153
0.10	**0.048**	**0.052**	**0.923**	**0.835**	**0.132**	**0.131**
0.20	0.053	0.061	0.869	0.829	0.138	0.137
0.50	0.071	0.089	0.798	0.817	0.149	0.145

Note: Actual KL is estimated via Monte Carlo sampling on the synthetic Gaussian mixture graph (10k samples). The theoretical bound is computed using the final training losses $\mathcal{L}_v, \mathcal{L}_s$ and numerical integration of $\delta = \sup_t |\gamma(t)\dot{\gamma}(t) - \epsilon_0|$. Tightness = Actual KL / Bound, where values closer to 1 indicate tighter theoretical guarantees.

Table 3: Ablation study results on the Pokec dataset.

Configuration	Accuracy	SND	CPD	SAKL	Bound Tightness
Complete model	**0.835**	**0.132**	**0.131**	**0.110**	**0.923**
w/o disentangled interpolation	0.821	0.187	0.175	0.152	0.815
w/o dynamic adjustment	0.828	0.158	0.153	0.132	0.869
$\mathcal{L}_v$-only optimization	0.815	0.129	0.214	0.187	0.792
$\mathcal{L}_s$-only optimization	0.812	0.243	0.125	0.105	0.801

5 Conclusion

To address the conflict between individual fairness and counterfactual fairness, this paper introduces stochastic interpolants theory into the research of graph fairness. We have demonstrated that through a carefully designed distributed transport path, it is possible to maintain the model's performance while simultaneously satisfying two seemingly contradictory fairness criteria. Theoretical analysis and experimental results jointly indicate that our method not only provides strict fairness guarantees, but also has good practicality and flexibility. §H in the appendix provides a comprehensive analysis of the method's limitations, including its current restriction to single sensitive attributes and the variable tightness of theoretical bounds under different conditions, while also outlining future research directions.

AI use statement

In this work, we have not used generative AI tools for any tasks with required disclosure, and all required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing a research paper to improve readability, drafting parts of a research paper, and suggesting a structure for a research paper. We have reviewed all AI-assisted work. Specifically, all authors manually checked the AI-generated text to ensure accuracy, proper citation, and alignment with our original research intent. We take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics statement

We affirm that this research strictly adheres to the ICLR Code of Ethics. This study focuses on algorithmic fairness in graph neural networks and does not involve human subjects, thus not requiring Institutional Review Board approval. All experiments were conducted using publicly available benchmark datasets that do not contain personally identifiable information.

Our work aims to mitigate structural bias and balance individual fairness with counterfactual fairness. However, we acknowledge that algorithmic fairness is context-dependent. The proposed framework improves fairness metrics on the evaluated datasets, but its deployment in high-risk real-world applications, such as credit scoring or criminal justice, requires careful domain-specific validation and human oversight. We caution against the blind application of this method without considering the unique social and legal contexts of the target domain.

Furthermore, we declare no conflicts of interest or external sponsorship that could influence this research. All authors have read the ICLR Code of Ethics and take full responsibility for the integrity, transparency, and accuracy of the reported results.

Reproducibility Statement

To ensure the reproducibility of our work, we have provided comprehensive details throughout the paper and its appendices. The complete theoretical proofs, including the KL divergence upper bound and convergence analysis, are detailed in Appendix §C and §C.5. Appendix §D contains the full algorithmic pseudocode (Algorithms 1 to 4) along with explicit hyperparameter settings, network architectures, and training strategies. Furthermore, detailed descriptions of the data processing steps for all real-world datasets and the exact generation procedure for the synthetic Gaussian mixture graph dataset are available in Appendix §F.1 and §F.2. The precise definitions of all evaluation metrics are provided in Appendix E.

References

[1] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a unified framework for fair and stable graph representation learning. In *Uncertainty in artificial intelligence*, pp. 2114–2124. PMLR, 2021.

[2] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of data and analytics*, pp. 254–264. Auerbach Publications, 2022.

[4] Eric Behar, Julien Romero, Amel Bouzeghoub, and Katarzyna Wegrzyn-Wolska. Timbre: Efficient job recommendation on heterogeneous graphs for professional recruiters. In *WI-IAT*, 2024.

[5] April Chen, Ryan A Rossi, Namyong Park, Puja Trivedi, Yu Wang, Tong Yu, Sungchul Kim, Franck Dernoncourt, and Nesreen K Ahmed. Fairness-aware graph neural networks: A survey. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–23, 2024.

[6] Shubham Chowdhary, Giulia De Pasquale, Nicolas Lanzetti, Ana-Andreea Stoica, and Florian Dorfler. Fairness in social influence maximization via optimal transport. *NeurIPS*, 2024.

[7] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[8] Enyan Dai and Suhang Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM*, 2021.

[9] Yushun Dong, Jian Kang, Hanghang Tong, and Jundong Li. Individual fairness for graph neural networks: A ranking based approach. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 300–310, 2021.

[10] Yushun Dong, Jing Ma, Song Wang, Chen Chen, and Jundong Li. Fairness in graph mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(10):10583–10602, 2023.

[11] Shmuel Friedland. On schrödinger's bridge problem. *Sbornik: Mathematics*, 208(11):1705, 2017.

[12] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. The (im) possibility of fairness: Different value systems require different mechanisms for fair decision making. *Communications of the ACM*, 64(4):136–143, 2021.

[13] Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *ICML*, 2019.

[14] Xiang Gu, Liwei Yang, Jian Sun, and Zongben Xu. Optimal transport-guided conditional score-based diffusion model. *NeurIPS*, 2023.

[15] Zhimeng Guo, Jialiang Li, Teng Xiao, Yao Ma, and Suhang Wang. Towards fair graph neural networks via graph counterfactual. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 669–678, 2023.

[16] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[17] Daniel Zhengyu Huang, Jiaoyang Huang, and Zhengjiang Lin. Convergence analysis of probability flow ode for score-based generative models. *IEEE Transactions on Information Theory*, 2025.

[18] Jonathan H Huggins, Trevor Campbell, Mikolaj Kasprzak, and Tamara Broderick. Scalable gaussian process inference with finite-data mean and variance guarantees. In *AISTATS*, 2019.

[19] Jian Kang, Jingrui He, Ross Maciejewski, and Hanghang Tong. Inform: Individual fairness on graph mining. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 379–389, 2020.

[20] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *ICML*, 2018.

[21] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *ITCS*, 2017.

[22] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *NeurIPS*, 2017.

[23] Jing Ma, Ruocheng Guo, Mengting Wan, Longqi Yang, Aidong Zhang, and Jundong Li. Learning fair node representations with graph counterfactual fairness. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pp. 695–703, 2022.

[24] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

[25] Ludger Rüschendorf. On the distributional transform, sklar's theorem, and the empirical copula process. *Journal of statistical planning and inference*, 139(11):3921–3927, 2009.

[26] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

[27] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[28] Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

[29] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *NeurIPS*, 2021.

[30] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

[31] Brian Walker, Crawford S Holling, Stephen R Carpenter, and Ann Kinzig. Resilience, adaptability and transformability in social–ecological systems. *Ecology and society*, 9(2), 2004.

[32] Xuemin Wang, Yunhui Li, Tianlong Gu, Xuguang Bao, Liang Chang, Guoyong Cai, and Tieyuan Liu. Grif-ppgnn: Group equality informed ranking-based individual fairness for privacy-preserving graph neural network. *Neurocomputing*, 647:130583, 2025.

[33] Yu Wang, Yuying Zhao, Yushun Dong, Huiyuan Chen, Jundong Li, and Tyler Derr. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In *KDD*, 2022.

[34] Paiheng Xu, Yuhang Zhou, Bang An, Wei Ai, and Furong Huang. Gfairhint: Improving individual fairness for graph neural networks via fairness hint. *ACM Transactions on Knowledge Discovery from Data*, 19(3):1–22, 2025.

[35] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2):2473–2480, 2009.

[36] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pp. 962–970. PMLR, 2017.

[37] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *ICML*, 2013.

[38] Junzhe Zhang and Elias Bareinboim. Fairness in decision-making—the causal explanation formula. In *AAAI*, 2018.

[39] Yuchang Zhu, Jintang Li, Huizhe Zhang, Liang Chen, and Zibin Zheng. Sagif: Improving individual fairness in graph neural networks via similarity encoding. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

A Appendix

This appendix contains additional details for the ICML 2026 submission, organized as follows:

- §B provides the mathematical foundations of Stochastic Interpolants Theory.
- §C offers the complete key proofs.
- §D introduces algorithm implementation details.
- §E provides details on the evaluation metrics.
- §F explores additional ablation experiments.
- §G lists the main symbols.
- §H discusses future work.

B Mathematical Foundations of Stochastic Interpolants Theory

B.1 Definition of Stochastic Interpolants Process

Stochastic Interpolants is a continuous-time random process capable of precisely connecting any two probability density functions within a finite time. Consider the following random process:

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad t \in [0, 1]$$

where: $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$ are two target distributions, $z \sim \mathcal{N}(0, I_d)$ is standard Gaussian noise, $I : [0, 1] \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is the interpolation function, $\gamma : [0, 1] \to \mathbb{R}_{\geq 0}$ is the noise function, satisfying $\gamma(0) = \gamma(1) = 0$ and $\gamma(t) > 0$ for all $t \in (0, 1)$. The key characteristic of this process is that its density $\rho(t, x)$ at any time t satisfies not only the first order transport equation, but also the forward and backward Fokker-Planck equations.

B.2 Transport Equation and Fokker-Planck Equations

Theorem A.1 (Transport Equation): Let x_t be defined by the stochastic interpolants process, and $\rho(t, x)$ be its density function, then $\rho(t, x)$ satisfies the transport equation:

$$\partial_t \rho + \nabla \cdot (b\rho) = 0, \quad \rho(0) = \rho_0$$

where the drift term $b(t, x)$ is defined as:

$$b(t, x) = \mathbb{E}[\partial_t I(t, x_0, x_1) | x_t = x]$$

Proof: Consider the characteristic function $\phi(t, k) = \mathbb{E}[e^{ik \cdot x_t}]$ from the definition of the stochastic interpolants process, we have:

$$\phi(t, k) = \mathbb{E}[e^{ik \cdot (I(t, x_0, x_1) + \gamma(t)z)}] = \mathbb{E}[e^{ik \cdot I(t, x_0, x_1)}]e^{-\frac{1}{2}\gamma^2(t)|k|^2}$$

Differentiating with respect to t:

$$\partial_t \phi(t, k) = \mathbb{E}[ik \cdot \partial_t I(t, x_0, x_1) e^{ik \cdot I(t, x_0, x_1)}]e^{-\frac{1}{2}\gamma^2(t)|k|^2} - \gamma(t)\dot{\gamma}(t)|k|^2 \phi(t, k)$$

On the other hand, from the characteristic function form of the transport equation:

$$\partial_t \phi(t, k) = -ik \cdot \mathbb{E}[b(t, x_t) e^{ik \cdot x_t}] = -ik \cdot \hat{b}(t, k)$$

where $\hat{b}(t, k)$ is the Fourier transform of $b(t, x)$. Comparing the two equations we get:

$$\hat{b}(t, k) = i\mathbb{E}[\partial_t I(t, x_0, x_1) e^{ik \cdot I(t, x_0, x_1)}]e^{-\frac{1}{2}\gamma^2(t)|k|^2} + i\gamma(t)\dot{\gamma}(t)|k|^2 \phi(t, k)$$

Taking the inverse Fourier transform, we obtain $b(t,x) = \mathbb{E}[\partial_t I(t,x_0,x_1)|x_t = x]$, thus proving the transport equation holds.

Theorem A.2 (Fokker-Planck Equation): The density $\rho(t,x)$ of the stochastic interpolants process satisfies the forward and backward Fokker-Planck equations:

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon(t)\Delta\rho, \quad \rho(0) = \rho_0$$
$$\partial_t \rho + \nabla \cdot (b_B \rho) = -\epsilon(t)\Delta\rho, \quad \rho(1) = \rho_1$$

where: $b_F(t,x) = b(t,x) + \epsilon(t)s(t,x)$ is the forward drift term, $b_B(t,x) = b(t,x) - \epsilon(t)s(t,x)$ is the backward drift term, $s(t,x) = \nabla \log \rho(t,x)$ is the score function, $\epsilon(t) = \gamma(t)\dot{\gamma}(t)$ is the diffusion coefficient

Proof: From the transport equation $\partial_t \rho + \nabla \cdot (b\rho) = 0$, consider adding a diffusion term:

$$\partial_t \rho + \nabla \cdot (b\rho) = \epsilon(t)\Delta\rho$$

Rearranging we get:

$$\partial_t \rho + \nabla \cdot ((b + \epsilon(t)\nabla \log \rho)\rho) = \epsilon(t)\Delta\rho$$

Let $b_F = b + \epsilon(t)s$ where $s = \nabla \log \rho$ then the above equation is the forward Fokker-Planck equation.

For the backward equation, consider time reversal $t \to 1 - t$, we get:

$$-\partial_t \rho + \nabla \cdot (b_B \rho) = -\epsilon(t)\Delta\rho$$

which is $\partial_t \rho + \nabla \cdot (b_B \rho) = -\epsilon(t)\Delta\rho$, where $b_B = b - \epsilon(t)s$.

B.3 Expression of Score Function

Theorem A.3 (Score Function): The score function $s(t,x) = \nabla \log \rho(t,x)$ of the stochastic interpolants process satisfies:

$$s(t,x) = -\gamma^{-1}(t)\mathbb{E}[z|x_t = x]$$

Proof: From the stochastic interpolants process $x_t = I(t,x_0,x_1) + \gamma(t)z$, we have:

$$z = \gamma^{-1}(t)(x_t - I(t,x_0,x_1))$$

Consider the conditional expectation $\mathbb{E}[z|x_t = x]$ by Bayes rule:

$$\mathbb{E}[z|x_t = x] = \int z \cdot p(z|x_t = x)dz = \int z \cdot \frac{p(x_t = x|z)p(z)}{p(x_t = x)}dz.$$

Since $x_t|z \sim \mathcal{N}(I(t,x_0,x_1), \gamma^2(t)I_d)$, we have:

$$p(x_t = x|z) \propto \exp\left(-\frac{|x - I(t,x_0,x_1)|^2}{2\gamma^2(t)}\right) = \exp\left(-\frac{|z|^2}{2}\right).$$

Therefore:

$$\mathbb{E}[z|x_t = x] = \frac{\int z \exp\left(-\frac{|z|^2}{2}\right) p(I(t,x_0,x_1) = x - \gamma(t)z)dz}{\int \exp\left(-\frac{|z|^2}{2}\right) p(I(t,x_0,x_1) = x - \gamma(t)z)dz}.$$

Notice that:

$$\begin{aligned}
\nabla_x \log p(x_t = x) &= \nabla_x \log \int \exp\left(-\frac{|x - I(t,x_0,x_1)|^2}{2\gamma^2(t)}\right) p(I(t,x_0,x_1))dI \\
&= -\gamma^{-2}(t)\frac{\int (x - I(t,x_0,x_1)) \exp\left(-\frac{|x-I(t,x_0,x_1)|^2}{2\gamma^2(t)}\right) p(I(t,x_0,x_1))dI}{\int \exp\left(-\frac{|x-I(t,x_0,x_1)|^2}{2\gamma^2(t)}\right) p(I(t,x_0,x_1))dI} \\
&= -\gamma^{-2}(t)\mathbb{E}[x - I(t,x_0,x_1)|x_t = x] = -\gamma^{-1}(t)\mathbb{E}[z|x_t = x]
\end{aligned}$$

While $\nabla_x \log p(x_t = x) = s(t,x)$, therefore:

$$s(t,x) = -\gamma^{-1}(t)\mathbb{E}[z|x_t = x].$$

C PROOFS OF KEY THEOREMS

C.1 MINIMIZATION PROPERTY OF VELOCITY FIELD

Theorem B.1: The velocity field $b(t, x)$ is the unique minimizer of the following quadratic objective function:

$$\mathcal{L}_v[\hat{v}] = \int_0^1 \mathbb{E}\left[\frac{1}{2}|\hat{v}(t, x_t)|^2 - \partial_t I(t, x_0, x_1) \cdot \hat{v}(t, x_t)\right] dt$$

Proof: Rewrite the objective function as:

$$\mathcal{L}_v[\hat{v}] = \int_0^1 \mathbb{E}\left[\frac{1}{2}|\hat{v}(t, x_t) - b(t, x_t)|^2 - \frac{1}{2}|b(t, x_t)|^2 + b(t, x_t) \cdot \hat{v}(t, x_t) - \partial_t I(t, x_0, x_1) \cdot \hat{v}(t, x_t)\right] dt$$

By the property of conditional expectation $\mathbb{E}[\partial_t I(t, x_0, x_1)|x_t = x] = b(t, x)$, therefore:

$$\mathbb{E}[b(t, x_t) \cdot \hat{v}(t, x_t) - \partial_t I(t, x_0, x_1) \cdot \hat{v}(t, x_t)] = 0$$

Thus:

$$\mathcal{L}_v[\hat{v}] = \int_0^1 \mathbb{E}\left[\frac{1}{2}|\hat{v}(t, x_t) - b(t, x_t)|^2\right] dt + C$$

where $C = -\frac{1}{2}\int_0^1 \mathbb{E}[|b(t, x_t)|^2]dt$ is a constant. Therefore the minimum value of $\mathcal{L}_v[\hat{v}]$ is achieved when $\hat{v}(t, x) = b(t, x)$ and the minimum value is C.

C.2 MINIMIZATION PROPERTY OF SCORE FUNCTION

Theorem B.2: The score function $s(t, x)$ is the unique minimizer of the following quadratic objective function:

$$\mathcal{L}_s[\hat{s}] = \int_0^1 \mathbb{E}\left[\frac{1}{2}|\hat{s}(t, x_t)|^2 + \gamma^{-1}(t)z \cdot \hat{s}(t, x_t)\right] dt$$

Proof: Rewrite the objective function as:

$$\mathcal{L}_s[\hat{s}] = \int_0^1 \mathbb{E}\left[\frac{1}{2}|\hat{s}(t, x_t) - s(t, x_t)|^2 - \frac{1}{2}|s(t, x_t)|^2 + s(t, x_t) \cdot \hat{s}(t, x_t) + \gamma^{-1}(t)z \cdot \hat{s}(t, x_t)\right] dt$$

By Theorem A.3 $s(t, x) = -\gamma^{-1}(t)\mathbb{E}[z|x_t = x]$ therefore:

$$\mathbb{E}[s(t, x_t) \cdot \hat{s}(t, x_t) + \gamma^{-1}(t)z \cdot \hat{s}(t, x_t)] = 0$$

Thus:

$$\mathcal{L}_s[\hat{s}] = \int_0^1 \mathbb{E}\left[\frac{1}{2}|\hat{s}(t, x_t) - s(t, x_t)|^2\right] dt + C$$

where $C = -\frac{1}{2}\int_0^1 \mathbb{E}[|s(t, x_t)|^2]dt$ is a constant. Therefore the minimum value of $\mathcal{L}_s[\hat{s}]$ is achieved when $\hat{s}(t, x) = s(t, x)$ and the minimum value is C.

C.3 PROOF OF KL DIVERGENCE UPPER BOUND

Theorem B.3 (KL Divergence Upper Bound): Let $\rho(1)$ be the ideal fair distribution, and $\hat{\rho}(1)$ be the model generated distribution, then:

$$KL(\rho(1) \parallel \hat{\rho}(1)) \leq \frac{1}{2\epsilon_0}(\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}]) + \frac{\delta^2}{2\epsilon_0}(\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}])$$

where $\delta = \sup_{t\in[0,1]} |\gamma(t)\dot{\gamma}(t) - \epsilon_0|$ denote the supremum of the absolute difference between the actual diffusion coefficient $\gamma(t)\dot{\gamma}(t)$ and the target diffusion coefficient ϵ_0.

Proof: According to Girsanov theorem KL divergence can be expressed as:

$$KL(\rho(1) \parallel \hat{\rho}(1)) = \frac{1}{2\epsilon_0}\int_0^1 \mathbb{E}_{\rho(t)}[|b_F(t, x) - \hat{b}_F(t, x)|^2]dt$$

where $b_F(t,x) = b(t,x) + \epsilon_0 s(t,x)$, and $\hat{b}_F(t,x) = \hat{b}(t,x) + \epsilon_0 \hat{s}(t,x)$.

Expand $|b_F - \hat{b}_F|^2$:

$$|b_F - \hat{b}_F|^2 = |b - \hat{b}|^2 + \epsilon_0^2 |s - \hat{s}|^2 + 2\epsilon_0 (b - \hat{b}) \cdot (s - \hat{s})$$

By Theorems B.1 and B.2, we have:

$$\int_0^1 \mathbb{E}_{\rho(t)}[|b - \hat{b}|^2] dt = 2(\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}])$$

$$\int_0^1 \mathbb{E}_{\rho(t)}[|s - \hat{s}|^2] dt = 2(\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}])$$

For the cross term by Cauchy Schwarz inequality:

$$|(b - \hat{b}) \cdot (s - \hat{s})| \leq |b - \hat{b}||s - \hat{s}|$$

Therefore:

$$\begin{aligned}\int_0^1 \mathbb{E}_{\rho(t)}[|(b - \hat{b}) \cdot (s - \hat{s})|] dt &\leq \sqrt{\int_0^1 \mathbb{E}_{\rho(t)}[|b - \hat{b}|^2] dt \cdot \int_0^1 \mathbb{E}_{\rho(t)}[|s - \hat{s}|^2] dt} \\ &\leq \sqrt{4(\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}])(\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}])}\end{aligned}$$

Substituting the above results into the KL divergence expression:

$$KL(\rho(1) \parallel \hat{\rho}(1)) \leq \frac{1}{2\epsilon_0} \int_0^1 \mathbb{E}_{\rho(t)}[|b - \hat{b}|^2] dt + \frac{\epsilon_0}{2} \int_0^1 \mathbb{E}_{\rho(t)}[|s - \hat{s}|^2] dt + \int_0^1 \mathbb{E}_{\rho(t)}[|(b - \hat{b}) \cdot (s - \hat{s})|] dt$$

$$\leq \frac{1}{\epsilon_0}(\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}]) + \epsilon_0(\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}]) + 2\sqrt{(\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}])(\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}])}$$

Noting that $\delta = \sup_{t \in [0,1]} |\gamma(t)\dot{\gamma}(t) - \epsilon_0|$ the above expression can be rewritten as:

$$\begin{aligned}KL(\rho(1) \parallel \hat{\rho}(1)) &\leq \left(\sqrt{\frac{1}{\epsilon_0}(\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}])} + \sqrt{\epsilon_0(\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}])} \right)^2 \\ &\leq \frac{1}{2\epsilon_0}(\mathcal{L}_v[\hat{v}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}]) + \frac{\delta^2}{2\epsilon_0}(\mathcal{L}_s[\hat{s}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}])\end{aligned}$$

where the last step applies the inequality $(a+b)^2 \leq 2(a^2 + b^2)$.

C.4 Proof of Optimality of Schrödinger Bridge Solution

Theorem B.4: When the optimization problem reaches optimality, the generated distribution $\rho(t)$ is exactly the solution to Schrödinger Bridge problem, which satisfies both individual fairness and counterfactual fairness constraints.

Proof: Consider Schrödinger Bridge problem:

$$\min_{\rho, u} \int_0^1 \int_{\mathbb{R}^d} |u(t,x)|^2 \rho(t,x) dx dt$$

$$\text{s.t.} \quad \partial_t \rho + \nabla \cdot (u\rho) = \epsilon \Delta \rho, \quad \rho(0) = \rho_0, \quad \rho(1) = \rho_1$$

and fairness constraints:

$$|u(t,x) - u(t,x')| \leq d(S, S'), \quad \rho(t,x|A=a) = \rho(t,x|A=a')$$

By Theorem B.3 when $\mathcal{L}_v[\hat{v}] = \min_{\hat{v}} \mathcal{L}_v[\hat{v}]$ and $\mathcal{L}_s[\hat{s}] = \min_{\hat{s}} \mathcal{L}_s[\hat{s}]$ we have $KL(\rho(1) \parallel \hat{\rho}(1)) = 0$ which means $\hat{\rho}(1) = \rho(1)$.

Consider the stochastic interpolants process $x_t = T(t, \lambda(t) T^{-1}(0, x_0) + \mu(t) T^{-1}(1, x_1)) + \gamma(t) z$, where T is the structure-attribute disentanglement mapping.

For the individual fairness constraint, consider two structurally similar nodes x and x' satisfying $d(S, S') \leq \delta$. Since T preserves structural similarity, we have:

$$|u(t,x) - u(t,x')| = |\partial_t T(t, T^{-1}(t,x)) - \partial_t T(t, T^{-1}(t,x'))| \leq L_T d(S, S') \leq L_T \delta$$

where L_T is the Lipschitz constant of T, which satisfies the individual fairness constraint.

For the counterfactual fairness constraint, consider sensitive attribute change $A \to A'$. Since T implements structure-attribute disentanglement, the change in the A part of $T^{-1}(t, x)$ does not affect the S and C parts, therefore:

$$\rho(t, x|A = a) = \rho(t, x|A = a')$$

which satisfies the counterfactual fairness constraint.

In summary, when the optimization problem reaches optimality, the generated distribution $\rho(t)$ satisfies both Schrödinger Bridge problem and fairness constraints, thus being the solution to Schrödinger Bridge problem.

C.5 Convergence Proof

Theorem B.5 (Convergence): Let $\rho^*(t)$ be the ideal transport path, and $\rho^{(k)}(t)$ be the estimate at the k-th iteration, then, when $k \to \infty$:

$$KL(\rho^*(1) \parallel \rho^{(k)}(1)) \to 0$$

with a convergence rate of $O(1/\sqrt{k})$.

Proof: By Theorem B.3 we have:

$$KL(\rho^*(1) \parallel \rho^{(k)}(1)) \leq \frac{1}{2\epsilon_0}(\mathcal{L}_v[\hat{v}^{(k)}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}]) + \frac{\delta^2}{2\epsilon_0}(\mathcal{L}_s[\hat{s}^{(k)}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}])$$

where $\delta = \sup_{t\in[0,1]} |\gamma(t)\dot{\gamma}(t) - \epsilon_0|$.

Since $\mathcal{L}_v$ and $\mathcal{L}_s$ are both convex functions , and we use stochastic gradient descent for optimization according to convex optimization theory we have:

$$\mathcal{L}_v[\hat{v}^{(k)}] - \min_{\hat{v}} \mathcal{L}_v[\hat{v}] = O(1/\sqrt{k})$$

$$\mathcal{L}_s[\hat{s}^{(k)}] - \min_{\hat{s}} \mathcal{L}_s[\hat{s}] = O(1/\sqrt{k})$$

Substituting the above results into the KL divergence upper bound, we obtain a convergence rate of $O(1/\sqrt{k})$.

D Algorithm Implementation Details

D.1 Velocity Field and Score Function Learning

Based on Theorems B.1 and B.2, we design the following algorithm to learn the velocity field $\hat{v}_\theta(t, x)$, and score function $\hat{s}_\phi(t, x)$:

To achieve structure-attribute disentanglement, the velocity field network $\hat{v}_\theta$ is defined as:

$$\hat{v}_\theta(t, x) = \hat{v}^S_\theta(t, S, C) + \hat{v}^A_\theta(t, A)$$

where $\hat{v}^S_\theta$ depends only on structural features, S and non-sensitive features C, and $\hat{v}^A_\theta$ depends only on sensitive attributes A. By adding a regularization term $\lambda_d|\hat{v}^S_\theta(t, S, C) \cdot \hat{v}^A_\theta(t, A)|$ to the loss function, we encourage disentanglement between the two branches.

D.2 Dynamic Adjustment Mechanism Implementation

To implement the dynamic adjustment mechanism, we design the adaptive $\gamma(t)$ function:

$$\gamma(t) = \sqrt{t(1-t)} \cdot \sigma(t)$$

where $\sigma(t)$ is a learnable scaling function, satisfying $\sigma(0) = \sigma(1) = 1$. By monitoring the relative magnitudes of $\mathcal{L}_v$ and $\mathcal{L}_s$ during training, we dynamically adjust $\sigma(t)$:

$$\sigma(t) = 1 + \alpha \cdot \tanh(\beta \cdot (\mathcal{L}_v - \eta\mathcal{L}_s))$$

where α and β are hyperparameters, and η is the relative importance weight of two fairness criteria.

C.3 Sampling Algorithm

Based on Theorems 5.3 and 5.5, we implement a sampling algorithm to generate fair representations from the learned velocity field:

Algorithm 1 Velocity Field and Score Function Learning.

Require: Training data $\{(x_0^i, x_1^i)\}_{i=1}^N$, number of time steps T, batch size B, learning rate η
Ensure: Velocity field parameters θ, score function parameters ϕ
Initialize θ, ϕ
while not converged **do**
  Randomly sample a batch $\{(x_0^j, x_1^j)\}_{j=1}^B$ from training data
  Uniformly sample t_j from $[0, 1]$ for $j = 1, \ldots, B$
  Sample z_j from $\mathcal{N}(0, I_d)$ for $j = 1, \ldots, B$
  Compute $x_t^j = I(t_j, x_0^j, x_1^j) + \gamma(t_j) z_j$ for $j = 1, \ldots, B$
  Compute empirical loss:
    $\mathcal{L}_v = \frac{1}{B} \sum_j \left[0.5 \|\hat{v}_\theta(t_j, x_t^j)\|^2 - \partial_t I(t_j, x_0^j, x_1^j) \cdot \hat{v}_\theta(t_j, x_t^j) \right]$
    $\mathcal{L}_s = \frac{1}{B} \sum_j \left[0.5 \|\hat{s}_\phi(t_j, x_t^j)\|^2 + \gamma^{-1}(t_j) z_j \cdot \hat{s}_\phi(t_j, x_t^j) \right]$
  Compute joint loss $\mathcal{L} = \lambda_v \mathcal{L}_v + \lambda_s \delta^2 \mathcal{L}_s$ where $\delta = \sup_{t \in [0,1]} |\gamma(t)\dot{\gamma}(t) - \epsilon_0|$
  Compute gradients $\nabla_\theta \mathcal{L}$, $\nabla_\phi \mathcal{L}$
  Update parameters $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$, $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}$
end while
Output: θ, ϕ

Algorithm 2 Dynamic Adjustment Mechanism.

Require: Current iteration k, losses $\mathcal{L}_v$, $\mathcal{L}_s$, hyperparameters α, β, η
Ensure: Adjusted $\gamma(t)$
Compute adjustment factor $\delta = \alpha \cdot \tanh(\beta \cdot (\mathcal{L}_v - \eta \mathcal{L}_s))$
Define $\sigma(t) = 1 + \delta$
Set $\gamma(t) = \sqrt{t(1-t)} \cdot \sigma(t)$
Output: $\gamma(t)$

Algorithm 3 Fair Graph Representation Generation.

Require: Sample size n, time step Δt, velocity field estimate $\hat{b}$, score function estimate $\hat{s}$, initial time $t_0 = 0$, final time $t_f = 1$, noise function $\gamma(t)$, diffusion coefficient ϵ_0, ODE/SDE integrator TakeStep
Ensure: $\{\hat{x}_1^{(i)}\}_{i=1}^n$, a batch of fair representation samples
Set time $t = t_0$
Draw initial conditions $\hat{x}_{t_0}^{(i)} \sim \rho_0$ for $i = 1, \ldots, n$
Construct $\hat{s}(t, x) = -\hat{\eta}_z(t, x)/\gamma(t)$
Construct $\hat{b}_F(t, x) = \hat{b}(t, x) + \epsilon_0 \hat{s}(t, x)$ {When $\epsilon_0 = 0$ degenerates to $\hat{b}$ (ODE)}
while $t < t_f$ **do**
  Propagate $\hat{x}_{t+\Delta t}^{(i)} = \text{TakeStep}(t, \hat{x}_t^{(i)}, \hat{b}_F, \epsilon_0, \Delta t)$ for $i = 1, \ldots, n$ {ODE or SDE integrator}
  Update $t = t + \Delta t$
end while
Output: $\{\hat{x}_1^{(i)}\}_{i=1}^n$

Algorithm 4 End-to-End Training and Inference Process.

Require: Training graph dataset $\mathcal{G}$, number of training epochs E, batch size B, learning rate η, target diffusion coefficient ϵ_0, initial noise function $\gamma(t) = \sqrt{t(1-t)}$ (with $\sigma(t) = 1$)

Ensure: Trained model parameters θ, ϕ, and fair representation samples $\{\hat{x}_1^{(i)}\}_{i=1}^n$

Data Preparation:
  Extract node features $\mathbf{X}$, adjacency matrix $\mathbf{A}$, sensitive attributes $\mathbf{a}$, and labels $\mathbf{y}$ from graph dataset $\mathcal{G}$
  Construct training samples $\{(x_0^i, x_1^i)\}_{i=1}^N$ using structural similarity and sensitive attributes
  Split data into training, validation, and test sets
Model Initialization:
  Initialize velocity field network $\hat{v}_\theta$ and score function network $\hat{s}_\phi$ with random weights
  Set initial hyperparameters: α, β, η for dynamic adjustment mechanism
Training Loop:
  for epoch $= 1$ **to** E **do**
    Sample batch: Randomly select $\{(x_0^j, x_1^j)\}_{j=1}^B$ from training set
    Sample time steps: Uniformly sample $t_j \in [0, 1]$ for $j = 1, \ldots, B$
    Sample noise: Generate $z_j \sim \mathcal{N}(0, I_d)$ for $j = 1, \ldots, B$
    Compute transport path: $x_t^j = I(t_j, x_0^j, x_1^j) + \gamma(t_j) z_j$ for $j = 1, \ldots, B$
    Update parameters:
      Call Algorithm 1 (Velocity Field and Score Function Learning) to compute $\mathcal{L}_v$, $\mathcal{L}_s$ and update θ, ϕ
    Monitor convergence: Evaluate $\mathcal{L}_v$, $\mathcal{L}_s$ on validation set
    If convergence criteria met **or** epoch $> E/2$:
      Call Algorithm 2 (Dynamic Adjustment Mechanism) to update $\sigma(t)$:
        $\delta = \alpha \cdot \tanh(\beta \cdot (\mathcal{L}_v - \eta \mathcal{L}_s))$
        $\sigma(t) = 1 + \delta$
        $\gamma(t) = \sqrt{t(1-t)} \cdot \sigma(t)$
    end if
  end for
Fair Representation Generation:
  Call Algorithm 3 (Fair Graph Representation Generation) with:
    Trained parameters θ, ϕ
    Initial time $t_0 = 0$, final time $t_f = 1$
    Time step $\Delta t = 0.01$
    Noise function $\gamma(t)$ and diffusion coefficient ϵ_0
  Return the generated fair representation samples $\{\hat{x}_1^{(i)}\}_{i=1}^n$
Output:
  Return trained parameters θ, ϕ and fair representation samples $\{\hat{x}_1^{(i)}\}_{i=1}^n$

D.3 End-to-End Training and Inference Framework

While the previous subsections have detailed the individual components of our framework—specifically the velocity field and score function learning (Algorithm 1), the dynamic adjustment mechanism (Algorithm 2), and the fair representation generation process (Algorithm 3), a comprehensive view of the entire training and inference pipeline is essential for practical implementation. The following algorithm (Algorithm 4) provides a unified framework that integrates these components into a cohesive workflow, enabling both model training and fair representation generation in a single, end-to-end process. This framework ensures that all parameters (e.g., θ, ϕ, and $\gamma(t)$) are consistently updated and applied across the training and inference phases, facilitating reproducibility and practical deployment.

D.4 Training strategies and implementation details

When using linear interpolation $I(t, x_0, x_1) = (1-t)x_0 + tx_1$, denoising steps can be simplified as:

$$x_{t-\Delta t} = \frac{\beta(t-\Delta t)}{\beta(t)} x_t + \left(\alpha(t-\Delta t) - \frac{\alpha(t)\beta(t-\Delta t)}{\beta(t)} \right) \eta_{zos}(t, x_t)$$

where $\eta_{zos}(t, x_t)$ is the denoiser of one-sided spatially linear interpolant.

In specific implementation, we build a complete framework based on PyTorch and PyTorch Geometric. The graph neural network backbone uses a two layer graph attention network GAT, with hidden dimension set to 64 to effectively capture graph structural information. Both the velocity field network $\hat{v}_\theta$ and score function network $\hat{s}_\phi$ are implemented by three layer multilayer perceptrons MLPs, ensuring the model's expressive power. The optimization process uses the Adam optimizer, with learning rate 0.001 weight decay 5×10^{-4}, batch size 128, and 200 training epochs to ensure convergence. The target diffusion coefficient ϵ_0 is set to 0.1, a value determined through the performance fairness trade-off curve on the validation set, capable of achieving optimal balance on most datasets.

To verify the effectiveness of the dynamic adjustment mechanism, we implement three different $\sigma(t)$ configurations, where $\gamma(t) = \sqrt{t(1-t)} \cdot \sigma(t)$. The first configuration $\sigma_1(t)$ is constantly equal to 1, representing a baseline method without dynamic adjustment. The second configuration $\sigma_2(t)$ uses linear decay function $\sigma_2(t) = 1 - 0.8t$, with larger $\sigma(t)$ values in early stages, emphasizing counterfactual fairness, and smaller $\sigma(t)$ values in later stages, emphasizing individual fairness. The third configuration $\sigma_3(t)$ uses cosine decay function $\sigma_3(t) = 0.5 + 0.5\cos(\pi t)$, achieving smooth transition between two fairness criteria.

In the inference phase, we use a fixed step size ODE solver for sampling, with time step Δt set to 0.01. To improve sampling efficiency, we implement simplified denoising steps, avoiding complex numerical integration processes. Specifically for linear interpolation cases, we use the denoiser $\eta_{zos}(t, x_t)$ of one-sided spatially linear interpolant significantly, reducing computational complexity while maintaining the accuracy of the transport path.

E Detailed Evaluation Metrics

We describe the fairness and theoretical evaluation metrics used in our experiments.

E.1 Individual Fairness: Structural Node Distance (SND)

SND measures the consistency of model predictions for structurally similar nodes:

$$\text{SND} = \frac{1}{|P|} \sum_{(i,j)\in P} \|f(x_i) - f(x_j)\|,$$

where $P = \{(i,j) : d(S_i, S_j) < \delta\}$ is the set of node pairs with structural similarity below threshold δ, and $f(x_i)$ is the model prediction for node i. Lower values indicate higher individual fairness.

E.2 Counterfactual Fairness: CPD and SAKL

Counterfactual Prediction Difference (CPD) quantifies prediction change under counterfactual perturbation of sensitive attributes:

$$\text{CPD} = \frac{1}{N} \sum_{i=1}^{N} \|f(x_i) - f(x_i^{\text{cf}})\|,$$

where x_i^{cf} is the counterfactual version of x_i with the sensitive attribute modified.

Sensitive Attribute KL Divergence (SAKL) measures the dependence of predictions on sensitive attributes:

$$\text{SAKL} = \text{KL}(f(x \mid A = a) \,\|\, f(x \mid A = a')).$$

Lower values indicate predictions are more independent of sensitive attributes.

F Supplementary Experimental Results

F.1 Datasets

We conducted experiments on four representative datasets to comprehensively evaluate the performance of the proposed method in various scenarios:

Pokec Social Network: The largest social network in Slovakia, containing 1,635,785 users and 30,622,564 edges. We selected a subgraph (approximately 13,000 nodes) containing complete gender information for the career recommendation prediction task. The sensitive attribute is gender (binary), and the task is to predict the career categories (10 categories) that a user may be interested in.

Facebook100 Campus Network: Contains social network data from 100 US universities. We selected a subset of 10 universities with a relatively balanced gender and ethnicity distribution (approximately 45,000 nodes in total). The sensitive attributes include gender (binary) and ethnicity (categorical), and the task is to predict the campus activities that students may participate in (20 categories).

Credit-Default Graph Dataset: A graph constructed based on the UCI credit card dataset, connecting similar users using the K-nearest neighbor method (approximately 30,000 nodes). The sensitive attributes are age (categorized as young, middle-aged, and elderly) and gender. The task is to predict credit default risk.

COMPAS Judicial Graph Dataset: A heterogeneous graph constructed based on real judicial data, containing defendant-case-judge relationships (approximately 18,000 nodes). The sensitive attribute is race (white, African American). The task is to predict recidivism risk.

In addition, we constructed a Gaussian mixture graph structure synthetic dataset, which contains known structural bias patterns and is used to accurately verify theoretical bounds. This dataset has 5,000 nodes divided into five communities. The sensitive attributes are partially, but not completely, correlated with the community structure.

F.2 Gaussian Mixture Graph Structure Synthetic Dataset

To precisely characterize the generation mechanism of the Gaussian Mixture Graph synthetic dataset, we present a formal mathematical definition combined with algorithmic pseudocode.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote the generated graph structure, where $\mathcal{V}$ is the node set, $\mathcal{E}$ is the edge set, and $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the node feature matrix. This dataset is generated through the following process:

1. Community structure generation: Define the community set $\mathcal{C} = \{C_1, C_2, \ldots, C_5\}$, where each community C_k contains $n_k = 1000$ nodes, satisfying $\cup_{k=1}^{5} C_k = \mathcal{V}$ and $C_i \cap C_j = \emptyset$ for $i \neq j$. The edge set $\mathcal{E}$ is generated with the following probabilities:

$$P((u, v) \in \mathcal{E}) = \begin{cases} p_{\text{in}} = 0.05 & \text{if } u, v \in C_k \text{ for some } k \\ p_{\text{out}} = 0.01 & \text{otherwise} \end{cases}$$

2. Node feature generation: For each node $v \in C_k$, its feature vector $\mathbf{x}_v \in \mathbb{R}^{20}$ is generated as follows:

$$\mathbf{x}_v = [\mathbf{x}_v^{\text{comm}}, \mathbf{x}_v^{\text{common}}] \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$$

where $\mathbf{x}_v^{\text{comm}} \in \mathbb{R}^{10}$ represents community-specific features, $\mathbf{x}_v^{\text{common}} \in \mathbb{R}^{10}$ represents common features, $\boldsymbol{\mu}_k = [\boldsymbol{\mu}_k^{\text{comm}}, \mathbf{0}]$, $\boldsymbol{\mu}_k^{\text{comm}}$ is the center vector of community k, and $\boldsymbol{\Sigma}$ is a block-diagonal covariance matrix.

3. Sensitive attribute generation: The sensitive attribute $a_v \in \{0, 1\}$ of node $v \in C_k$ follows a Bernoulli distribution:

$$a_v \sim \text{Bernoulli}(\theta_k), \quad \theta_k = \begin{cases} 0.8 & k = 1, 2 \\ 0.2 & k = 3, 4 \\ 0.5 & k = 5 \end{cases}$$

4. Label generation: The label $y_v \in \{-1, 1\}$ of node v is generated through the following process:

$$y_v = \text{sign}(\boldsymbol{\beta}_k^T \mathbf{x}_v + \epsilon_v), \quad \epsilon_v \sim \mathcal{N}(0, 0.5)$$

where $\boldsymbol{\beta}_k = \boldsymbol{\beta}_0 + \delta_k \mathbf{e}_5$, $\boldsymbol{\beta}_0 = [1.0\mathbf{1}_5^T, 0.2\mathbf{1}_{15}^T]^T$, $\mathbf{e}_5$ is a vector with the first five dimensions as 1 and the rest as 0, and δ_k is the structural bias parameter:

$$\delta_k = \begin{cases} +0.3 & k = 1, 2 \\ -0.3 & k = 3, 4 \\ 0.0 & k = 5 \end{cases}$$

Algorithm 5 Gaussian Mixture Graph Synthetic Dataset Generation.

Require: Number of communities $K = 5$, nodes per community $n = 1000$, intra-community connection probability $p_{\text{in}} = 0.05$, inter-community connection probability $p_{\text{out}} = 0.01$
Require: Feature dimension $d = 20$, community-specific feature dimension $d_1 = 10$
Require: Sensitive attribute parameters $\boldsymbol{\theta} = [0.8, 0.8, 0.2, 0.2, 0.5]$, bias parameters $\boldsymbol{\delta} = [0.3, 0.3, -0.3, -0.3, 0.0]$
Initialize empty graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and feature matrix $\mathbf{X} \in \mathbb{R}^{Kn \times d}$
for $k = 1$ to K **do**
  Generate community center $\boldsymbol{\mu}_k^{\text{comm}} \sim \mathcal{U}([-5, 5]^{d_1})$
  for $i = 1$ to n **do**
    Add node $v_{k,i}$ to $\mathcal{V}$
    Generate community features $\mathbf{x}_{k,i}^{\text{comm}} \sim \mathcal{N}(\boldsymbol{\mu}_k^{\text{comm}}, \mathbf{I}_{d_1})$
    Generate common features $\mathbf{x}_{k,i}^{\text{common}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d-d_1})$
    $\mathbf{x}_{k,i} \leftarrow [\mathbf{x}_{k,i}^{\text{comm}}, \mathbf{x}_{k,i}^{\text{common}}]$
    Generate sensitive attribute $a_{k,i} \sim \text{Bernoulli}(\theta_k)$
    Compute weights $\boldsymbol{\beta}_k \leftarrow [1.0\mathbf{1}_5^T, 0.2\mathbf{1}_{15}^T]^T + \delta_k \mathbf{e}_5$
    Generate label $y_{k,i} \leftarrow \text{sign}(\boldsymbol{\beta}_k^T \mathbf{x}_{k,i} + \epsilon_{k,i})$, $\epsilon_{k,i} \sim \mathcal{N}(0, 0.5)$
  end for
  Add intra-community edges: for each pair of nodes $(u, v) \in C_k \times C_k$, add edge (u, v) with probability p_{in}
end for
for each pair of different communities $(C_k, C_l), k \neq l$ **do**
  Add inter-community edges: for each pair of nodes $(u, v) \in C_k \times C_l$, add edge (u, v) with probability p_{out}
end for
Output: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, sensitive attribute vector $\mathbf{a}$, label vector $\mathbf{y}$

F.3 Parameter Sensitivity Analysis

Table D.1 shows the impact of different ϵ_0 values on model performance:

Table 4: Impact of different ϵ_0 values on model performance.

ϵ_0	Accuracy	SND	CPD	SAKL
0.01	0.812	0.178	0.185	0.158
0.05	0.827	0.149	0.157	0.136
0.10	0.835	0.132	0.131	0.110
0.20	0.829	0.138	0.137	0.119
0.50	0.817	0.149	0.145	0.126

Results show that $\epsilon_0 = 0.10$ achieves the tightest bound, while maintaining the best performance fairness balance.

D.2 Comparison with Schrödinger Bridge Method

Table D.2 compares our method with traditional Schrödinger Bridge methods in fairness metrics:

Results show that our method not only optimizes both fairness metrics simultaneously, but also maintains higher model accuracy with computational efficiency comparable to traditional methods.

Table 5: Comparison of different methods in fairness metrics.

Method	Accuracy	SND	CPD	SAKL
Vanilla SB	0.819	0.167	0.172	0.148
SB + Individual Fairness	0.805	0.135	0.215	0.186
SB + Counterfactual Fairness	0.802	0.243	0.125	0.105
SI-Fair (Ours)	0.835	0.132	0.131	0.110

G Symbol Explanation

To facilitate reader understanding we list the main symbols used in this paper and their meanings:

Table 6: Symbol explanation.

Symbol	Meaning
ρ_0, ρ_1	Biased distribution and ideal fair distribution
x_t	State of stochastic interpolants process at time t
$I(t, x_0, x_1)$	Interpolation function
$\gamma(t)$	Noise function
$b(t, x)$	Drift term of transport equation
$s(t, x)$	Score function $s(t, x) = \nabla \log \rho(t, x)$
$\epsilon(t)$	Diffusion coefficient $\epsilon(t) = \gamma(t)\dot{\gamma}(t)$
ϵ_0	Target diffusion coefficient
δ	Structural similarity threshold in individual fairness constraint
$\mathcal{L}_v[\hat{v}]$	Learning objective function for velocity field
$\mathcal{L}_s[\hat{s}]$	Learning objective function for score function
S, A, C	Structural features sensitive attributes other features
SND	Structural node distance
CPD	Counterfactual prediction difference
SAKL	Sensitive attribute perturbation KL divergence
SPD(v, v')	Shortest path distance between nodes v and v'
λ_g	Balancing parameter in structural distance measure

H Conclusion and Future Work

H.1 Summary of Key Contributions

This paper proposes a unified framework based on stochastic interpolants theory to address the conflict between individual fairness and counterfactual fairness in graph neural networks. Our key contributions can be summarized as follows:

- We reformulate the fairness problem as an optimal transmission problem from a biased distribution to an ideal fair distribution. We utilize stochastic interpolants theory to construct a precise transmission path, enabling us to dynamically balance the two fairness criteria during the transmission process.
- We design a structure-attribute disentanglement representation method that decomposes node representations into structural features and sensitive attributes. This method preserves useful structural information while eliminating the undue influence of sensitive attributes.
- We prove an upper bound on the KL divergence between our model and the ideal fair distribution. We decompose this upper bound into the degree of achievement of individual fairness and counterfactual fairness, providing quantifiable fairness guarantees.
- By adjusting the $\gamma(t)$ function, a dynamic balance between the two fairness criteria is achieved during transmission, making the method adaptable to the needs of different application scenarios.
- Systematic experiments on multiple real and synthetic datasets verify the validity of the theoretical bounds and demonstrate the superiority of the proposed method in resolving fairness conflicts.

H.2 Limitation Analysis

Although our method has made significant progress, it still has some limitations:

- Compared with standard GNNs, our method requires the additional calculation of the velocity field and score function, which increases the computational overhead by approximately 20%. On large-scale graphs, this overhead may become a bottleneck.
- The method's performance is sensitive to the choice of ϵ and $\gamma(t)$, requiring adjustment based on the specific application scenario. There is a lack of an automatic parameter tuning mechanism.
- The current framework is primarily designed for a single sensitive attribute and requires further extension to handle multiple sensitive attributes.
- Although the upper bound on the KL divergence has been proven, the tightness of the bound is affected by various factors and sometimes remains significantly loose, limiting the practical value of theoretical guidance.

H.3 Future Research Directions

Based on the current work, we believe the following directions warrant further exploration:

- Explore more efficient algorithm implementations, such as utilizing the "antithetic sampling" technique mentioned in the PDF to reduce variance, or developing fast solvers specialized for graph data.
- Design an adaptive $\gamma(t)$ function to automatically adjust the weights of the two fairness measures based on data characteristics and task requirements, reducing the need for manual parameter tuning.
- Extend the framework to scenarios with multiple sensitive attributes to study the interactions between different sensitive attributes and their impact on fairness.
- Further tighten the theoretical bounds, improve the tightness of the bounds, and provide more precise quantitative metrics for fairness assurance.