FAIRSHIFT: CONTROLLER-ONLY FAIRNESS ADAPTATION OF PRETRAINED GRAPH DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fairness gaps in link prediction may reflect structural bias embedded in graph data rather than only bias in the downstream model. We therefore study fair topology generation as a data-level intervention. A pretrained graph generator should be reusable when fairness requirements change. We propose FairShift, a post-training controller that adjusts edge formation during reverse diffusion while keeping the denoiser frozen. A persistent score state supplies guidance for statistical parity (SP) or score-based equal opportunity (EO). FAIRSHIFT-T calibrates a T-parameter logit-shift schedule from replayed trajectories, whereas FAIRSHIFT-F uses a constant shift strength without controller optimization. We evaluate generated graphs through the utility and fairness of link predictors trained on them. Relative to the corresponding uncontrolled EDGE-cond backbones, selected retention-oriented FAIRSHIFT-T configurations reduce SP by 47.1%, 45.7%, and 23.6% on Cora, Citeseer, and Amazon Photo, respectively, with AUC decreases of 1.02, 0.93, and 0.53 percentage points. The edge-logit interface also supports a joint feature–edge diffusion backbone and EO control. These results demonstrate reusable fairness adaptation across the evaluated backbones and objectives without fairness-specific denoiser retraining. Code and supplementary materials are available at `https://anonymous.4open.science/r/FairShift-A6CB`.

1 INTRODUCTION

Graph learning provides a framework for modeling relational data (Xia et al., 2021), while deep graph generators synthesize networks for downstream learning and experimentation (Guo & Zhao, 2023; Li et al., 2024). Among these generators, discrete diffusion models construct graph topology through successive reverse transitions, allowing edge formation to depend on the evolving graph state (Vignac et al., 2023; Chen et al., 2023).

However, preserving graph structure or predictive utility does not ensure fairness. Group-dependent connectivity in a generated graph can become a training signal for downstream link predictors, leading to disparities in their scores (Li et al., 2021; Kose & Shen, 2024). We therefore study graph generation as a data-level fairness intervention: the generated topology, rather than only the predictor, is the object of adaptation. We evaluate dyadic statistical parity (SP) through the mean-score gap between same-group and different-group evaluation pairs. For score-based equal opportunity (EO), we restrict this comparison to held-out positive edges. Both criteria are assessed jointly with link-prediction utility.

Existing fair graph generators incorporate fairness into their training objectives or model designs (Zheng et al., 2024; Kose & Shen, 2024; Wang & Zhang, 2025; Wang et al., 2025; 2026), coupling fairness intervention to how the generator is built or trained. FAROS enables post-training adaptation by switching sensitive attributes during sampling (Badaoui et al., 2025). We instead study how to adapt edge formation while preserving the pretrained denoiser, group attributes, and node-feature mechanism. A central difficulty is that downstream fairness becomes available only after graph generation and predictor training, making its direct use at every reverse step impractical.

We propose FAIRSHIFT, a post-training framework that adapts graph generation through fairness-guided shifts of reverse edge logits. It reuses a frozen denoiser while retaining the backbone's candidate-selection rule and node-feature mechanism. A persistent score state combines cached and current logits with a fixed update coefficient k, providing a sampling-time surrogate for downstream

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.

fairness. The surrogate compares groupwise scores for SP and adds soft positive-class conditioning for score-based equal opportunity (EO).

Our primary variant, FAIRSHIFT-T, learns one shift strength per reverse step, using only T scalar parameters. These parameters are calibrated on replayed trajectories using surrogate fairness and utility-preservation losses, while k and the denoiser remain fixed. The calibrated schedule is then applied during online reverse sampling.

For comparison, FAIRSHIFT-F uses the same score construction and logit intervention with a constant shift strength across reverse steps, without controller optimization. It provides a fixed-strength reference for assessing the benefit of controller-only adaptation.

On Cora, Citeseer, and Amazon Photo, selected retention-oriented FAIRSHIFT-T configurations reduce the downstream SP gap by 47.1%, 45.7%, and 23.6%, respectively, relative to their uncontrolled EDGE-cond backbones, with AUC decreases of 1.02, 0.93, and 0.53 percentage points. The same edge-logit interface reduces SP on a joint feature–edge diffusion backbone and supports score-based EO control on all three datasets. On Amazon Photo, an EO-targeted configuration reduces the EO gap from 7.97 to 2.86 while AUC changes from 96.98 to 96.13.

Our contributions are given as follows:

- We formulate fairness adaptation as reuse of a pretrained graph-diffusion backbone, separating denoiser training from subsequent utility–fairness control.
- We introduce a persistent score surrogate for SP and score-based EO, together with a replay-calibrated controller that learns only T reverse-logit shift strengths. A fixed-strength variant provides a controller-training-free comparison.
- We evaluate adaptation on three datasets and two backbone interfaces, including matched uncontrolled, fixed-controller, and calibrated-controller comparisons and SP- versus EO-targeted guidance.

2 RELATED WORK

2.1 FAIR GRAPH LEARNING AND GENERATION

Fairness in machine learning encompasses multiple notions and intervention stages (Mehrabi et al., 2021), including similarity-based individual fairness (Dwork et al., 2012). Early approaches preprocess training data (Kamiran & Calders, 2012) or learn representations that retain useful information while obscuring protected-group membership (Zemel et al., 2013).

On graphs, previous work modifies random walks for fair embeddings (Rahman et al., 2019), uses adversarial regularization (Dai & Wang, 2021), or applies group-aware edge dropout (Spinelli et al., 2022). Agarwal et al. (2021) jointly address fairness and stability in graph representation learning. For link prediction, Li et al. (2021) formalize dyadic fairness and introduce FairAdj, while Laclau et al. (2021) repair adjacency structure using optimal transport.

Fair synthetic-data generation has also been studied beyond graph topology. FairGAN and Fairness GAN incorporate fairness into adversarial data generation (Xu et al., 2018; Sattigeri et al., 2019). DECAF embeds a structural causal model in a tabular generator and enables inference-time debiasing by modifying causal dependencies (van Breugel et al., 2021). These works motivate data-generation interventions, but do not provide the reverse edge-logit interface for relational topology studied here.

Graph-specific generators include FairGen, which combines label-informed generation with fair representation learning (Zheng et al., 2024), and FairWire, which adds a structural-fairness regularizer to diffusion-model training (Kose & Shen, 2024). FDGen and FairGEM address structural and feature bias (Wang & Zhang, 2025; Wang et al., 2025). FairGDiff introduces counterfactual treatment into forward diffusion and reverse denoising (Wang et al., 2026). The fairness mechanisms of these methods are incorporated into generator design or fairness-aware training. In contrast, FAIRSHIFT retains a pretrained denoiser and adapts its edge sampling after backbone training.

2.2 Post-Training Control of Graph Diffusion

Diffusion models learn reverse transitions for a progressive corruption process (Sohl-Dickstein et al., 2015; Ho et al., 2020). Multinomial and structured discrete diffusion extend this construction to categorical states (Hoogeboom et al., 2021; Austin et al., 2021). Graph-specific instantiations include DiGress, EDGE, and GraphMaker (Vignac et al., 2023; Chen et al., 2023; Li et al., 2024).

FAROS performs post-training fairness adaptation by switching selected sensitive attributes during graph diffusion sampling (Badaoui et al., 2025). FairShift instead controls reverse edge logits while keeping group attributes and the backbone's node-feature mechanism unchanged. Within the broader setting of discrete-diffusion guidance (Schiff et al., 2025), we focus on a persistent surrogate for downstream predictor-based fairness.

3 Problem Formulation

3.1 Frozen-Backbone Fair Graph Adaptation

Let $G = (V, E, X, s)$ be an attributed graph, where $V = \{v_1, \ldots, v_N\}$ is the node set, E is the edge set, $X \in \mathbb{R}^{N \times F}$ is the node-feature matrix, and $s \in \{1, \ldots, K\}^N$ is a multi-valued group attribute. We write $A \in \{0,1\}^{N \times N}$ for the adjacency matrix. We consider simple undirected graphs, so A is symmetric and has a zero diagonal. The universe of unordered node pairs is $\mathcal{E}_N = \{(i,j) : 1 \le i < j \le N\}$.

Let $\theta^\star$ denote the parameters of a graph-diffusion backbone trained once with its original denoising objective. FairShift does not update $\theta^\star$. In our primary feature-preserving setting, the node population, node features, and group attributes remain fixed, and a controller with parameters ω modifies only the reverse edge-sampling process:

$$\widehat{A}_\omega \sim p_{\theta^\star,\omega}\Big(\widehat{A} \mid X, s\Big), \qquad \widehat{G}_\omega = (V, \widehat{A}_\omega, X, s). \tag{1}$$

Optional backbone-specific conditioning variables, such as a degree sequence, are omitted from equation 1 for clarity. We denote by $\widehat{A}_{\text{base}}$ a sample from the same frozen backbone with the controller disabled, equivalently with zero logit-shift strength at every reverse step. Section 4.5 extends the same controller to an attributed-diffusion backbone with a separate node-feature branch.

3.2 Downstream Utility and Dyadic Statistical Parity

We evaluate a generated topology as training data for a downstream link-prediction procedure. For any topology $B \in \{0,1\}^{N \times N}$, let $g_B = \mathcal{A}_{\text{eval}}(B, X)$ denote the predictor obtained by a fixed training protocol. Its score $g_B(i,j) \in [0,1]$ is the predicted probability of an edge between v_i and v_j. Thus, fairness is not computed from adjacency counts alone; it is measured from the predictor induced by the generated training topology.

For a generated topology B, let $\mathcal{P} \subseteq \mathcal{E}_N$ denote the node pairs scored by its downstream evaluator. We suppress any dependence of $\mathcal{P}$ on B in the notation. The node group labels s are inherited from the original node metadata and remain fixed across generated graphs. Following dyadic fairness (Li et al., 2021; Kose & Shen, 2024), we partition $\mathcal{P}$ according to endpoint-group equality:

$$\begin{aligned} \mathcal{P}^{=} &= \{(i,j) \in \mathcal{P} : s_i = s_j\}, \\ \mathcal{P}^{\neq} &= \{(i,j) \in \mathcal{P} : s_i \neq s_j\}. \end{aligned} \tag{2}$$

Assuming that both subsets are nonempty, the signed downstream gap and its statistical-parity magnitude are

$$d_{\text{SP}}(B) = \frac{1}{|\mathcal{P}^{=}|} \sum_{(i,j) \in \mathcal{P}^{=}} g_B(i,j) - \frac{1}{|\mathcal{P}^{\neq}|} \sum_{(i,j) \in \mathcal{P}^{\neq}} g_B(i,j), \tag{3}$$

$$\Delta_{\text{SP}}(B) = |d_{\text{SP}}(B)|. \tag{4}$$

For a specified evaluation pair set, SP uses only the predictor scores and endpoint-group memberships, without conditioning on edge-existence labels.

Lower Δ_{SP} indicates a smaller disparity between same-group and different-group link scores. Utility is measured by the AUC of g_B on the held-out evaluation pairs, which we denote by $U(B) = \mathrm{AUC}(g_B; \mathcal{P})$. Both $U(\widehat{A}_\omega)$ and $\Delta_{\mathrm{SP}}(\widehat{A}_\omega)$ are computed only after graph generation and downstream-model training are complete.

3.3 Utility–Fairness Trade-offs and Reverse-Logit Interface

For a fixed pretrained backbone $\theta^\star$, a controller ω induces expected downstream disparity and utility

$$\mathcal{F}(\omega) = \mathbb{E}\Big[\Delta_{\mathrm{SP}}(\widehat{A}_\omega)\Big], \mathcal{U}(\omega) = \mathbb{E}\Big[U(\widehat{A}_\omega)\Big],$$

where the expectations account for graph-generation and downstream-training randomness under the stated evaluation protocol. The controller family defines the attainable trade-off set $\mathcal{R}_{\theta^\star} = \big\{\big(\mathcal{F}(\omega), \mathcal{U}(\omega)\big) : \omega \in \Omega\big\}, \theta = \theta^\star$. We seek operating points with smaller disparity and higher utility without prescribing a single AUC budget. We report configurations selected after inspecting their utility–fairness trade-offs; Appendix D.4 specifies the reporting procedure.

At reverse step t, the frozen denoiser exposes a candidate pair set $\mathcal{C}_t$ and an edge-existence logit $z_{ij,t}$ for each $(i, j) \in \mathcal{C}_t$. Its corresponding reverse edge probability is

$$\pi_{ij,t} = p_{\theta^\star}\big(A^{t-1}_{ij} = 1 \mid A^t, \mathcal{I}_t, t\big) = \sigma(z_{ij,t}), \tag{5}$$

where $\mathcal{I}_t$ collects backbone-specific conditioning information. For a multi-class edge decoder whose class 0 denotes no edge, $z_{ij,t}$ is the log-odds of all edge-present classes against class 0; in the binary case it is the difference between the positive- and negative-edge logits.

FairShift operates through the generic transformation $\mathcal{T}_\omega : (z_t, \mathcal{C}_t, s, H_{t+1}, t) \mapsto (\widetilde{z}_t, H_t)$, and samples A^{t-1} from $\widetilde{z}_t$ while leaving $\theta^\star$ unchanged. Here H_t contains the persistent score state and, for EO, the additional condition state. The next section constructs $\mathcal{T}_\omega$ using a reverse-time surrogate for the unavailable downstream gap in Eq. (4).

4 FairShift

4.1 Overview

FairShift addresses two sampling-time challenges: the downstream fairness metric is unavailable before graph generation and predictor training, and a sparse backbone may score different candidate pairs at successive reverse steps. We maintain a persistent pair-score state so that the fairness proxy is evaluated over a fixed tracked population. At each step, the controller refreshes active scores, computes the proxy, and shifts the current edge logits. FairShift-T calibrates the shift strengths on replayed trajectories, whereas FairShift-F uses a constant strength. We first describe the SP construction and then its score-based EO extension.

4.2 Reverse-Time Fairness State and Logit Steering

Pair pool and active updates. Let $\mathcal{U}$ denote the persistent pool of tracked unordered pairs and let $\mathcal{C}_t \subseteq \mathcal{U}$ contain the pairs scored by the backbone at reverse step t. We write $e = (i, j)$ and partition $\mathcal{U}$ into $\mathcal{U}^=$ and $\mathcal{U}^\neq$ according to whether $s_i = s_j$. The current EDGE implementation uses $\mathcal{U} = \mathcal{E}_N$ but refreshes only $\mathcal{C}_t$ at each step. Thus, an inactive pair retains its cached score rather than being removed from the proxy population.

We construct the score surrogate with a fixed update coefficient $k \in [0, 1]$. For each configuration, k remains constant across reverse steps and controller optimization. For each tracked pair, let $m_{e,t+1}$ be the memory logit before processing step t, where sampling proceeds from T to 1. The provisional state and score are

$$\begin{aligned} \bar{m}_{e,t} &= \begin{cases} (1-k)m_{e,t+1} + kz_{e,t}, & e \in \mathcal{C}_t, \\ m_{e,t+1}, & e \notin \mathcal{C}_t, \end{cases} \\ \bar{q}_{e,t} &= \sigma(\bar{m}_{e,t}). \end{aligned} \tag{6}$$

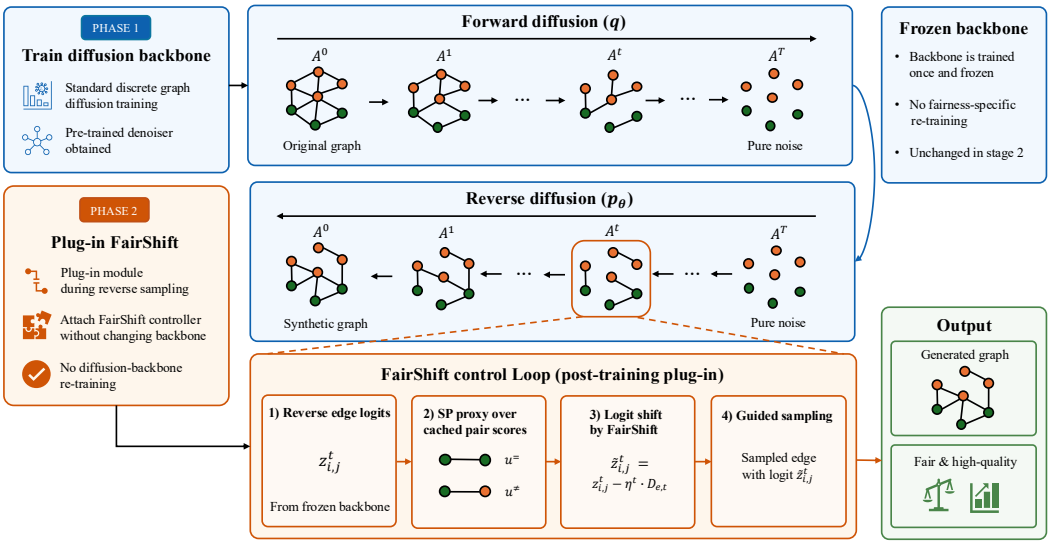


Figure 1: Shared FAIRSHIFT sampling loop with a frozen denoiser. T calibrates the per-step logit-shift strengths, whereas F uses a constant strength. The EO extension changes the fairness proxy.

All states are initialized from the backbone's graph-density prior; Appendix B specifies the initialization and cached reductions. The SP proxy is

$$\delta_t^{\mathrm{SP}} = \frac{1}{|\mathcal{U}^{=}|} \sum_{e \in \mathcal{U}^{=}} \bar{q}_{e,t} - \frac{1}{|\mathcal{U}^{\neq}|} \sum_{e \in \mathcal{U}^{\neq}} \bar{q}_{e,t}. \tag{7}$$

This is a surrogate of the downstream score gap, not an evaluation of a predictor trained on an intermediate graph.

Logit intervention. For either metric, write the signed proxy as δ_t and set $\Phi_t = \frac{1}{2}\delta_t^2$. With the incoming state held fixed, the SP gradient for an active pair is

$$\begin{aligned} G_{e,t} &= \frac{\partial \Phi_t}{\partial z_{e,t}} = \delta_t a_e k \bar{q}_{e,t}(1 - \bar{q}_{e,t}), \\ a_e &= \frac{\mathbf{1}\{e \in \mathcal{U}^{=}\}}{|\mathcal{U}^{=}|} - \frac{\mathbf{1}\{e \in \mathcal{U}^{\neq}\}}{|\mathcal{U}^{\neq}|}. \end{aligned} \tag{8}$$

Both FairShift-T and FairShift-F use graph-wise normalized guidance in the primary SP and EO experiments: $D_{e,t} = G_{e,t} / \max(\mathrm{mean}_{f \in \mathcal{C}_t} |G_{f,t}|, 10^{-30})$, with $D_{e,t} = 0$ for every active pair when the mean is zero. We then use

$$\begin{aligned} \widetilde{z}_{e,t} &= z_{e,t} - \eta_t D_{e,t}, \\ m_{e,t} &= (1-k) m_{e,t+1} + k \widetilde{z}_{e,t}, \\ A_{ij}^{t-1} &\sim \mathrm{Bernoulli}\big(\sigma(\widetilde{z}_{e,t})\big), \qquad e \in \mathcal{C}_t. \end{aligned} \tag{9}$$

Pairs outside $\mathcal{C}_t$ keep their state. Invalid comparison populations receive zero guidance. When $\eta_t = 0$, the backbone's sampling distribution is unchanged. For fixed weights and sufficiently small effective steps, the shift is a local descent direction for Φ_t; this does not imply monotonic improvement of final downstream fairness.

4.3 FAIRSHIFT-T: CONTROLLER-ONLY ADAPTATION

With k fixed, our primary variant, FAIRSHIFT-T, learns only the per-step logit-shift strengths:

$$\eta_t = \eta_{\mathrm{ref}} \frac{\mathrm{softplus}(\xi_t)}{\mathrm{softplus}(0)}, \qquad \omega = \{\xi_t\}_{t=1}^{T}. \tag{10}$$

Here $\eta_{\text{ref}} > 0$ is a fixed initialization scale. Only these T scalars are optimized. Replayed candidate sets and raw denoiser outputs are fixed inputs, and the denoiser is never updated by the controller objective.

Let $q_{e,t} = \sigma(m_{e,t})$ denote the post-guidance score. The calibration objective is

$$\begin{aligned} \mathcal{L}_{\text{ctrl}} &= w_{\text{fair}}\mathcal{L}_{\text{fair}} + w_{\text{util}}\mathcal{L}_{\text{util}}, \\ \mathcal{L}_{\text{fair}} &= \frac{1}{2|\mathcal{G}_{\text{valid}}|} \sum_{g \in \mathcal{G}_{\text{valid}}} \left(\delta_g^{\text{final}}\right)^2, \qquad \theta = \theta^\star. \end{aligned} \tag{11}$$

Here δ_g^{final} is evaluated on the final carried scores using the chosen metric. If no graph is valid, the fairness term is zero. The preservation term penalizes deviations from the provisional unshifted score:

$$\mathcal{L}_{\text{util}} = \sum_t \left\langle D_{\text{KL}}(\text{Bern}(\text{sg}[\bar{q}_{e,t}]) \,\|\, \text{Bern}(q_{e,t})) \right\rangle_t . \tag{12}$$

Here $\langle\cdot\rangle_t$ averages over active pairs in the replay mini-batch, and sg denotes stop-gradient. These losses optimize the shift schedule through the constructed score surrogate, rather than directly optimizing downstream AUC. During replay, the guidance direction is recomputed at each optimization step but detached during backpropagation; the guided logits retain their dependence on η_t. Appendix B specifies the gradient paths and loss reductions.

After calibration, the learned schedule is fixed and applied during online reverse sampling. At each step, raw edge logits are evaluated on the current graph state produced by the preceding guided transitions, rather than reused from a recorded unguided trajectory. Appendix D.4 describes operating-point selection.

4.4 FairShift-F: Fixed-Strength Control

FairShift-F uses the same fixed-k score construction and sets $\eta_t = \eta$ at every reverse step, without controller optimization. We evaluate fixed configurations over a parameter sweep; its results and reporting procedure are given in Appendix E and Appendix D.4, respectively.

4.5 Backbone Interfaces, Extensions and Computational Cost

The primary feature-conditioned EDGE-style backbone first exposes pairs through its active-node mechanism and then scores their edges (Chen et al., 2023). FairShift does not modify the selection rule or its parameters. In a FairWire-style attributed backbone, the controller modifies only the edge branch, leaving the node-feature mechanism unchanged (Kose & Shen, 2024); the plug-in comparison uses a backbone trained with $\lambda_{\text{FW}} = 0$.

Extension to score-based equal opportunity. Both FairShift-T and FairShift-F support a soft-positive-conditioned score objective through the same controller interface.

We replace the SP gap by

$$\delta_t^{\text{EO}} = \frac{\sum_{e \in \mathcal{U}^=} w_{e,t}\bar{q}_{e,t}}{\sum_{e \in \mathcal{U}^=} w_{e,t}} - \frac{\sum_{e \in \mathcal{U}^\neq} w_{e,t}\bar{q}_{e,t}}{\sum_{e \in \mathcal{U}^\neq} w_{e,t}}. \tag{13}$$

Here $\bar{q}_{e,t}$ is the pre-correction score, and $w_{e,t}$ is a detached soft-positive weight from a separate memory of pre-correction denoiser logits. The weights and their normalizers are fixed in the local guidance derivative. The proxy has a conditional mean-score interpretation under an auxiliary positive-label model, rather than enforcing thresholded TPR parity. Appendix A develops its relation to downstream EO; Appendix F specifies the generated-positive evaluation protocol.

Computational cost. Let $M_{\text{bb}}(N)$ denote the backbone's sampling memory and $\mathcal{U}$ the controller's tracked pair pool. Cached group statistics require $O(|\mathcal{U}|)$ initialization and $O(|\mathcal{C}_t|)$ additional arithmetic at reverse step t, without an additional denoiser forward pass. The controller stores $O(|\mathcal{U}|+T)$ state, giving total sampling memory $M_{\text{bb}}(N) + O(|\mathcal{U}| + T)$. Our implementation uses $\mathcal{U} = \mathcal{E}_N$. Thus, its full-pair cache preserves the quadratic asymptotic memory order of a dense backbone but can limit the memory advantage of a sparse backbone. The reverse-logit interface does not require a dense backbone; evaluating memory-efficient controller states is left to future work.

Table 1: SP operating points. Each cell reports mean AUC / Δ_{SP} (%). R: published reference; O: our experiment. Standard deviations are reported in Appendix E. The EDGE-cond SP controller settings and evaluation protocols are given in Appendix D.6. Reference rows are contextual, not matched-checkpoint comparisons.

Method	Cora	Citeseer	Amazon Photo
GraphMaker-style^{O}	87.31/35.56	92.19/37.56	94.45/33.49
FairAdj^{O}	82.13/15.47	82.67/15.45	—
Adversarial^{O}	83.66/16.35	89.59/24.20	94.24/29.17
FairWire^{O}	86.81/14.79	91.47/20.71	92.28/26.22
FAROS^{R}	89.08/13.30	84.45/12.39	94.10/27.03
EDGE-cond, base^{O}	93.60/26.07	95.03/26.83	96.98/32.35
FAIRSHIFT-F^{O}	92.74/24.85	93.75/24.50	96.71/29.71
FAIRSHIFT-T (ret.)^{O}	92.58/13.78	94.10/14.58	96.45/24.72
FAIRSHIFT-T (fair.)^{O}	87.52/9.31	93.02/9.25	95.69/20.42

5 EXPERIMENTS

We evaluate FAIRSHIFT-T first, then isolate controller learning and transfer across backbones. SP is the main target. We also evaluate the score-based EO extension in Appendix F.

5.1 EVALUATION OVERVIEW

We evaluate on Cora, Citeseer, and Amazon Photo, using node categories as benchmark group attributes. For each controller configuration, we evaluate eight generated graphs and train a one-layer GCN link predictor with an 80/10/10 edge split on each graph. Evaluation pairs consist of held-out edges and nonedges of the same generated graph. We report AUC and SP in percent, with deviations summarizing variability across generated-graph evaluations. *Ret.* and *fair.* are descriptive labels for manually selected configurations emphasizing AUC retention and fairness, respectively; no fixed AUC threshold was imposed (Appendix D.4). Matched controller comparisons keep the backbone checkpoint, diffusion steps, and evaluator settings fixed and apply the same pair-construction and splitting rules; detailed evaluation settings are given in Appendix D, with SP controller configurations and reproduction protocols in Appendix D.6.

R and O denote published and in-house results, respectively. FAROS is included as a reported reference (Badaoui et al., 2025), since its implementation was unavailable to us.

5.2 MAIN UTILITY–FAIRNESS PERFORMANCE

Within-backbone adaptation. Relative to uncontrolled EDGE-cond, selected retention-oriented FAIRSHIFT-T configurations reduce SP by 47.1%, 45.7%, and 23.6% on Cora, Citeseer, and Amazon Photo, respectively, with AUC decreases of 1.02, 0.93, and 0.53 percentage points. The fairness-oriented configurations reach AUC/SP pairs of $87.52/9.31$, $93.02/9.25$, and $95.69/20.42$, respectively, providing additional SP reductions at lower AUC. Importantly, this highlights that feature-conditioned diffusion is capable of striking a highly effective balance between utility and fairness.

Contextual comparisons. The fairness-oriented configurations attain higher mean AUC and lower mean SP than the reproduced FairWire references shown in Table 1 on all three datasets. Compared with the published FAROS references, FAIRSHIFT-T (fair.) attains higher mean AUC and lower mean SP on Citeseer and Amazon Photo, while the Cora configurations trade utility against fairness. These cross-generator comparisons provide context rather than isolating controller effects or establishing statistical significance.

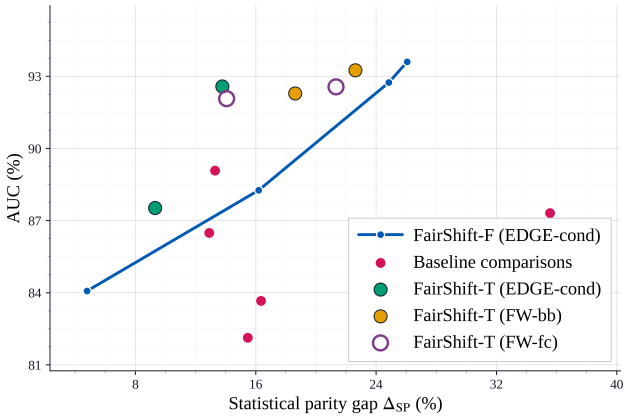


Figure 2: Cora AUC–SP trade-offs. The curve connects evaluated Uncontrolled and FAIRSHIFT-F settings, and FAIRSHIFT-T supplies calibrated operating points. The published FAROS and reproduced FairWire points are contextual references from different generators; FW-fc is diagnostic. Additional EDGE-cond trade-off plots for Citeseer and Amazon Photo are provided in Figure 3 (Appendix E.6).

Table 2: Transfer to the fairness-agnostic FairWire-style backbone (FW-bb, $\lambda_{\mathrm{FW}} = 0$). All FAIRSHIFT variants retain the frozen denoiser. *Ret.* and *fair.* follow the descriptive selection labels defined in Appendix D.4. Cells report mean AUC/Δ_{SP} (%). Standard deviations are reported in Appendix E.6.

Controller	Cora	Citeseer	Amazon Photo
Uncontrolled	97.42/44.36	79.17/29.24	97.82/46.99
FAIRSHIFT-F	95.40/40.72	77.21/26.33	97.09/31.08
FAIRSHIFT-T (ret.)	93.25/22.63	75.12/23.39	97.55/28.57
FAIRSHIFT-T (fair.)	92.29/18.62	72.06/5.56	96.90/22.84

5.3 WHAT DOES CONTROLLER-ONLY LEARNING ADD?

Shift scheduling. On EDGE-cond, the reported FAIRSHIFT-T configurations attain lower mean SP than the displayed FAIRSHIFT-F settings at similar mean AUC. This comparison combines controller fitting with a more flexible temporal parameterization; it does not by itself isolate the contribution of time dependence. To separate these factors, we compare a single calibrated shift strength shared across all reverse steps with the full step-dependent schedule, using the same fixed coefficient, controller objective, and calibration and selection budgets. At the reference update coefficient, stepwise shifts increase mean AUC by 3.23 percentage points and reduce SP by 1.91 points on Cora. On Citeseer, they reduce SP by 1.43 points with nearly unchanged AUC (94.10 versus 94.18). On Amazon Photo, they increase AUC by 0.28 points and reduce SP by 0.73 points. The observed scheduling benefit is largest on Cora.

Temporal blending. Replacing the reference coefficient by $k = 1$ removes blending with the previous score at active updates, yielding last-value caching while retaining inactive pair states. With stepwise shifts, this increases SP by 5.76, 8.14, and 3.36 points on Cora, Citeseer, and Amazon Photo, respectively. Appendix E.8 reports all four configurations for each dataset and their graph-level variability.

5.4 TRANSFERABILITY AND SUPPORTING ANALYSES

Transfer to a FairWire-style backbone. The same reverse-logit interface supports SP control on the frozen joint feature–edge backbone, with dataset-dependent utility costs. On Cora, FAIRSHIFT-T (ret.) changes the AUC/SP pair from 97.42/44.36 to 93.25/22.63; on Amazon Photo, it changes 97.82/46.99 to 97.55/28.57. On Citeseer, FAIRSHIFT-T (fair.) changes 79.17/29.24 to 72.06/5.56.

These results extend controller-based fairness adaptation to a second backbone without denoiser retraining.

Extension to score-based EO. The same controller interface supports positive-conditioned guidance on all three datasets. On Cora, EO-targeted C2 attains a lower mean EO gap than SP-targeted guidance (4.68 versus 5.17), with lower AUC (91.21 versus 92.58). On Citeseer, S3 reduces EO from 6.80 to 3.00 while AUC changes from 95.03 to 94.68. Compared with SP-targeted guidance, S3 attains higher mean AUC and lower mean EO, but a larger SP gap (18.49 versus 14.58). On Amazon Photo, A1 reduces EO from 7.97 to 2.86 with an AUC decrease from 96.98 to 96.13; it also attains lower EO than SP-targeted guidance (3.42), at a 0.32-point AUC cost. These results illustrate target-dependent utility–fairness trade-offs. Appendix F reports uncontrolled, fixed, and calibrated comparisons over eight generated-graph evaluations per configuration.

Sampling-time proxy diagnostics. The Cora study in Appendix C examines the two components used by the controller. Late-stage running scores show positive rank association with the terminal GCN, while condition weights allocate a larger share of their total mass to generated positives than uniform weighting. These observations support the respective link-ranking and positive-emphasis roles of the proxy components.

6 Discussion and Future Work

Discussion. The results support post-training fairness control rather than establish equivalence between reverse edge probabilities and independently trained link predictors. The proxy guides optimization, while final utility and fairness are evaluated downstream. In sparse samplers, candidate selection determines which pairs receive direct intervention.

Future work. The controller is not tied to the two evaluated backbone families; its interface requires accessible edge logits and candidate-pair group relations. Future work will combine efficient backbones and controller states for larger graphs, and broaden matched comparisons across graph domains and fair generators.

7 Conclusion

We presented FairShift, a post-training framework for reusing graph diffusion models under different fairness requirements. The controller adjusts reverse edge logits while preserving the pre-trained denoiser; fixed and replay-calibrated variants offer different adaptation costs. Experiments show utility–fairness trade-offs on three datasets, transfer across topology and joint feature–edge diffusion interfaces, and adaptation to SP and score-based EO objectives. The central result is that fairness adaptation can be separated from denoiser retraining through a shared generation-time interface.

AI Use Statement

In this work, we used generative AI tools to assist with drafting portions of the manuscript, polishing and improving the clarity and readability of the writing, retrieving and discovering relevant literature, and supporting research execution through code development and implementation.

All AI-assisted content and code were reviewed, revised, and verified by the authors. In particular, literature identified through AI-assisted retrieval was independently checked against the original publications, AI-assisted text was manually reviewed for technical accuracy, consistency with our methodology and experimental results, and appropriate attribution.

The authors take full responsibility for the final content of this work, including all text, claims, citations, code, experimental results, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies fairness-aware graph generation using established public benchmark datasets. The node categories used as group attributes are adopted for controlled evaluation following prior fair graph-generation benchmarks and are not intended to represent legally protected demographic attributes. Our experiments do not involve human subjects or newly collected personal data. While the proposed method aims to mitigate group-level disparities in generated graphs, fairness definitions and group constructions are application-dependent, and deployment in real-world settings would require domain-specific assessment of potential harms and affected populations.

REFERENCES

Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a unified framework for fair and stable graph representation learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 2114–2124. PMLR, 2021. URL https://proceedings.mlr.press/v161/agarwal21b.html.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Abdennacer Badaoui, Oussama Kharouiche, Hatim Mrabet, Daniele Malitesta, and Fragkiskos D. Malliaros. FAROS: Fair graph generation via attribute switching mechanisms. *arXiv preprint arXiv:2507.03728*, 2025. doi: 10.48550/arXiv.2507.03728. URL https://arxiv.org/abs/2507.03728.

Xiaohui Chen, Jiaxing He, Xu Han, and Liping Liu. Efficient and degree-guided graph generation via discrete diffusion modeling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4585–4610. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23k.html.

Enyan Dai and Suhang Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 680–688. ACM, 2021. doi: 10.1145/3437963.3441752. URL https://dl.acm.org/doi/10.1145/3437963.3441752.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 214–226. ACM, 2012. doi: 10.1145/2090236.2090255. URL https://dl.acm.org/doi/10.1145/2090236.2090255.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5370–5390, 2023. doi: 10.1109/TPAMI.2022.3214832. URL https://ieeexplore.ieee.org/document/9920219.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3315–3323, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/6a9659feb1216f14f7384ba499518b38-Abstract.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https:

//proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/67d96d458abdef21792e6d8e590244e7-Abstract.html.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012. doi: 10.1007/s10115-011-0463-8. URL https://link.springer.com/article/10.1007/s10115-011-0463-8.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

O. Deniz Kose and Yanning Shen. FairWire: Fair graph generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124451–124478, 2024. doi: 10.52202/079017-3953. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e105d1cabb6ee2495595e2baf25493b5-Abstract-Conference.html.

Charlotte Laclau, Ievgen Redko, Manvi Choudhary, and Christine Largeron. All of the fairness for edge prediction with optimal transport. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1774–1782. PMLR, 2021. URL https://proceedings.mlr.press/v130/laclau21a.html.

Mufei Li, Eleonora Kreačić, Vamsi K. Potluru, and Pan Li. GraphMaker: Can diffusion models generate large attributed graphs? *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=0q4zjGMKoA.

Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. On dyadic fairness: Exploring and mitigating bias in graph connections. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xgGS6PmzNq6.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6):115:1–115:35, 2021. doi: 10.1145/3457607. URL https://dl.acm.org/doi/10.1145/3457607.

Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 3289–3295, 2019. doi: 10.24963/ijcai.2019/456. URL https://www.ijcai.org/proceedings/2019/456.

Prasanna Sattigeri, Samuel C. Hoffman, Vijil Chenthamarakshan, and Kush R. Varshney. Fairness GAN: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3:1–3:9, 2019. doi: 10.1147/JRD.2019.2945519. URL https://research.ibm.com/publications/fairness-gan-generating-datasets-with-fairness-properties-using-a-generative-adversarial-network.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=i5MrJ6g5G1.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008. doi: 10.1609/aimag.v29i3.2157. URL https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2157.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018. doi: 10.48550/arXiv.1811.05868. URL https://arxiv.org/abs/1811.05868.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265. PMLR, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. FairDrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Transactions on Artificial Intelligence*, 3(3):344–354, 2022. doi: 10.1109/TAI.2021.3133818. URL https://ieeexplore.ieee.org/document/9645324.

Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. DECAF: Generating fair synthetic data using causally-aware generative networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22221–22233, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/ba9fab001f67381e56e410575874d967-Abstract.html.

Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UaAD-Nu86WX.

Wendi Wang, Jiaxi Yang, Yongkang Du, and Lu Lin. Mitigating topology biases in graph diffusion via counterfactual intervention. *arXiv preprint arXiv:2603.02005*, 2026. doi: 10.48550/arXiv.2603.02005. URL https://arxiv.org/abs/2603.02005.

Zichong Wang and Wenbin Zhang. FDGen: A fairness-aware graph generation model. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 65412–65428. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25ek.html.

Zichong Wang, Zhipeng Yin, and Wenbin Zhang. A unified framework for fair graph generation: Theoretical guarantees and empirical advances. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2175. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5ddbbb466737865d1e8d36f045fc9341-Abstract-Conference.html.

Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021. doi: 10.1109/TAI.2021.3076021. URL https://ieeexplore.ieee.org/document/9416834.

Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. FairGAN: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 570–575. IEEE, 2018. doi: 10.1109/BigData.2018.8622525. URL https://ieeexplore.ieee.org/document/8622525.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 40–48. PMLR, 2016. URL https://proceedings.mlr.press/v48/yanga16.html.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 325–333. PMLR, 2013. URL https://proceedings.mlr.press/v28/zemel13.html.

Lecheng Zheng, Dawei Zhou, Hanghang Tong, Jiejun Xu, Yada Zhu, and Jingrui He. FairGen: Towards fair graph generation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 2285–2297. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00181. URL https://ieeexplore.ieee.org/document/10597925.

A SCORE-BASED EQUAL-OPPORTUNITY EXTENSION

This appendix develops the score-based EO extension of the shared frozen-backbone controller. We first define the downstream metric and construct the soft-positive condition, then derive the local update and analyze its relation to the downstream target. Experimental protocols and results are reported separately in Appendix F.

A.1 DOWNSTREAM DEFINITION AND THE POSITIVE POPULATION

Equal opportunity conditions comparisons on the positive class (Hardt et al., 2016). For a completed generated adjacency B, let $Y_e = B_{ij}$ be the edge-existence label of pair $e = (i, j)$, and let $r(e) \in \{=, \neq\}$ denote its endpoint-group relation under the fixed node group labels s. For the evaluator's pair set $\mathcal{P}$, define $\mathcal{P}^{r,+} = \{e \in \mathcal{P} : r(e) = r,\ Y_e = 1\}$.

For a downstream predictor g_B trained on topology B, the implemented score-based gap is

$$\begin{aligned} d_{\mathrm{EO}}^{\mathrm{score}}(B) &= \frac{1}{|\mathcal{P}^{=,+}|} \sum_{e \in \mathcal{P}^{=,+}} g_B(e) - \frac{1}{|\mathcal{P}^{\neq,+}|} \sum_{e \in \mathcal{P}^{\neq,+}} g_B(e), \\ \Delta_{\mathrm{EO}}^{\mathrm{score}}(B) &= |d_{\mathrm{EO}}^{\mathrm{score}}(B)|. \end{aligned} \tag{14}$$

Both positive subsets must be nonempty. This measures a conditional *mean score* difference, not the TPR gap obtained by replacing $g_B(e)$ with $\mathbf{1}\{g_B(e) \geq \tau\}$ for a fixed threshold τ. Equality of these means also does not imply equality of the full positive-class score distributions.

The group labels s are inherited from the original node metadata, whereas positive membership $Y_e = 1$ is determined by the generated adjacency B. SP compares mean scores without conditioning on Y_e; score-based EO restricts the comparison to positive pairs. Our EO experiments use held-out positives from each generated graph, as detailed in Appendix F.

A.2 A DETACHED SOFT-POSITIVE CONDITION

Two quantities needed for downstream EO are unavailable during reverse sampling: the final predictor scores and membership in the evaluated positive set. We use the running score surrogate of the SP controller for the first quantity and a separate soft-positive weight for the second. In particular, the provisional controller score $\bar{q}_{e,t}$ is distinct from the downstream score $g_B(e)$.

Let $m_{e,t+1}$ and $c_{e,t+1}$ denote the incoming score and condition memories, respectively, and let $z_{e,t}$ be the raw denoiser logit before the current correction. The update coefficient k is fixed within each configuration and is shared by the score and condition memories. The provisional score memory is

$$\begin{aligned} \bar{m}_{e,t} &= \begin{cases} (1-k)m_{e,t+1} + kz_{e,t}, & e \in \mathcal{C}_t, \\ m_{e,t+1}, & e \notin \mathcal{C}_t, \end{cases} \\ \bar{q}_{e,t} &= \sigma(\bar{m}_{e,t}). \end{aligned} \tag{15}$$

The separate condition memory uses pre-correction logits:

$$\begin{aligned} c_{e,t} &= \begin{cases} (1 - \mathrm{sg}(k))c_{e,t+1} + \mathrm{sg}(k)\,\mathrm{sg}(z_{e,t}), & e \in \mathcal{C}_t, \\ c_{e,t+1}, & e \notin \mathcal{C}_t, \end{cases} \\ w_{e,t} &= \mathrm{sg}\big(\sigma(c_{e,t})\big). \end{aligned} \tag{16}$$

Here sg denotes stop-gradient. Both memories start from the same density prior. In particular, the implemented weight is a sigmoid of a *running unshifted logit*; it is not generally $\sigma(z_{e,t})$ from the current step alone. Before the current shift, $\bar{q}_{e,t} = \sigma(\bar{m}_{e,t})$ incorporates the raw logit into the history of previous guided updates. After the shift, the stored score is $q_{e,t} = \sigma(m_{e,t})$.

During online generation, "pre-correction" refers to the output before the current shift on the current reverse state, not a separate unguided trajectory. In a fixed replay, the condition weights are determined by the recorded raw logits and fixed k. Detachment excludes gradients through the weights and their normalizers. Different online trajectories can still produce different numerical condition weights.

A.3 Positive-Weighted Conditional Score Means

For each relation r, define the soft-positive mass and score moment on the persistent pool:

$$W_t^r = \sum_{e \in \mathcal{U}^r} w_{e,t}, \qquad S_t^r = \sum_{e \in \mathcal{U}^r} w_{e,t} \bar{q}_{e,t}, \qquad \mu_t^r(e) = \frac{w_{e,t}}{W_t^r}, \quad e \in \mathcal{U}^r. \tag{17}$$

The normalized weights are defined when $W_t^r > 0$. Eq. (13) is therefore $\delta_t^{\mathrm{EO}} = S_t^{=}/W_t^{=} - S_t^{\neq}/W_t^{\neq}$. Normalization is performed separately within each relation. A larger number of same-group candidates consequently does not, by itself, produce a larger conditional mean. The surrogate emphasizes pairs that receive more positive mass from the frozen denoiser without requiring a hard positive mask at intermediate steps.

Conditional-mean interpretation. The weighting has an exact interpretation under an *auxiliary* positive-label model. Condition on the current score and weight arrays $\{\bar{q}_{e,t}, w_{e,t}\}_{e \in \mathcal{U}}$. Draw E uniformly from $\mathcal{U}^r$ and introduce $\widetilde{Y} \mid E = e \sim \mathrm{Bernoulli}(w_{e,t})$. Because $\bar{q}_{e,t}$ is then fixed for each pair,

$$\begin{aligned} \mathbb{E}[\bar{q}_{E,t} \mid \widetilde{Y} = 1, r] &= \frac{\mathbb{E}[\widetilde{Y} \bar{q}_{E,t} \mid r]}{\Pr(\widetilde{Y} = 1 \mid r)} \\ &= \frac{\sum_{e \in \mathcal{U}^r} w_{e,t} \bar{q}_{e,t}}{\sum_{e \in \mathcal{U}^r} w_{e,t}}. \end{aligned} \tag{18}$$

Eq. (18) establishes a positive-conditioned score contrast under the auxiliary-label model. The running scores and soft-positive weights provide sampling-time surrogates for downstream predictor scores and positive membership, respectively. Appendix C examines these component roles, and Appendix F evaluates the downstream effect of using the proxy for guidance.

Two limiting cases clarify the construction. If w_e is the true binary positive mask on the evaluation pool and $\bar{q}_e = g_B(e)$, the surrogate recovers Eq. (14). If weights are constant within each relation, it reduces to the SP score gap on the same pool. Multiplying all weights within a relation by a common positive constant leaves its conditional mean unchanged; the relevant assumption concerns *relative positive mass within each relation*, not equality of global positive prevalence.

A.4 EO Logit Update and Numerical Support

For the local guidance derivative, hold the incoming histories, k, and the condition weights fixed. For an active pair $e \in \mathcal{C}_t$, define

$$\begin{aligned} a_{e,t}^{\mathrm{EO}} &= w_{e,t} \left(\frac{\mathbf{1}\{e \in \mathcal{U}^{=}\}}{W_t^{=}} - \frac{\mathbf{1}\{e \in \mathcal{U}^{\neq}\}}{W_t^{\neq}} \right), \\ G_{e,t}^{\mathrm{EO}} &= \frac{\partial \, \frac{1}{2} (\delta_t^{\mathrm{EO}})^2}{\partial z_{e,t}} = \delta_t^{\mathrm{EO}} a_{e,t}^{\mathrm{EO}} k \bar{q}_{e,t} (1 - \bar{q}_{e,t}). \end{aligned} \tag{19}$$

The factor $k\bar{q}_{e,t}(1-\bar{q}_{e,t})$ follows from Eq. (15). The derivative excludes paths through the detached weights and their normalizers.

Let $D_{e,t}^{\mathrm{EO}}$ denote the direction obtained using the same graph-wise rescaling as in Section 4.2. The numerical forward updates for the corrected logit and committed score memory are

$$\begin{aligned} \tilde{z}_{e,t} &= z_{e,t} - \eta_t D_{e,t}^{\mathrm{EO}}, \qquad e \in \mathcal{C}_t, \\ m_{e,t} &= \begin{cases} (1-k) m_{e,t+1} + k \tilde{z}_{e,t}, & e \in \mathcal{C}_t, \\ m_{e,t+1}, & e \notin \mathcal{C}_t, \end{cases} \\ q_{e,t} &= \sigma(m_{e,t}). \end{aligned} \tag{20}$$

The corrected logits are used by the backbone's edge-sampling rule. The condition memory continues to follow Eq. (16) and never commits the corrected logit. At the end of replay, the EO calibration loss uses $\sum_{e \in \mathcal{U}^r} w_{e,1} q_{e,1} / \sum_{e \in \mathcal{U}^r} w_{e,1}$ for each relation: the final stored guided scores, rather than

the pre-correction provisional scores. The local logit derivative in Eq. (19) is distinct from the shift-schedule gradients used during controller replay, which are specified in Appendix B.

A graph contributes EO guidance only when

$$|\mathcal{U}^{=}| > 0, \quad |\mathcal{U}^{\neq}| > 0, \quad W_t^{=} > \tau_m, \quad W_t^{\neq} > \tau_m, \qquad \tau_m = 10^{-6}. \tag{21}$$

If this condition fails, EO guidance is skipped at that step. Likewise, a graph contributes zero to the EO fairness loss if its conditioning masses fail the validity check when that loss is evaluated. This is an optimization convention for excluded terms, not a definition of an otherwise undefined EO gap. No pseudo-mass is added to the denominators. The checks concern the persistent pool $\mathcal{U}$, not only the currently active set $\mathcal{C}_t$: a step that refreshes one relation may remain valid because the other relation has sufficient cached mass. The downstream evaluator records EO as undefined when either actual positive subset is empty. The utility-preservation term remains separate from the EO-validity mask.

For fixed incoming histories, fixed k, fixed condition weights, and a valid graph, G_t^{EO} is the gradient of the local score-gap objective with respect to the active logits. If $D_t^{\mathrm{EO}} = s_t G_t^{\mathrm{EO}}$ with $0 < s_t < \infty$ and $G_t^{\mathrm{EO}} \neq 0$, then $\langle G_t^{\mathrm{EO}}, -D_t^{\mathrm{EO}} \rangle = -s_t \|G_t^{\mathrm{EO}}\|_2^2 < 0$. Thus the update decreases this fixed-weight local objective for sufficiently small η_t. This does not imply monotonic improvement of final downstream EO.

A.5 What Must Align with the Downstream Metric?

The auxiliary-label identity explains the form of the proxy but does not identify its positive labels with the downstream target. We therefore distinguish population positive conditioning from comparison with a completed graph and its realized evaluation set.

Population conditioning and weight error. Fix the current sampler state and stored histories, and draw a pair E uniformly from the persistent pool $\mathcal{U}$. Let $R = r(E)$, $Q = \bar{q}_{E,t}$, and $W = w_{E,t}$. For the generated-graph interpretation, let $Y = B_E$, where B is obtained by completing the current controlled sampling trajectory with the controller schedule fixed. The particular held-out evaluation set is treated separately below. All expectations in this paragraph are conditional on the fixed sampler state and histories; this conditioning is suppressed in the notation.

Let $\mathcal{F}_t$ contain the sampled pair information and the current state and histories, so that Q, W, and R are $\mathcal{F}_t$-measurable, and define $p = \Pr(Y = 1 \mid \mathcal{F}_t)$. For a relation r with $\mathbb{E}[p \mid R = r] > 0$ and $\mathbb{E}[W \mid R = r] > 0$, define

$$\begin{aligned} M_{t,r}^{\star} &:= \mathbb{E}[Q \mid Y = 1, R = r] = \frac{\mathbb{E}[pQ \mid R = r]}{\mathbb{E}[p \mid R = r]}, \\ \widetilde{M}_{t,r} &:= \frac{\mathbb{E}[WQ \mid R = r]}{\mathbb{E}[W \mid R = r]} = \frac{S_t^r}{W_t^r}. \end{aligned} \tag{22}$$

The first equality of ratios follows from conditional expectation and the measurability of Q and R. Their difference satisfies

$$\widetilde{M}_{t,r} - M_{t,r}^{\star} = \frac{\mathbb{E}[(W - p)(Q - M_{t,r}^{\star}) \mid R = r]}{\mathbb{E}[W \mid R = r]}. \tag{23}$$

Indeed, $\mathbb{E}[p(Q - M_{t,r}^{\star}) \mid R = r] = 0$; subtracting this zero term from $\mathbb{E}[W(Q - M_{t,r}^{\star}) \mid R = r]$ gives Eq. (23).

Consequently, $W = p$ is sufficient for exact recovery of the population positive-conditioned mean of Q. More generally, $W = a_r p$ within relation r, for an admissible constant $a_r > 0$, gives the same normalized mean. These are sufficient conditions, not properties established for the implemented weights. In particular, a memory of pre-correction reverse logits is not generally the conditional probability of the final target label. A native next-state edge probability and the probability of an eventual evaluation positive are different quantities.

This exactness statement concerns the population conditional mean of the controller score Q. It does not establish that Q equals the downstream score $g_B(E)$, that a finite-sample ratio estimator is unbiased, or that a soft-weighted mean equals the hard-positive mean in each realized graph.

Comparison with a realized evaluation set. The population analysis above isolates positive-conditioning error for the controller score Q. We now fix a completed graph B, its trained predictor g_B, and nonempty evaluation positive sets $\mathcal{P}^{r,+} \subseteq \mathcal{U}^r$. Define a probability distribution on each relation by

$$\nu^r(e) = \frac{\mathbf{1}\{e \in \mathcal{P}^{r,+}\}}{|\mathcal{P}^{r,+}|}, \qquad e \in \mathcal{U}^r. \tag{24}$$

For $\bar{q}_{e,t}, g_B(e) \in [0,1]$, a deterministic decomposition gives

$$\left|\delta_t^{\mathrm{EO}} - d_{\mathrm{EO}}^{\mathrm{score}}(B)\right| \le \sum_{r \in \{=,\neq\}} \left[\mathbb{E}_{e\sim\mu_t^r}|\bar{q}_{e,t} - g_B(e)| + \mathrm{TV}(\mu_t^r, \nu^r)\right]. \tag{25}$$

Here $\mathrm{TV}(\mu,\nu) = \frac{1}{2}\sum_e |\mu(e) - \nu(e)|$. To obtain the bound, insert $\mathbb{E}_{\mu_t^r} g_B$ into each groupwise difference and use $|\mathbb{E}_\mu f - \mathbb{E}_\nu f| \le \mathrm{TV}(\mu,\nu)$ for $f \in [0,1]$. The reverse triangle inequality gives the same upper bound for $\left||\delta_t^{\mathrm{EO}}| - \Delta_{\mathrm{EO}}^{\mathrm{score}}(B)\right|$.

Equation (25) separates score mismatch from conditioning-distribution mismatch for a particular realized evaluation set. It does not establish that either term is small. It also differs from the population statement above: even correct positive probabilities can distribute substantial weight over pairs that are nonpositive in a particular realization. Low weight concentration on realized positives therefore does not, by itself, refute population correctness; conversely, enrichment of realized positives does not establish accurate recovery of the target conditional mean. Appendix C reports component-level ranking and positive-enrichment diagnostics, rather than estimates of p or the two terms in Eq. (25).

B Controller Implementation and Replay Gradients

This appendix specifies the persistent-state implementation used by the EDGE backbone. It distinguishes numerical forward updates from the surrogate gradient paths used in controller replay. These details also apply when its metric switch is set to SP.

B.1 Initialization and Cached Pair-Pool Statistics

The pair pool is the complete upper triangle, $\mathcal{U} = \mathcal{E}_N$. If a degree condition d is supplied, the initialization probability for graph g is

$$p_{0,g} = \mathrm{clip}\left(\frac{\sum_i d_i}{N_g(N_g-1) + 10^{-8}}, 10^{-4}, 1 - 10^{-4}\right). \tag{26}$$

Without degrees the initial probability is 10^{-3}. The guided memory and positive-condition memory are initialized as $m_{e,T+1} = c_{e,T+1} = \mathrm{logit}(p_{0,g})$. These are priors for unvisited pairs, not observed positive labels. The separate group sums are

$$R_g^r = \sum_{e\in\mathcal{U}_g^r} q_e, \qquad W_g^r = \sum_{e\in\mathcal{U}_g^r} w_e, \qquad S_g^r = \sum_{e\in\mathcal{U}_g^r} w_e q_e. \tag{27}$$

For an active update, the cached reductions add only $q_e^{\mathrm{new}} - q_e^{\mathrm{old}}$, $w_e^{\mathrm{new}} - w_e^{\mathrm{old}}$, and $w_e^{\mathrm{new}} q_e^{\mathrm{new}} - w_e^{\mathrm{old}} q_e^{\mathrm{old}}$, respectively. Provisional reductions are used to compute guidance; post-guidance reductions are stored. Thus inactive pairs continue to influence the pool-level gap without being rescored by the backbone at every step.

B.2 Guidance Rescaling

Let $G_{e,t}$ be the unnormalized local gradient. For each graph, normalized mode computes

$$b_t = \frac{1}{|\mathcal{C}_t|}\sum_{e\in\mathcal{C}_t} |G_{e,t}|, \qquad D_{e,t} = \begin{cases} G_{e,t}/\max(b_t, 10^{-30}), & b_t > 0, \\ 0, & b_t = 0. \end{cases} \tag{28}$$

All primary SP and EO experiments use this normalized mode for both FairShift-T and FairShift-F. Normalization is computed separately for each graph and reverse step over the active candidate set $\mathcal{C}_t$. An empty active set is skipped; when $b_t = 0$, guidance is zero for every active pair. The alternative sum-scaled mode available in the implementation is not used in these experiments.

B.3 REPLAY DIFFERENTIATION

For a replayed transition, the candidate indices and raw logits z_t are constants, and k is the fixed update coefficient. Write $m^- = m_{t+1}$ and $\widehat{D}_t = \mathrm{sg}(D_t)$, where sg denotes stop-gradient. The guided score state is

$$\begin{aligned} \widetilde{z}_t &= z_t - \eta_t \widehat{D}_t, \\ m^+ &= (1-k)m^- + k\widetilde{z}_t, \qquad q^+ = \sigma(m^+). \end{aligned} \tag{29}$$

The positive-condition memory follows Eq. (16) with the same fixed k. The direction is recomputed numerically at each optimization step but is treated as constant during that step's backpropagation. The guided logit $\widetilde{z}_t$ is not detached, preserving the path from the fairness and preservation losses to η_t.

Let $\langle f_e \rangle_t$ denote the mean over active pairs across the replay mini-batch at reverse step t. The utility reference and preservation loss are

$$\begin{aligned} m^{\mathrm{ref}} &= \mathrm{sg}\big((1-k)m^- + kz_t\big), \\ \mathcal{L}_{\mathrm{util}} &= \sum_t \Big\langle D_{\mathrm{KL}}\big(\mathrm{Bern}\big(\sigma(m_e^{\mathrm{ref}})\big) \,\big\|\, \mathrm{Bern}(q_e^+)\big)\Big\rangle_t. \end{aligned} \tag{30}$$

This compares unshifted and shifted memory-state probabilities, not the raw logits directly. The final fairness loss averages $\frac{1}{2}(\delta_g^{\mathrm{final}})^2$ over valid graphs. If no graph is valid, that fairness term is zero; the utility term remains separate. The optimizer contains only the shift-schedule parameter tensor $\omega = (\xi_1, \ldots, \xi_T)$. The update coefficient k and backbone parameters are fixed. Replay does not differentiate through the stochastic generator or the downstream predictor.

A calibrated time-constant comparator. We define a tied-schedule restriction of FAIRSHIFT-T by

$$\eta_t = c(\xi) = \eta_{\mathrm{ref}} \frac{\mathrm{softplus}(\xi)}{\mathrm{softplus}(0)}, \qquad t = 1, \ldots, T. \tag{31}$$

Only one scalar ξ is optimized using the fairness and preservation objective; k and the backbone remain fixed. Unlike FAIRSHIFT-F, this comparator undergoes controller calibration. Unlike the full schedule, it cannot choose different strengths for different reverse steps. At inference it belongs to the same time-constant policy family as F. This definition does not imply that it outperforms a sufficiently tuned F sweep or that a full schedule will outperform it. Appendix E.8 compares this calibrated shared-strength restriction with the full schedule and evaluates the effect of replacing temporal blending by last-value caching.

Fixed-replay sensitivity of the score memory. Consider a pair e visited n_e times in reverse execution order $t_1 > \cdots > t_{n_e}$. In the replay differentiation branch, raw logits and $\widehat{D}_{e,t} = \mathrm{sg}(D_{e,t})$ are held fixed. Assuming the incoming memory retains its autograd connection, unrolling the written recurrence gives

$$m_e^{\mathrm{final}} = (1-k)^{n_e} m_e^{\mathrm{init}} + k \sum_{j=1}^{n_e} (1-k)^{n_e - j} \left(z_{e,t_j} - \eta_{t_j} \widehat{D}_{e,t_j} \right). \tag{32}$$

The partial derivative in this fixed replay graph is therefore

$$\left. \frac{\partial m_e^{\mathrm{final}}}{\partial \eta_{t_j}} \right|_{\text{fixed replay direction}} = -k(1-k)^{n_e - j} \widehat{D}_{e,t_j}. \tag{33}$$

The recurrence follows by induction over visits. Between visits the pair state is unchanged, so the exponent counts subsequent visits to that pair, not all intervening diffusion steps. At $k = 1$, only the last visit remains in the final memory; for $0 < k < 1$, earlier direct interventions can also contribute. This is a property of the stated stop-gradient replay branch, not the total derivative of the online sampler or of downstream fairness. If an implementation detaches carried memories between visits, the displayed replay derivative does not describe that implementation.

For the tied parameterization, the replay chain rule accumulates all stepwise contributions,

$$\frac{\partial \mathcal{L}_{\text{ctrl}}}{\partial \xi} = c'(\xi) \sum_{t=1}^{T} \frac{\partial \mathcal{L}_{\text{ctrl}}}{\partial \eta_t}\bigg|_{\eta_1=\cdots=\eta_T=c(\xi)}, \tag{34}$$

with the same stop-gradient conventions. This explains the parameter restriction; it does not prove that the additional degrees of freedom are needed empirically.

B.4 Storage and Evaluation Scope

FairShift-T has T trainable shift-schedule scalars, whereas its persistent pair state uses $O(|\mathcal{U}|)$ storage. The evaluated implementation allocates $\mathcal{U} = \mathcal{E}_N$. Recorded candidates and logits additionally require storage proportional to $\sum_t |\mathcal{C}_t|$ per replay trajectory; peak calibration memory also depends on replay batching and differentiation. These quantities are distinct from the backbone's storage and from the controller parameter count.

During evaluation, the supplied GAE path computes EO from positive test labels belonging to the generated adjacency matrix. An absent positive group is recorded as undefined, not as zero gap.

C Proxy Rationale and Component-Level Diagnostics

FairShift uses two complementary sampling-time quantities: a running score $\bar{q}_t$ representing relative link propensity and a weight w_t emphasizing the positive-edge condition required by EO. We motivate their construction and assess these two component roles through downstream ranking agreement and generated-positive enrichment.

C.1 Proxy construction and motivation

Let $\mathcal{U}$ be the controller's full unordered pair cache, and partition any pair set $\mathcal{S}$ into $\mathcal{S}^{=}$ and $\mathcal{S}^{\neq}$ by endpoint label equality. For nonnegative weights v, define

$$D^{v}_{\mathcal{S}}(f) = \frac{\sum_{e\in\mathcal{S}^{=}} v_e f_e}{\sum_{e\in\mathcal{S}^{=}} v_e} - \frac{\sum_{e\in\mathcal{S}^{\neq}} v_e f_e}{\sum_{e\in\mathcal{S}^{\neq}} v_e}. \tag{35}$$

Both denominators must be positive; undefined cases are not set to zero.

At reverse step t, let C_t be the active candidate set and $\pi_t(e) = \sigma(z_{e,t})$ the native next-state edge probability. For $e \in \mathcal{C}_t$, the incoming score and condition memories are $m_{e,t+1}$ and $c_{e,t+1}$, respectively. The corresponding provisional score and detached condition weight are

$$\bar{q}_{e,t} = \sigma((1-k)m_{e,t+1} + kz_{e,t}), \qquad e \in \mathcal{C}_t. \tag{36}$$

$$w_{e,t} = \mathrm{sg}[\sigma((1-k)c_{e,t+1} + kz_{e,t})]. \tag{37}$$

Here k is the fixed update coefficient and sg denotes stop-gradient. Inactive entries are retained. The score cache commits shifted logits, whereas the condition cache commits unshifted logits; consequently, w_t is distinct from both raw π_t and, in general, $\mathrm{sg}(q_t)$. The two objectives are

$$d_{\text{SP},t} = D^{1}_{\mathcal{U}}(\bar{q}_t), \qquad d_{\text{EO},t} = D^{w_t}_{\mathcal{U}}(\bar{q}_t), \qquad \Phi_{M,t} = \frac{1}{2} d^2_{M,t}. \tag{38}$$

Reverse probabilities as link-formation information. Condition on the current state and candidate selection, denoted by $\mathcal{I}_t$. Under the native, unshifted transition, linearity gives

$$\mathbb{E}_\theta\big[D^{\mathbf{1}}_{C_t}(A^{t-1}) \mid \mathcal{I}_t\big] = D^{\mathbf{1}}_{C_t}(\pi_t), \tag{39}$$

whenever both candidate relations are present. Thus reverse probabilities provide the sampler's conditional expectations of edge formation, and their contrast measures expected current structural disparity.

Fixed-weight logit pooling. Write $K(u, a) = D_{\mathrm{KL}}(\mathrm{Bern}(u) \| \mathrm{Bern}(a))$. For an active pair $e \in \mathcal{C}_t$, finite logits, and fixed $k \in [0, 1]$, the provisional score is the unique solution of

$$\bar{q}_{e,t} = \arg \min_{u \in (0,1)} \left[(1-k) K(u, \sigma(m_{e,t+1})) + k K(u, \pi_t(e)) \right]. \tag{40}$$

Indeed, the first derivative is $\mathrm{logit}(u) - (1-k)m_{e,t+1} - kz_{e,t}$, and the second derivative is $1/[u(1-u)] > 0$. Thus, the fixed logit update is the closed-form minimizer of a weighted sum of Bernoulli KL divergences. This characterizes the surrogate construction rather than introducing an additional training objective. Its association with downstream scores is examined separately below.

EO through soft positive membership. Appendix A.3 gives the auxiliary-label interpretation of the weighted EO contrast. Here we examine the intended role of its condition weights: whether w_t assigns greater relative weight to positive edges in the completed graph.

C.2 Empirical evidence for the two proxy roles

Protocol. We use a frozen Cora–EDGE-cond backbone with 256 reverse steps and fixed node features. The SP and EO shift schedules are calibrated separately for 100 optimizer steps, with k fixed within each configuration; both use the final checkpoint. We generate 10 independent graphs per policy (uncontrolled, T-SP, and T-EO) and record controlled pre-shift caches at generation progress 0.25, 0.50, and 0.90.

For each completed graph B, positives are split 80/10/10 and one GCN g_B is trained only on the training adjacency, then reused for all saved stages. The fixed evaluator has one layer, hidden dimension 128, training-only input dropout 0.1, Adam learning rate 0.01, zero weight decay, and at most 1,000 epochs with validation-AUC patience 5. No test outcomes are used for setting or checkpoint selection. Here EO concerns mean predicted probabilities on generated test positives.

Saved proxy scores and terminal GCN scores are compared at corresponding unordered node-pair IDs; intermediate diffusion graphs are not supplied to the GCN. The SP and EO diagnostics use different trajectories and evaluation supports, so their correlations are not a matched comparison between controllers.

Downstream ranking information in $\bar{q}_t$. We measure within-graph Spearman correlation between $\bar{q}_t$ and g_B. The SP diagnostic uses held-out test positives and sampled nonedges; the EO diagnostic uses held-out test positives only. At each recorded progress level, Table 3 reports the mean and sample standard deviation of the defined within-graph correlations.

At 90% progress, the correlations are 0.5811 for T-SP and 0.2584 for T-EO, showing positive late-stage ranking association on their respective supports. These observations support $\bar{q}_t$ as a source of downstream-relevant link-ranking information.

Table 3: Running-score agreement with the terminal GCN. Values are mean $\pm$ sample standard deviation of within-graph Spearman correlations. At 25%, $\bar{q}_t$ is constant on the measured support, so correlation is undefined. One T-EO graph is also constant at 50%.

Policy	Progress	$\mathrm{Spearman}(\bar{q}_t, g_B)$	Valid graphs
T-SP	0.25	Undefined	0/10
T-SP	0.50	0.0859 ± 0.0312	10/10
T-SP	0.90	0.5811 ± 0.0268	10/10
T-EO	0.25	Undefined	0/10
T-EO	0.50	0.0800 ± 0.0554	9/10
T-EO	0.90	0.2584 ± 0.0539	10/10

Generated-positive emphasis in w_t. Using the saved T-EO summaries, we retrospectively compare condition weights on actual positives and nonedges within each relation r. Define positive prevalence p_r, positive weight share $\alpha_{t,r}$, and enrichment relative to uniform weighting $L_{t,r}$:

$$p_r = \frac{\sum_{e \in \mathcal{U}^r} B_e}{|\mathcal{U}^r|}, \qquad \alpha_{t,r} = \frac{\sum_{e \in \mathcal{U}^r} w_{e,t} B_e}{\sum_{e \in \mathcal{U}^r} w_{e,t}}, \qquad L_{t,r} = \frac{\alpha_{t,r}}{p_r}. \tag{41}$$

Uniform weights give $L_{t,r} = 1$. We also report $\bar{w}_+ = \mathbb{E}_{\mathcal{U}^r}[w_t \mid B = 1]$ and $\bar{w}_- = \mathbb{E}_{\mathcal{U}^r}[w_t \mid B = 0]$. Each statistic is computed per graph before averaging.

This membership diagnostic uses all generated positives, including GCN training edges; B is used only retrospectively, not to construct the sampling-time weights.

At 90% progress, actual positives receive substantially larger mean weights than nonedges in both relations. Relative to uniform weighting, positive weight shares increase by $83.91\times$ and $198.45\times$ for same- and different-group pairs, respectively. Thus w_t carries generated-positive membership information and preferentially weights positive edges, matching its intended positive-emphasis role in the EO construction.

Interpretation. The two diagnostics provide complementary empirical support in this Cora pilot: late-stage $\bar{q}_t$ is associated with downstream link ranking, and w_t increases the relative contribution of generated positives. These are component-level findings about ranking and positive emphasis.

Table 4: EO condition-weight diagnostics on full $\mathcal{U}$; means over 10 T-EO graphs. α reports the absolute positive weight share, whereas L reports enrichment relative to uniform weighting. Mean positive prevalences are 0.5126% (same) and 0.03754% (different).

Progress	Relation	$\bar{w}_+$	$\bar{w}_-$	L	α (%)
0.25	Same	0.00144	0.00144	1.00	0.51
0.25	Different	0.00144	0.00144	1.00	0.04
0.50	Same	0.00278	0.00144	1.92	0.98
0.50	Different	0.00255	0.00144	1.77	0.07
0.90	Same	0.21094	0.00144	83.91	43.01
0.90	Different	0.21204	0.00099	198.45	7.45

D Experimental Setup and Reporting Protocol

Experiment groups. The SP and EO performance comparisons use the common downstream evaluation protocol and evaluation count specified in Appendix D.4, with a shared uncontrolled reference for each EDGE-cond dataset. The proxy diagnostics and timing study use their own settings: Appendix C uses ten graphs per policy and final checkpoints after 100 optimizer steps, while the timing study uses one graph per setting and is not a performance operating-point comparison. Appendix F specifies the EO-targeted controller configurations and checkpoint rules.

D.1 Datasets and Group Attributes

We evaluate the primary SP study, the EO extension, and the controller ablations on Cora (Sen et al., 2008), Citeseer (Yang et al., 2016), and Amazon Photo (Shchur et al., 2018). We select these datasets to facilitate comparison with prior fair graph-generation studies, following the benchmarks used in FairWire (Kose & Shen, 2024) and FAROS (Badaoui et al., 2025). Cora and Citeseer are citation networks; Amazon Photo is a product co-purchase network. Node attributes are used as input features and node categories are used as multi-valued group attributes, following the original fair graph-generation benchmark protocol. The grouping is a controlled benchmark construction, not an assertion that the categories represent legally protected demographic attributes. Table 5 summarizes the dataset statistics used in our experiments.

D.2 Backbones and Controller Variants

EDGE-cond. The primary model is a node-feature-conditioned sparse EDGE-style topology generator (Chen et al., 2023). It keeps the node population and features X fixed, corrupts only the topology, and uses an active-node selection stage followed by edge prediction. The denoiser uses a message-passing neural network (Gilmer et al., 2017), with node-feature embeddings concatenated into the backbone. The backbone is trained with its original reconstruction objective $\mathcal{L}_{\text{node}} + \mathcal{L}_{\text{edge}}$,

Table 5: Edge counts refer to the number of entries in the dataset edge list after removing self-loops and duplicate entries, rather than the number of unique unordered node pairs.

Dataset	Nodes	Edges	Features	Groups
Cora	2,708	10,556	1,433	7
Citeseer	3,327	9,228	3,703	6
Amazon Photo	7,650	238,162	745	8

without a fairness regularizer. The controller leaves selection unchanged and modifies the reverse edge logits.

FW-bb. The second model is a FairWire-style joint node-feature and edge diffusion backbone (Kose & Shen, 2024). For plug-in experiments, the original fairness regularizer is removed: $\lambda_{\mathrm{FW}} = 0$, with backbone objective $\mathcal{L}_X + \mathcal{L}_A$. Its node-feature denoising branch is unchanged and the controller acts only on the edge branch. The fixed-controller FW-bb experiments use $T = 8$; the separate $T = 3$ versus $T = 8$ comparison is reported in Appendix E.5.

FW-fc. This feature-conditioned FairWire-style variant keeps X fixed and generates only edges. It is a diagnostic transfer setting rather than the primary backbone.

Controllers. FAIRSHIFT-T freezes the denoiser, holds k constant, and learns only the per-step logit-shift strengths from replayed trajectories. FAIRSHIFT-F uses fixed (k, η) values evaluated in parameter sweeps and does not optimize a controller objective. Neither variant uses the FairWire fairness loss during backbone training. Uncontrolled rows disable reverse-logit steering. The explicit $\lambda_{\mathrm{FW}} > 0$ rows in Table 12 are separate training-time fairness baselines, not FAIRSHIFT runs.

D.3 Downstream Evaluation

For each generated topology B, we train a one-layer GCN link predictor (Kipf & Welling, 2017) g_B. Evaluation positives are held-out edges of B, and evaluation negatives are nonedges of B. AUC uses these generated-graph labels. The original node group labels s partition evaluation pairs into same-group and different-group relations, and SP is the absolute difference between their mean predictor scores without conditioning on edge existence. All primary entries are percentages. Sampling-time proxy values are not substituted for the downstream scores.

SP and score-based EO are both evaluated through predictors trained on generated topologies. EO additionally restricts the score comparison to held-out positive edges of the corresponding generated graph. Matched within-backbone comparisons keep the backbone checkpoint, diffusion steps, and evaluator settings fixed and apply the same pair-construction and splitting rules. Appendix F specifies the EO-targeted controller configurations and reports comparisons with the shared uncontrolled references.

D.4 Operating Points and Uncertainty

Selection of reported operating points. We manually selected a small number of evaluated configurations to illustrate contrasting utility–fairness trade-offs. The labels *ret.* and *fair.* describe the emphasis of these examples; they do not denote solutions to a prespecified constrained optimization problem or guarantee Pareto optimality. To make this reporting choice inspectable, the supplementary material includes all evaluated configurations, their identifiers and aggregate metrics that distinguish the displayed operating points from the remaining observations. Any plotted Pareto frontier is the empirical nondominated subset of the evaluated configurations. Reporting-point selection is distinct from checkpoint selection within each controller run and from tuning the downstream evaluator. To make this reporting choice visually inspectable, Appendix E.7 plots the evaluated FAIRSHIFT-T trade-off fronts for all three EDGE-cond datasets and highlights the reported ret. and fair. operating points.

Table 6: SP-targeted FAIRSHIFT-T settings on frozen EDGE-cond backbones. Ret. and fair. identify the reported retention-oriented and fairness-oriented operating points. AUC and SP are mean $\pm$ population standard deviation across eight generated graphs, multiplied by 100. LR denotes the controller learning rate.

Dataset	Point	AUC ↑	Δ_{SP} ↓	T	k	η_0	LR	w_{fair}	w_{util}	Best epoch
Cora	ret.	92.58 ± 2.06	13.78 ± 5.60	256	0.5	0.0075	0.001	100000	0.2	$500^{\dagger}$
Cora	fair.	87.52 ± 1.63	9.31 ± 2.43	256	0.7	0.015	0.001	100000	0.2	450
Citeseer	ret.	94.10 ± 0.56	14.58 ± 4.23	128	0.3	0.01	0.001	50000	0.2	$500^{\dagger}$
Citeseer	fair.	93.02 ± 1.26	9.25 ± 3.84	128	0.3	0.03	0.001	50000	0.3	500
Amazon Photo	ret.	96.45 ± 0.03	24.72 ± 0.73	512	0.2	0.1	0.01	500000	0.01	$578^{\dagger}$
Amazon Photo	fair.	95.69 ± 0.24	20.42 ± 0.61	512	0.12	0.12	0.01	500000	0.01	$125^{\dagger}$

w_{fair} and w_{util} weight the SP and utility-preservation losses, respectively. η_0 initializes the learned per-timestep shift schedule. k is not optimized. Best epoch is one-based and selected by minimum total controller calibration training loss. †: verified in reproductions matching the reported AUC/SP means and population standard deviations. Checkpoint epochs are reported on a one-based scale: a stored zero-based epoch e is displayed as $e + 1$.

Result provenance. Rows marked O report our experiments, whereas R denotes values taken from prior publications. FAROS is included using published results (Badaoui et al., 2025), since its implementation was unavailable to us. The other baseline methods follow the definitions of Kose & Shen (2024); their rerun configurations and evaluation settings are reported separately. For each configuration in our primary SP and EO performance studies, we use the same number of generated-graph evaluations, 8, and report the mean and standard deviation of the resulting downstream metrics. Each evaluation trains a downstream predictor on its generated graph. Results repeated in multiple tables use the same run-level aggregation. Controller-calibration seeds are recorded separately from graph-generation and downstream-evaluation repetitions.

D.5 COMPUTING ENVIRONMENT

The main SP study used Ubuntu 22.04.5 LTS and one NVIDIA H200 GPU with 144 GB of VRAM. Appendix G reports a separate timing study on NVIDIA RTX A6000 GPUs (48 GB per GPU), with its own settings and synchronized stage timers. These measurements are not assigned to the main performance-table configurations.

D.6 SP CONTROLLER SETTINGS AND REPRODUCTION PROTOCOL

The companion repository at `https://anonymous.4open.science/r/FairShift-A6CB` provides the configuration files, evaluated search grids, controller and sampling commands, evaluator settings, and a mapping from reported table rows to run identifiers. It also includes complete operating-point plots and instructions for reproducing the reported aggregations. Each run configuration specifies the fixed coefficient k, shift-strength initialization, trainable shift parameters, fairness and preservation weights, and checkpoint-selection rule.

Scope and configurations. Table 6 records the SP-targeted FAIRSHIFT-T configurations for the reported retention-oriented (ret.) and fairness-oriented (fair.) operating points on Cora, Citeseer, and Amazon Photo. The corresponding EDGE-cond backbones use $T = 256$, 128, and 512 reverse steps, respectively. The denoiser remains frozen throughout controller calibration and subsequent generation. Within each configuration, k is fixed, while the controller learns one shift-strength parameter per reverse step. The initialization η_0, controller learning rate, and loss weights are listed separately for each operating point. Table 7 specifies the reproduction protocols for the reported SP controller configurations on Cora, Citeseer, and Amazon Photo.

Calibration and checkpoint selection. For Cora and Citeseer, controllers are calibrated with Adam for 500 epochs using controller seed 0. The replay pool contains two graphs and is refreshed every 25 epochs, and eight graphs are generated for monitoring every 100 epochs. For the Amazon Photo configurations, Adam calibration uses 600 epochs, controller seed 0, one replay graph refreshed every 150 epochs, and monitoring with one graph at epoch 300. Both documented protocols

use normalized guidance and gradient clipping by value at 1. Within a controller run, the checkpoint is selected by the minimum total calibration training loss. This within-run checkpoint selection is distinct from the selection of the reported ret. and fair. operating points described in Appendix D.4. Table 6 reports the selected checkpoint epochs on a one-based scale.

Final generation. Fixed-logit replay is used for controller calibration, not as the source of raw predictions for final graph generation. For each evaluation graph, we initialize the backbone sampler and controller state and execute the reverse process online. At step t, the backbone applies its native candidate-selection procedure and evaluates raw edge logits on the current graph state A_t^ω. The controller shifts these logits before edge sampling. The resulting A_{t-1}^ω is carried into the next reverse step, where the backbone is evaluated again. When the native selector depends on the evolving graph state, earlier guided edge samples may indirectly affect later candidate sets; this is not a separate selection intervention. Recorded raw logits from an unguided trajectory are not substituted for these stepwise evaluations. Final generation produces eight graphs with sample batch size 1. Generation continues the training random-number-generator state without reseeding.

Table 7: Reproduction protocols for the reported SP controller configurations in Table 6.

Setting	Cora / Citeseer	Amazon Photo
Controller optimizer / training budget	Adam / 500 epochs	Adam / 600 epochs
Controller seed / evaluator seed	0 / 0	0 / 0
Gradient clipping	Value clipping at 1	Value clipping at 1
η schedule	One learned parameter per diffusion step	One learned parameter per diffusion step
k	Fixed during calibration and generation	Fixed during calibration and generation
Replay samples / refresh interval	2 graphs / 25 epochs	1 graph / 150 epochs
Guidance normalization	Enabled	Enabled
Monitoring	8 graphs every 100 epochs	1 graph at epoch 300
Checkpoint selection	Minimum total calibration training loss	Minimum total calibration training loss
Calibration protocol	Controller optimization on recorded trajectories	Controller optimization on recorded trajectories
Final generation protocol	Online reverse sampling; raw logits evaluated on the current graph	Online reverse sampling; raw logits evaluated on the current graph
Group attribute	Original node metadata `y`	Original node metadata `y`
LP evaluator	One-layer GCN encoder within a GAE link-prediction evaluator	One-layer GCN encoder within a GAE link-prediction evaluator
Generated-edge train/validation/test split	80/10/10	80/10/10
Evaluator hidden sizes	$\{16, 32, 128, 512\}$	$\{16, 32, 128, 512\}$
Evaluator learning rates	$\{0.03, 0.01, 0.003, 0.001\}$	$\{0.03, 0.01, 0.003, 0.001\}$
Evaluator nominal dropout	$\{0, 0.1, 0.2\}$	$\{0, 0.1, 0.2\}$
Evaluator selection	Validation AUC over 48 configurations	Validation AUC over 48 configurations
Evaluator epochs / patience	1000 / 5	1000 / 5
Backbone hidden dimension	128	160
Backbone heads	$[8, 8, 8, 8, 1]$	$[8, 8, 8, 8, 1]$
Backbone LR argument	0.0005	0.0001
Edge dropout	Cora: 0.05; Citeseer: 0	0
Batch size	Cora: 2; Citeseer: 4	2
Backbone checkpoint	Cora: `checkpoint_4999.pt`; Citeseer: `checkpoint_4399.pt`	`checkpoint_69999.pt`

Downstream evaluation. The evaluator is seeded once with seed 0, and the eight generated graphs are evaluated sequentially. For each graph, the one-layer GAE evaluator uses an 80/10/10 generated-edge training/validation/test split. Its configuration is selected by validation AUC over the 48 com-

Table 8: Complete external-reference comparison, including the diagnostic FW-fc operating points. R denotes published results; O denotes in-house results.

Method	Src.	AUC ↑	Δ_{SP} ↓
Cora			
Original graph G	O	94.92	27.71
GraphMaker-style synthetic graph	O	87.31 ± 1.11	35.56 ± 1.48
FairAdj	O	82.13 ± 1.07	15.47 ± 2.39
Adversarial	O	83.66 ± 5.64	16.35 ± 9.80
FairWire	O	86.81 ± 1.20	14.79 ± 5.22
FAROS-Prior	R	89.08 ± 2.72	13.30 ± 8.62
EDGE-cond + FAIRSHIFT-T (ret.)	O	92.58 ± 2.06	13.78 ± 5.60
EDGE-cond + FAIRSHIFT-T (fair.)	O	87.52 ± 1.63	9.31 ± 2.43
FW-fc + FAIRSHIFT-T (ret.)	O	92.56 ± 1.17	21.34 ± 5.14
FW-fc + FAIRSHIFT-T (fair.)	O	92.07 ± 0.99	14.07 ± 0.92
Citeseer			
Original graph G	O	95.76	29.05
GraphMaker-style synthetic graph	O	92.19 ± 1.06	37.56 ± 1.29
FairAdj	O	82.67 ± 2.78	15.45 ± 2.68
Adversarial	O	89.59 ± 2.70	24.20 ± 5.82
FairWire	O	91.47 ± 3.79	20.71 ± 6.32
FAROS-Uniform	R	84.45 ± 7.71	12.39 ± 8.05
EDGE-cond + FAIRSHIFT-T (ret.)	O	94.10 ± 0.56	14.58 ± 4.23
EDGE-cond + FAIRSHIFT-T (fair.)	O	93.02 ± 1.26	9.25 ± 3.84
FW-fc + FAIRSHIFT-T (ret.)	O	94.22 ± 0.78	16.76 ± 0.99
FW-fc + FAIRSHIFT-T (fair.)	O	93.03 ± 1.17	13.09 ± 1.97
Amazon Photo			
Original graph G	O	96.91	32.58
GraphMaker-style synthetic graph	O	94.45 ± 0.21	33.49 ± 0.28
Adversarial	O	94.24 ± 1.20	29.17 ± 2.83
FairWire	O	92.28 ± 0.35	26.22 ± 0.32
FAROS-Prior	R	94.10 ± 0.25	27.03 ± 0.36
EDGE-cond + FAIRSHIFT-T (ret.)	O	96.45 ± 0.03	24.72 ± 0.73
EDGE-cond + FAIRSHIFT-T (fair.)	O	95.69 ± 0.24	20.42 ± 0.61
FW-fc + FAIRSHIFT-T (ret.)	O	95.03 ± 0.08	28.85 ± 0.56
FW-fc + FAIRSHIFT-T (fair.)	O	94.87 ± 0.20	28.55 ± 1.33

binations of hidden size, learning rate, and nominal dropout listed in Table 7. Each evaluator configuration is trained for at most 1,000 epochs with patience 5. Group memberships are taken from the original node metadata `y`. AUC and Δ_{SP} are reported as mean $\pm$ population standard deviation across the eight generated-graph evaluations, with both statistics multiplied by 100. These deviations summarize graph-level evaluation variability, rather than variability across independently calibrated controllers.

E Complete SP Results and Additional Diagnostics

E.1 External Reference Points

Table 8 contains the complete external comparison, including the FW-fc trainable-controller rows omitted from the main performance table. These rows remain contextual comparisons rather than controlled evidence for a particular intervention.

Table 9: Fixed plug-in comparisons with no controller-training objective. Each controlled row shares its denoiser checkpoint with the corresponding uncontrolled row. No fairness loss is used during backbone training; FW-bb uses $\lambda_{\mathrm{FW}} = 0$.

Dataset	T	Controller	AUC ↑	Δ_{SP} ↓
EDGE-cond				
Cora	256	Uncontrolled	93.60 ± 0.07	26.07 ± 4.43
Cora	256	FAIRSHIFT-F	92.74 ± 0.21	24.85 ± 0.54
Citeseer	128	Uncontrolled	95.03 ± 0.07	26.83 ± 2.26
Citeseer	128	FAIRSHIFT-F	93.75 ± 0.31	24.50 ± 2.11
Amazon Photo	512	Uncontrolled	96.98 ± 0.01	32.35 ± 0.21
Amazon Photo	512	FAIRSHIFT-F	96.71 ± 0.08	29.71 ± 0.39
FW-bb				
Cora	8	Uncontrolled	97.42 ± 0.22	44.36 ± 0.51
Cora	8	FAIRSHIFT-F	95.40 ± 0.68	40.72 ± 1.17
Citeseer	8	Uncontrolled	79.17 ± 0.78	29.24 ± 0.70
Citeseer	8	FAIRSHIFT-F	77.21 ± 0.23	26.33 ± 0.31
Amazon Photo	8	Uncontrolled	97.82 ± 0.10	46.99 ± 0.76
Amazon Photo	8	FAIRSHIFT-F	97.09 ± 0.00	31.08 ± 0.58

E.2 FIXED PLUG-IN CONTROL

Table 9 isolates the fixed controller: only sampling-time edge logits change relative to each uncontrolled checkpoint. All six paired settings reduce the reported mean SP gap. The utility cost differs across datasets and settings; the experiment supports controllability, not uniform improvement in both metrics.

E.3 FIXED-SHIFT SWEEPS

Tables 10 and 11 report the evaluated fixed-controller sweeps. On Cora with EDGE-cond, increasing η from 0.005 to 0.015 at $k = 0.5$ changes AUC from 92.74 to 84.07 and SP from 24.85 to 4.79. The corresponding Citeseer endpoints are $93.75/24.50$ and $83.26/6.64$. On Amazon Photo, increasing η from 0.005 to 0.010 at fixed $k = 0.3$ changes AUC from 96.71 to 95.36 and SP from 29.71 to 21.47. Table 10 additionally reports a configuration with $k = 0.5$ and $\eta = 0.010$, which is separate from this fixed-k comparison. FW-bb exhibits analogous utility–fairness trade-offs, but its absolute operating points differ substantially from EDGE-cond.

E.4 TRAINING-TIME REGULARIZATION VERSUS SAMPLING-TIME CONTROL

Table 12 separates fairness-regularized backbone training from fixed sampling-time control. Larger training weights sometimes produce much lower SP, for example the Cora $\lambda_{\mathrm{FW}} = 50$ point at AUC 80.84 and SP 2.00. Those rows train a different denoiser. The FAIRSHIFT-F rows instead reuse a fairness-agnostic checkpoint. They address a different adaptation constraint and should not be treated as the same mechanism or a uniform dominance comparison.

E.5 DIAGNOSTIC FEATURE-CONDITIONED TRANSFER AND DIFFUSION STEPS

Table 13 retains the diagnostic FW-fc fixed-controller results. Table 14 shows why diffusion steps must be matched: on Cora, FW-bb AUC is 76.20 for $T = 3$ and 97.42 for $T = 8$, while SP also changes. A result at a different T therefore cannot isolate the effect of adding the controller.

E.6 COMPLETE TRAINABLE-CONTROLLER COMPARISON

Table 15 compares uncontrolled sampling, fixed schedules, and learned schedules on both backbones. The main results use these estimates for T/F comparison, without replacing their deviations

Table 10: Fixed-shift sweep on EDGE-cond. All results use downstream AUC and SP.

Controller	η	k	AUC ↑	Δ_{SP} ↓
Cora				
Uncontrolled	–	–	93.60 ± 0.07	26.07 ± 4.43
FAIRSHIFT-F	0.005	0.5	92.74 ± 0.21	24.85 ± 0.54
FAIRSHIFT-F	0.010	0.5	88.26 ± 0.18	16.20 ± 1.41
FAIRSHIFT-F	0.015	0.5	84.07 ± 0.63	4.79 ± 0.87
Citeseer				
Uncontrolled	–	–	95.03 ± 0.07	26.83 ± 2.26
FAIRSHIFT-F	0.005	0.5	93.75 ± 0.31	24.50 ± 2.11
FAIRSHIFT-F	0.010	0.5	86.93 ± 0.32	15.70 ± 0.67
FAIRSHIFT-F	0.015	0.5	83.26 ± 0.15	6.64 ± 0.58
Amazon Photo				
Uncontrolled	–	–	96.98 ± 0.01	32.35 ± 0.21
FAIRSHIFT-F	0.005	0.3	96.71 ± 0.08	29.71 ± 0.39
FAIRSHIFT-F	0.010	0.5	95.25 ± 0.29	22.05 ± 1.06
FAIRSHIFT-F	0.010	0.3	95.36 ± 0.10	21.47 ± 0.38

Table 11: Fixed-shift sweep on FW-bb with $\lambda_{\mathrm{FW}} = 0$ and $T = 8$. All results use downstream AUC and SP.

Controller	η	k	AUC ↑	Δ_{SP} ↓
Cora				
Uncontrolled	–	–	97.42 ± 0.22	44.36 ± 0.51
FAIRSHIFT-F	0.005	0.5	95.40 ± 0.68	40.72 ± 1.17
FAIRSHIFT-F	0.010	0.5	90.91 ± 0.32	32.95 ± 0.24
FAIRSHIFT-F	0.015	0.5	84.57 ± 0.37	22.37 ± 0.36
FAIRSHIFT-F	0.050	0.5	77.95 ± 0.34	2.20 ± 0.26
Citeseer				
Uncontrolled	–	–	79.17 ± 0.78	29.24 ± 0.70
FAIRSHIFT-F	0.005	0.5	77.21 ± 0.23	26.33 ± 0.31
FAIRSHIFT-F	0.015	0.5	74.12 ± 0.46	23.49 ± 0.25
FAIRSHIFT-F	0.050	0.5	66.87 ± 0.32	14.91 ± 1.01
Amazon Photo				
Uncontrolled	–	–	97.82 ± 0.10	46.99 ± 0.76
FAIRSHIFT-F	0.010	0.5	97.89 ± 0.07	46.65 ± 0.09
FAIRSHIFT-F	0.050	0.5	97.09 ± 0.00	31.08 ± 0.58

by those of the fixed-sweep tables. In particular, the EDGE-cond Citeseer fairness-oriented T result is AUC 93.02 ± 1.26 and SP 9.25 ± 3.84.

Additional AUC–SP trade-off visualizations. Figure 3 complements the Cora visualization in Figure 2 with results on Citeseer and Amazon Photo. Both panels compare FAIRSHIFT-F and FAIRSHIFT-T on the EDGE-cond backbone, alongside the displayed baseline comparisons. The calibrated operating points combine lower SP than the high-AUC fixed-strength settings with higher AUC than the low-SP fixed-strength settings, illustrating the utility–fairness trade-offs available through controller calibration. These plots summarize the evaluated operating points; the component ablations are reported separately in Appendix E.8.

Table 12: Training-time FairWire regularization versus sampling-time control. Rows with $\lambda_{\mathrm{FW}} > 0$ retrain the denoiser with a fairness objective; FAIRSHIFT-F rows keep a $\lambda_{\mathrm{FW}} = 0$ denoiser frozen.

Variant	λ_{FW}	AUC ↑	Δ_{SP} ↓
Cora			
FW-bb, no controller	0	97.42 ± 0.22	44.36 ± 0.51
FW-bb, no controller	50	80.84 ± 0.38	2.00 ± 0.24
FW-bb, no controller	100	82.17 ± 0.48	6.31 ± 0.25
FW-bb + FAIRSHIFT-F	0	95.40 ± 0.68	40.72 ± 1.17
FW-bb + FAIRSHIFT-F	0	84.57 ± 0.37	22.37 ± 0.36
Citeseer			
FW-bb, no controller	0	79.17 ± 0.78	29.24 ± 0.70
FW-bb, no controller	50	86.39 ± 0.13	7.20 ± 0.37
FW-bb, no controller	100	93.75 ± 0.15	10.58 ± 0.52
FW-bb + FAIRSHIFT-F	0	77.21 ± 0.23	26.33 ± 0.31
FW-bb + FAIRSHIFT-F	0	74.12 ± 0.46	23.49 ± 0.25
Amazon Photo			
FW-bb, no controller	0	97.82 ± 0.10	46.99 ± 0.76
FW-bb, no controller	50	87.56 ± 0.91	5.91 ± 4.23
FW-bb, no controller	100	86.70 ± 1.14	25.10 ± 1.83
FW-bb + FAIRSHIFT-F	0	97.89 ± 0.07	46.65 ± 0.09
FW-bb + FAIRSHIFT-F	0	97.09 ± 0.00	31.08 ± 0.58

Table 13: Fixed-controller transfer to FW-fc. This feature-conditioned FairWire-style setting is diagnostic.

Controller	AUC ↑	Δ_{SP} ↓
Cora		
Uncontrolled	86.29 ± 1.70	23.59 ± 6.72
FAIRSHIFT-F	85.88 ± 1.20	14.87 ± 1.79
Citeseer		
Uncontrolled	97.69 ± 0.74	27.00 ± 5.58
FAIRSHIFT-F	92.91 ± 0.78	15.02 ± 1.36
Amazon Photo		
Uncontrolled	95.34 ± 0.20	31.78 ± 0.97
FAIRSHIFT-F	94.90 ± 0.20	28.01 ± 0.75

E.7 SELECTION TRANSPARENCY FOR THE REPORTED FAIRSHIFT-T OPERATING POINTS

Figure 4 shows the evaluated FAIRSHIFT-T trade-off fronts on the EDGE-cond backbone. Each panel shows the AUC–SP trade-off over the evaluated FAIRSHIFT-T configurations for one dataset. The reported retention-oriented (ret.) and fairness-oriented (fair.) operating points are highlighted. These labels are descriptive and were selected post hoc to illustrate contrasting utility–fairness trade-offs; they are not solutions to a prespecified constrained optimization problem.

E.8 TEMPORAL BLENDING AND SHIFT-SCHEDULE ABLATIONS

Setup. We use the retention-oriented FAIRSHIFT-T configurations on frozen EDGE-cond backbones documented in Appendix D.6. The reference settings are $(T, k_{\mathrm{ref}}) = (256, 0.5)$ for Cora, $(128, 0.3)$ for Citeseer, and $(512, 0.2)$ for Amazon Photo. For each dataset, we compare the stepwise shift schedule with a directly calibrated shared shift strength, and evaluate both parameterizations

Table 14: Diffusion-step sensitivity of FW-bb. The result motivates keeping T fixed in paired plug-in comparisons. No uncertainty estimate is available for the Cora $T = 3$ row.

Dataset	T	AUC ↑	Δ_{SP} ↓
Cora	3	76.20	28.16
Cora	8	97.42 ± 0.22	44.36 ± 0.51
Citeseer	3	75.94 ± 1.66	29.80 ± 2.51
Citeseer	8	79.17 ± 0.78	29.24 ± 0.70
Amazon Photo	3	94.79 ± 0.18	44.55 ± 1.55
Amazon Photo	8	97.82 ± 0.10	46.99 ± 0.76

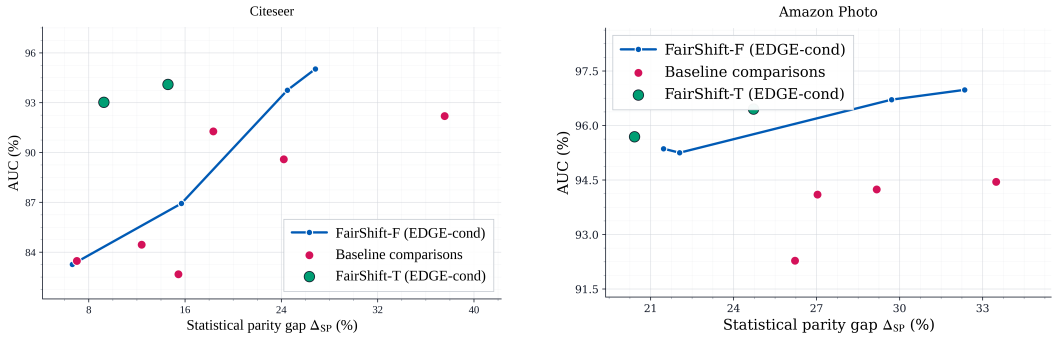
Table 15: Complete controller comparison on EDGE-cond and FW-bb. Only the schedule is learned by FAIRSHIFT-T; the denoiser remains frozen. Ret. and fair. preserve the reported operating-point labels.

Controller	AUC ↑	Δ_{SP} ↓
EDGE-cond / Cora		
Uncontrolled	93.60 ± 0.07	26.07 ± 4.43
FAIRSHIFT-F	92.74 ± 0.21	24.85 ± 0.54
FAIRSHIFT-T (ret.)	92.58 ± 2.06	13.78 ± 5.60
FAIRSHIFT-T (fair.)	87.52 ± 1.63	9.31 ± 2.43
EDGE-cond / Citeseer		
Uncontrolled	95.03 ± 0.07	26.83 ± 2.26
FAIRSHIFT-F	93.75 ± 0.31	24.50 ± 2.11
FAIRSHIFT-T (ret.)	94.10 ± 0.56	14.58 ± 4.23
FAIRSHIFT-T (fair.)	93.02 ± 1.26	9.25 ± 3.84
EDGE-cond / Amazon Photo		
Uncontrolled	96.98 ± 0.01	32.35 ± 0.21
FAIRSHIFT-F	96.71 ± 0.08	29.71 ± 0.39
FAIRSHIFT-T (ret.)	96.45 ± 0.03	24.72 ± 0.73
FAIRSHIFT-T (fair.)	95.69 ± 0.24	20.42 ± 0.61
FW-bb / Cora		
Uncontrolled	97.42 ± 0.22	44.36 ± 0.51
FAIRSHIFT-F	95.40 ± 0.68	40.72 ± 1.17
FAIRSHIFT-T (ret.)	93.25 ± 0.53	22.63 ± 1.31
FAIRSHIFT-T (fair.)	92.29 ± 0.29	18.62 ± 1.39
FW-bb / Citeseer		
Uncontrolled	79.17 ± 0.78	29.24 ± 0.70
FAIRSHIFT-F	77.21 ± 0.23	26.33 ± 0.31
FAIRSHIFT-T (ret.)	75.12 ± 2.67	23.39 ± 2.41
FAIRSHIFT-T (fair.)	72.06 ± 3.51	5.56 ± 3.87
FW-bb / Amazon Photo		
Uncontrolled	97.82 ± 0.10	46.99 ± 0.76
FAIRSHIFT-F	97.09 ± 0.00	31.08 ± 0.58
FAIRSHIFT-T (ret.)	97.55 ± 0.40	28.57 ± 3.70
FAIRSHIFT-T (fair.)	96.90 ± 0.45	22.84 ± 2.98

with $k = k_{\mathrm{ref}}$ and $k = 1$. All four configurations are calibrated separately with their specified fixed k and shift parameterization.

The shared-strength comparator is directly calibrated as a single parameter, following Appendix B.3, rather than obtained by averaging a learned stepwise schedule. For the schedule comparison, the fixed update coefficient, controller objective, and calibration and selection budgets are matched. Unlike FAIRSHIFT-F, the shared-strength comparator undergoes controller optimization. Setting $k = 1$ replaces temporal blending by last-value caching: active entries overwrite their previous states, whereas inactive entries remain cached. Thus, this ablation does not remove the persistent pair pool.

Each configuration is evaluated on eight generated graphs using the downstream link-prediction protocol in Appendix D.3. We report AUC and Δ_{SP} in percent. The reported deviations summarize variability across generated-graph evaluations, rather than across independent controller-calibration runs.



(a) Citeseer (b) Amazon Photo

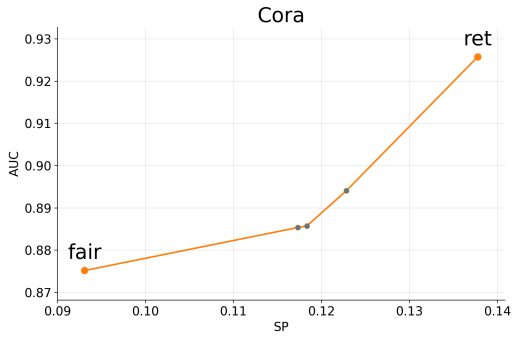
Figure 3: Additional AUC–SP trade-offs on Citeseer and Amazon Photo. All FAIRSHIFT-F and FAIRSHIFT-T results in these panels use the EDGE-cond backbone. Blue markers and connecting lines show the uncontrolled reference and evaluated FAIRSHIFT-F settings; green markers show selected calibrated FAIRSHIFT-T configurations; magenta markers show baseline comparison points. Higher AUC and lower Δ_{SP} (toward the upper left) are preferred. Lines are visual guides between evaluated settings, not an exhaustive Pareto frontier.

Stepwise versus shared shifts. At the reference coefficient k_{ref}, the stepwise schedule achieves higher mean AUC and lower mean SP on Cora: AUC increases from 89.35 to 92.58, while SP decreases from 15.69 to 13.78. On Citeseer, it reduces SP from 16.01 to 14.58, with a small AUC decrease from 94.18 to 94.10. On Amazon Photo, it increases AUC from 96.17 to 96.45 and reduces SP from 25.45 to 24.72. Thus, stepwise shifts attain lower mean SP on all three datasets at the reference coefficients, with the largest observed scheduling benefit on Cora. Table 16 reports the graph-level variability.

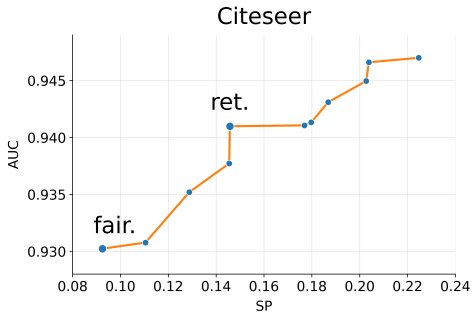
Table 16: Component ablations on frozen EDGE-cond backbones across three datasets. The reference settings are $(T, k_{\mathrm{ref}}) = (256, 0.5)$ for Cora, $(128, 0.3)$ for Citeseer, and $(512, 0.2)$ for Amazon Photo. Values are mean $\pm$ standard deviation over eight generated-graph evaluations, in percent. The first row repeats the FAIRSHIFT-T (ret.) reference. Bold marks the best observed mean for each dataset and metric.

		Cora		Citeseer		Amazon Photo	
k	Shift schedule	AUC ↑	Δ_{SP} ↓	AUC ↑	Δ_{SP} ↓	AUC ↑	Δ_{SP} ↓
k_{ref}	Stepwise	**92.58** $\pm$ 2.06	**13.78** $\pm$ 5.60	94.10 $\pm$ 0.56	**14.58** $\pm$ 4.23	96.45 $\pm$ 0.03	**24.72** $\pm$ 0.73
1	Stepwise	91.86 $\pm$ 0.76	19.54 $\pm$ 5.51	93.72 $\pm$ 0.63	22.72 $\pm$ 2.28	**96.48** $\pm$ 0.21	28.08 $\pm$ 0.83
k_{ref}	Shared	89.35 $\pm$ 0.85	15.69 $\pm$ 5.01	**94.18** $\pm$ 0.86	16.01 $\pm$ 5.56	96.17 $\pm$ 0.27	25.45 $\pm$ 0.42
1	Shared	91.96 $\pm$ 1.00	22.66 $\pm$ 5.09	94.08 $\pm$ 0.50	22.28 $\pm$ 3.28	96.28 $\pm$ 0.04	27.85 $\pm$ 0.05

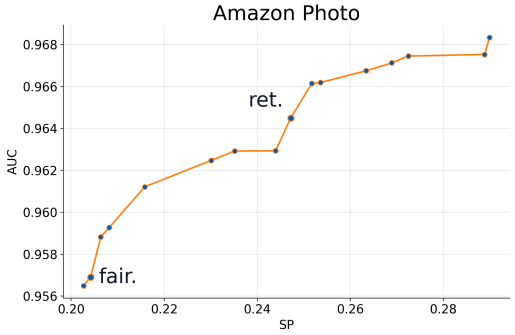
Temporal blending versus last-value caching. With stepwise shifts, replacing k_{ref} by $k = 1$ increases the mean SP gap by 5.76, 8.14, and 3.36 points on Cora, Citeseer, and Amazon Photo, respectively. Mean AUC decreases by 0.72 and 0.38 points on Cora and Citeseer, but increases slightly by 0.03 points on Amazon Photo. With shared shifts, $k = 1$ also produces larger mean SP gaps on all three datasets, although it attains higher mean AUC on Cora and Amazon Photo. Thus, temporal blending is associated with lower mean SP under both evaluated shift parameterizations across all three datasets, while its utility effect depends on the configuration.



(a) Cora



(b) Citeseer



(c) Amazon Photo

Figure 4: Empirical AUC–SP trade-off fronts of FAIRSHIFT-T on EDGE-cond. Each point is an evaluated controller configuration. The highlighted ret. and fair. markers denote the operating points reported in Table 6. They are descriptive selections from the evaluated trade-off space, not solutions chosen by a prespecified AUC constraint.

F ADDITIONAL SCORE-BASED EQUAL-OPPORTUNITY EXPERIMENTS

F.1 EVALUATION SCOPE AND SELECTION STATUS

We additionally evaluate score-based EO control on Cora, Citeseer and Amazon Photo using the frozen `EDGE-cond` backbone for each dataset. Each configuration uses the same number of generated-graph evaluations as the primary SP study (Appendix D.4), with the common one-layer GCN link predictor and an 80/10/10 generated-edge training/validation/test split. EO conditions on the held-out positive edges of each generated graph.

For a generated topology B, let g_B be the trained evaluator and let $\mathcal{P}_B^{r,+}$ contain its held-out generated positive pairs of relation $r \in \{=, \neq\}$. We report

$$\Delta_{\text{EO}}^{\text{score}}(B) = \left| \frac{\sum_{e \in \mathcal{P}_B^{=,+}} g_B(e)}{|\mathcal{P}_B^{=,+}|} - \frac{\sum_{e \in \mathcal{P}_B^{\neq,+}} g_B(e)}{|\mathcal{P}_B^{\neq,+}|} \right|. \tag{42}$$

This preserves positive-class conditioning (Hardt et al., 2016) but compares mean scores rather than thresholded true-positive rates. AUC, SP, and score-based EO in Table 17 are computed using the

same downstream evaluator for each generated graph. C0, S0, and A0 reuse the corresponding uncontrolled EDGE-cond evaluations reported in Table 15. SP- and EO-targeted controllers are compared against these shared uncontrolled references.

Comparison scope. We compare uncontrolled sampling, EO-targeted FairShift-F, EO-targeted FairShift-T, and SP-targeted FairShift-T. Within each dataset, matched comparisons keep the backbone checkpoint, diffusion-step count, evaluator settings, and pair-construction and splitting rules fixed. The uncontrolled condition establishes the reference utility and disparity. The fixed and learned EO-targeted configurations examine the effect of controller calibration, while the SP-targeted FairShift-T configuration reveals how optimizing SP affects EO. Comparing the two targets therefore characterizes their different AUC–SP–EO trade-offs within the evaluated controller configurations.

Table 17: Score-based EO comparisons on `EDGE-cond`. Values are mean ± standard deviation over the same number of generated-graph evaluations used in the primary SP study (Appendix D.4), in percent. C1–C3, S1–S3 and A1–A3 are EO-targeted FairShift-T configurations. The remaining rows report uncontrolled sampling, EO-targeted FairShift-F, and primary SP-targeted FairShift-T. The uncontrolled AUC and SP results in C0, S0, and A0 repeat the corresponding EDGE-cond results in Table 15.

ID	Descriptor	AUC ↑	Δ_{SP} ↓	$\Delta_{\mathrm{EO}}^{\mathrm{score}}$ ↓
Cora				
C0	Uncontrolled	93.60 ± 0.07	26.07 ± 4.43	10.05 ± 0.28
C-F	FairShift-F (EO)	92.40 ± 0.08	24.07 ± 0.47	8.97 ± 0.50
C-SP	FairShift-T (SP targeted)	92.58 ± 2.06	13.78 ± 5.60	5.17 ± 2.37
C1	High AUC	92.40 ± 0.87	22.06 ± 5.08	7.21 ± 2.37
C2	Intermediate	91.21 ± 1.45	20.47 ± 3.30	4.68 ± 2.02
C3	Low EO	89.46 ± 1.66	13.15 ± 4.40	4.15 ± 1.90
Citeseer				
S0	Uncontrolled	95.03 ± 0.07	26.83 ± 2.26	6.80 ± 0.56
S-F	FairShift-F (EO)	93.96 ± 0.43	21.85 ± 1.59	5.46 ± 0.27
S-SP	FairShift-T (SP targeted)	94.10 ± 0.56	14.58 ± 4.23	3.19 ± 1.69
S1	High AUC	95.48 ± 0.85	24.96 ± 3.00	6.12 ± 0.90
S2	Intermediate	95.01 ± 0.57	24.15 ± 1.56	4.67 ± 1.45
S3	Low EO	94.68 ± 0.50	18.49 ± 5.34	3.00 ± 1.49
Amazon Photo				
A0	Uncontrolled	96.98 ± 0.01	32.35 ± 0.21	7.97 ± 0.52
A-F	FairShift-F (EO)	96.25 ± 0.18	25.58 ± 0.81	4.80 ± 0.22
A-SP	FairShift-T (SP targeted)	96.45 ± 0.03	24.72 ± 0.73	3.42 ± 0.04
A1	High AUC	96.13 ± 0.10	24.61 ± 0.25	2.86 ± 1.41
A2	Intermediate	94.91 ± 0.16	17.99 ± 0.54	2.72 ± 0.46
A3	Low EO	94.75 ± 0.09	18.01 ± 0.28	2.26 ± 0.33

Descriptors indicate the emphasis of the displayed EO-targeted FairShift-T configurations within each dataset. EO is evaluated on held-out positive edges from each generated graph.

F.2 Backbone and Controller Configuration

All configurations use node features and retain a frozen EDGE-cond backbone. The Cora runs use `cora_checkpoint_4999.pt` with 256 reverse steps, the Citeseer runs use `citeseer_checkpoint_4399.pt` with 128 reverse steps, and the Amazon Photo EO runs use `checkpoint_69999.pt` with 512 reverse steps. The EO-targeted trainable configurations are C1–C3 for Cora, S1–S3 for Citeseer, and A1–A3 for Amazon Photo. The Cora and Citeseer configurations use diffusion dimension 128, controller seed 0, and 500 optimization epochs. The EO-targeted Amazon Photo configurations use diffusion dimension 160, controller seed 0, and 300

optimization epochs. Each configuration fixes k and learns only the shift schedule. Table 18 lists the fixed coefficients, shift initializations, learning rates, fairness and preservation weights, and selected checkpoint epochs.

Samples use the selected controller checkpoint from each run, rather than necessarily the checkpoint at the final optimization epoch. Checkpoint selection within a run is distinct from the post-hoc selection of configurations for reporting.

Table 18: Configuration metadata for the selected EO points. Every Cora and Citeseer run uses 500 controller epochs, controller seed 0, diffusion dimension 128, and node features. Amazon Photo runs use 300 controller epochs. The final column is the recorded best-checkpoint epoch on a one-based scale; evaluated samples come from `controller_best`, not necessarily the final optimization epoch.

ID	k	η_0	LR	w_{EO}	w_{util}	Best epoch
C1	0.3	0.01	0.001	1	0.05	500
C2	0.5	0.01	0.0005	1	0.05	430
C3	0.7	0.01	0.001	10	0.05	430
S1	0.5	0.001	0.0005	1	0.05	410
S2	0.3	0.005	0.0005	1	0.1	290
S3	0.3	0.01	0.0005	10	0.05	247
A1	0.2	0.1	0.01	10	0.05	300
A2	0.7	0.035	0.001	10	0.01	300
A3	0.7	0.05	0.001	10	0.01	296

Additional comparison configurations. Uncontrolled sampling disables reverse-logit steering and has no controller-training stage. EO-targeted FairShift-F uses fixed $(k, \eta) = (0.5, 0.005)$ on Cora, $(k, \eta) = (0.3, 0.005)$ on Citeseer and $(k, \eta) = (0.5, 0.005)$ on Amazon Photo, without controller optimization. C-SP, S-SP, and A-SP use the primary SP-targeted FAIRSHIFT-T retention-oriented configurations reported in Table 6. The calibration settings and checkpoint epochs listed in Table 18 apply to the EO-targeted trainable configurations identified in that table.

F.3 SELECTED UTILITY–FAIRNESS OPERATING POINTS

Table 17 reports nine EO-targeted FairShift-T configurations together with uncontrolled, fixed EO-targeted, and SP-targeted comparisons. These configurations illustrate how the presence of guidance, controller calibration, and the target fairness criterion affect downstream utility and disparity.

Improvement relative to uncontrolled sampling. The shared uncontrolled Cora reference attains AUC 93.60 and a score-based EO gap of 10.05. EO-targeted FAIRSHIFT-F reduces the EO gap to 8.97, while C1 reduces it to 7.21; both attain AUC 92.40, corresponding to a 1.20-percentage-point AUC decrease. On Citeseer, the uncontrolled AUC/EO pair is $95.03/6.80$. EO-targeted FAIRSHIFT-F attains $93.96/5.46$, whereas S1 attains $95.48/6.12$, combining a smaller mean EO gap with a 0.45-point increase in mean AUC. S3 further reduces EO to 3.00 at AUC 94.68, a 0.35-point decrease from the uncontrolled reference. On Amazon Photo, the uncontrolled AUC/EO pair is $96.98/7.97$. EO-targeted FAIRSHIFT-F attains $96.25/4.80$, whereas A1 attains $96.13/2.86$, corresponding to a 64.1% relative EO reduction from uncontrolled sampling with a 0.85-percentage-point AUC decrease. Thus, the displayed fixed and learned EO-targeted configurations reduce mean EO disparity relative to the shared uncontrolled references, with configuration-dependent changes in AUC.

Fixed versus learned EO control. On Cora, C1 achieves an EO gap 1.76 percentage points lower than the fixed EO-targeted controller, with the same reported mean AUC. On Citeseer, S2 and S3 achieve both higher mean AUC and lower mean EO disparity than the displayed fixed controller: AUC increases from 93.96 to $95.01/94.68$, while the EO gap decreases from 5.46 to $4.67/3.00$, respectively. On Amazon Photo, A1 reduces the EO gap from the fixed controller's 4.80 to 2.86,

with a small AUC decrease from 96.25 to 96.13. These comparisons show the benefit of the reported learned schedules relative to the evaluated fixed settings, without implying that every learned configuration dominates every fixed configuration.

Effect of the guidance target. SP-targeted FairShift-T reduces the mean EO gap from 10.05 to 5.17 on Cora, from 6.80 to 3.19 on Citeseer, and from 7.97 to 3.42 on Amazon Photo. On Cora, C-SP achieves higher mean AUC and lower mean SP and EO gaps than C1. C2 and C3 attain smaller EO gaps of 4.68 and 4.15, respectively, with lower AUC than C-SP. On Citeseer, S-SP achieves a smaller EO gap than S1 and S2, whereas S3 achieves higher mean AUC and a smaller mean EO gap than S-SP, but a larger mean SP gap. On Amazon Photo, A1 attains a lower mean EO gap than A-SP (2.86 versus 3.42), with lower AUC (96.13 versus 96.45) and a slightly lower mean SP gap (24.61 versus 24.72). Thus, SP-targeted guidance provides cross-metric improvements, while EO-targeted guidance reaches lower EO gaps at some operating points.

Additional EO reduction and utility cost. The displayed configurations exhibit different utility costs for additional EO reduction. On Cora, moving from C1 to C3 reduces mean AUC from 92.40 to 89.46, while the EO gap changes from 7.21 to 4.15. On Citeseer, moving from S1 to S3 reduces the mean EO gap from 6.12 to 3.00, while AUC decreases from 95.48 to 94.68. On Amazon Photo, moving from A1 to A3 reduces mean EO from 2.86 to 2.26, while AUC decreases from 96.13 to 94.75. These comparisons motivate reporting EO jointly with utility rather than selecting configurations solely by their smallest observed EO gap.

The reported means and standard deviations use the same number of generated-graph evaluations as the primary SP study, as specified in Appendix D.4. For C1–C3, S1–S3 and A1–A3, these statistics summarize repeated evaluations of the controller checkpoints specified in Table 18. The comparisons above describe the reported means and their graph-level variability.

F.4 Positive Support and Interpretation

For the nine configurations in the positive-support diagnostics, the EO-defined fraction is 1.0. The minimum same-group and different-group positive counts are 150 and 89 on Cora, 229 and 108 on Citeseer, and 3203 and 2345 on Amazon Photo. Both positive relations are therefore represented in every evaluation covered by these diagnostics, so the score-based EO gap is numerically defined.

The exported `last_surrogate_abs_gap` is a last-epoch value, whereas downstream scores come from the best checkpoint. These fields are not paired observations for proxy validation, which requires matching checkpoint and graph identifiers.

Table 19: Positive-support diagnostics. The two minimum counts are minima across the generated-graph evaluations, not group totals. The last two columns are mean predicted scores on positive pairs, expressed as percentages.

ID	Defined fraction	$\min n^{=,+}$	$\min n^{\neq,+}$	Mean $g^{=,+}$	Mean $g^{\neq,+}$
C1	1.00	295	89	80.96	73.75
C2	1.00	238	96	80.95	76.27
C3	1.00	150	105	73.34	69.18
S1	1.00	299	114	87.96	81.84
S2	1.00	246	117	88.82	84.15
S3	1.00	229	108	80.76	77.77
A1	1.00	4355	2345	88.95	86.10
A2	1.00	3229	3183	85.89	83.17
A3	1.00	3203	3299	85.72	83.46

G WALL-CLOCK COST OF POST-TRAINING FAIRNESS ADAPTATION

G.1 TIMING SETUP

We measure the cost of applying FAIRSHIFT to existing FairWire checkpoints rather than retraining the diffusion backbone. The study uses the standard attributed backbone with $T = 8$ and $\alpha_A = \alpha_X = 0$, not the feature-only variant. For each of Cora, Citeseer, and Amazon Photo, both controller variants reuse the same checkpoint and generate one graph with seed 0. Each of the six configurations was timed once using an NVIDIA RTX A6000 with 48 GB of memory and four PyTorch/BLAS CPU threads.

FAIRSHIFT-F uses normalized guidance with $k = 0.3$ and $\eta = 0.25$ and requires no controller fitting. The timed FAIRSHIFT-T configuration learns only the shift schedule $\{\eta_t\}_{t=1}^{8}$, initialized at 0.25, while fixing $k = 0.3$. With $T = 8$, this corresponds to eight trainable shift parameters. Calibration runs for all 500 epochs with Adam, learning rate 10^{-3}, and fairness weight $50{,}000$ and utility-preservation weight 0.3. One replay is used per epoch and refreshed every 100 epochs, giving five replay generations during calibration. The best controller-loss checkpoint is then used for generation and evaluation. The denoiser remains frozen throughout. Sampling uses a batch size of $32{,}768$ and four DataLoader workers.

Both variants use the same downstream GAE evaluator: 48 configurations (four learning rates, four hidden sizes, and three dropout rates), one encoder layer, at most 1,000 epochs per configuration, patience 5, batch size $16{,}384$, and an 80/10/10 edge split. Thus the timings include a downstream evaluator search but not a search over FairShift configurations. The settings were chosen for timing, not to reproduce the selected performance points in the main tables. Different graph densities and early-stopping epochs can produce different evaluation costs even under this common evaluator.

G.2 MEASURED APPLICATION COST AND TRAINING REFERENCES

Table 20 contrasts the cost of using an existing checkpoint with available backbone-training references. Including controller calibration, generation, and AUC/SP evaluation, FAIRSHIFT-T takes approximately 6.3, 11.6, and 28.6 minutes on Cora, Citeseer, and Amazon Photo, respectively. Without controller fitting, FAIRSHIFT-F takes approximately 1.3, 1.4, and 6.6 minutes. These are additional one-run costs after pretraining; neither run includes a new diffusion-backbone training stage.

Table 20: One-run FairShift costs and contextual FairWire training times. FairShift-T includes 500 controller epochs with fixed k; both variants include one generated graph and downstream AUC/SP evaluation. E denotes a batch-extrapolated reference and L a historical run log.

	FairWire reference	Measured one-run cost	
Dataset	(hours)	FAIRSHIFT-T (fixed k, min:s)	FAIRSHIFT-F (fixed k, min:s)
Cora	10.80^{E}	06:20	01:18
Citeseer	7.71^{L}	11:38	01:25
Amazon Photo	50.39^{E}	28:38	06:38

The reference column has mixed provenance. Cora and Amazon Photo are extrapolated from ten warmed-up training batches and five validation batches to stored checkpoint epochs 546 and 3,219, respectively. They are not completed-training wall-clock measurements; the actual stopping epochs are unknown. The estimates omit per-epoch worker startup and shuffling, checkpoint saving, and W&B communication/logging, and approximate a partial final batch as a full batch. The Citeseer reference is a completed 673-epoch run under the same basic $T = 8$ setting, with logged time $27{,}767.5$ s.

G.3 STAGE-WISE ACCOUNTING AND INTERPRETATION

Table 21 separates setup, controller fitting and I/O, graph generation, and downstream evaluation. The measured FAIRSHIFT-T fitting blocks are 312.3, 604.8, and 1,336.3 s. They include replay

refreshes, forward/backward computation, optimization, logging, checkpoint operations, and associated setup/reloading; they are not pure gradient-computation times. A last checkpoint is saved every epoch and a best checkpoint whenever the controller loss improves.

Table 21: Stage-wise wall-clock times in seconds. T learns only η_t and F uses fixed guidance. Graph-stage accounting differs between the two variants as described in the text; totals are reported directly. Rounding can produce a 0.1-second discrepancy in row sums.

Dataset	Variant	Setup/ other	Fitting/ I/O	Graph	AUC/SP evaluation	Total
Cora	T	6.7	312.3	46.8	14.6	380.4
	F	7.4	0.0	44.4	26.3	78.1
Citeseer	T	8.0	604.8	74.3	11.0	698.2
	F	9.0	0.0	66.9	9.3	85.2
Amazon Photo	T	6.9	1,336.3	332.2	42.5	1,717.9
	F	7.7	0.0	326.7	63.6	398.0

The total is the sum of the generation-process and evaluation-process lifetimes, including imports, model loading, saving, and shutdown; brief parent-process handling between them is excluded. For T, the graph stage includes online reverse sampling with the calibrated controller, PyG conversion, and file writing. For F, the graph timer covers the sampler only, with conversion and saving charged to setup/other. Sampling already included in export is not counted twice. Evaluation times also differ because generated edge counts and early-stopping epochs differ despite the shared evaluator.

These measurements characterize a minutes-scale application workflow using a pretrained backbone, compared with hours-scale contextual training references. Each entry is a single observation including instrumentation and shared-resource effects, not a matched speedup estimate. Broader controller tuning and persistent-state memory remain separate profiling questions.