LAB-Tab: LLM-Augmented Bayesian Network Adaptation for Few-Shot Tabular Generation

Zijian Shen, Taijie Chen, Bin Zhou, Ziyang Jiang, Jintao Ke*

Department of Civil Engineering, The University of Hong Kong

arXiv:2608.01879v1 [cs.LG] 3 Aug 2026

Abstract

Tabular data generation supports analysis and decision-making when target-domain data are scarce, yet collecting complete target samples is often costly. A practical but under-explored setting provides only a few target records together with richer source data from a related domain. Existing few-shot tabular generators often either fit sparse target statistics directly, which can overfit incidental patterns, or reuse source-domain generators, which may preserve dependencies that no longer hold in the target domain. To address this problem, we propose **LAB-Tab**, an LLM-augmented Bayesian network (BN) adaptation framework for source-aware few-shot tabular generation. LAB-Tab first fits a BN from source data and then uses an LLM to propose plausible target-domain BN edges that are absent from the source BN graph. This step converts semantic and weak statistical evidence into explicit structural hypotheses, thereby expanding the editable edge space beyond the source-fitted graph. Because the proposed edges may be noisy and interact with existing dependencies, a PPO policy calibrates edges in the augmented BN through edge-level actions, including *keep*, *weaken*, *strengthen*, *flip*, and *deactivate*. The PPO policy is trained with a reward that combines distributional alignment, downstream utility, and preservation of target-relevant dependencies. The adapted BN is then sampled to synthesize target-domain tables. Across six source–target distribution-shift scenarios built from three US Census (ACS) prediction tasks, **LAB-Tab** achieves the best performance at the 10% target-data budget, leads four of the six individual scenarios, and reduces the macro Overall score by 33.8% relative to the strongest baseline. It also obtains the best macro JSD, WAPE, and UtilityGap while maintaining competitive feature–label preservation.

1 Introduction

Tabular data generation supports data analysis, privacy-preserving sharing, simulation, and benchmark construction across domains such as healthcare, finance, transportation, and public-sector analytics. In many real-world applications, however, obtaining a representative target-domain table is difficult. Collecting sufficient target-domain records, together with the annotations required for downstream tasks, is often costly and time-consuming (Zhuang et al. 2021). Even when some records are accessible, they may be sparse, incomplete, or affected by missing values (Dong and Peng 2013; Emmanuel et al. 2021). Consequently, practitioners may have access to only a small number of target-domain records. In this low-data regime, both marginal distributions and multivariate dependencies must be estimated under substantial uncertainty, making it difficult to learn a target-domain generator without overfitting incidental patterns. This practical challenge motivates the problem of few-shot tabular generation.

Although target-domain data are scarce in this setting, a substantially larger table from a related source domain is often available. The source table may originate from an earlier survey, another geographic region, or a broader population, and can provide more reliable statistical information than the few observed target records. However, source-domain information cannot be transferred indiscriminately. As illustrated in Figure 1, distribution shift may affect both column marginals and conditional relationships among variables (Quiñonero-Candela et al. 2009; Koh et al. 2021). Some dependencies from the source data may remain useful, while others may change dramatically within the target-specific relationships. For example, education may remain associated with income across regions, whereas the relationship between working hours and income may vary with local labor markets. The central challenge is therefore to exploit reliable source-domain knowledge while accommodating target-specific changes supported by only limited evidence.

Existing tabular generators, transfer-learning methods, and recent LLM-based approaches offer complementary capabilities in distribution modeling, cross-domain reuse, and semantic guidance (Solatorio and Dupriez 2023; Shi et al. 2025; Yang et al. 2025; Chen et al. 2026). However, they do not jointly support explicit source-to-target dependency adaptation and feedback-driven calibration under limited target evidence. To address this gap, we propose **LAB-Tab**, an LLM-augmented Bayesian network (BN) adaptation framework for source-aware few-shot tabular generation. LAB-Tab first learns a BN from the source table, using its explicit structure and conditional distributions as a stable and inspectable prior. Then, LAB-Tab utilizes an LLM to propose target-relevant edges absent from the source graph, expanding the structural search space through sparse target data. Finally, PPO calibrates these dependencies using target-domain data, resolving inaccurate edges before sampling. This design allows transferable source dependencies to be retained while introducing structural changes based on few-shot target-specific dataset.

*Corresponding author.

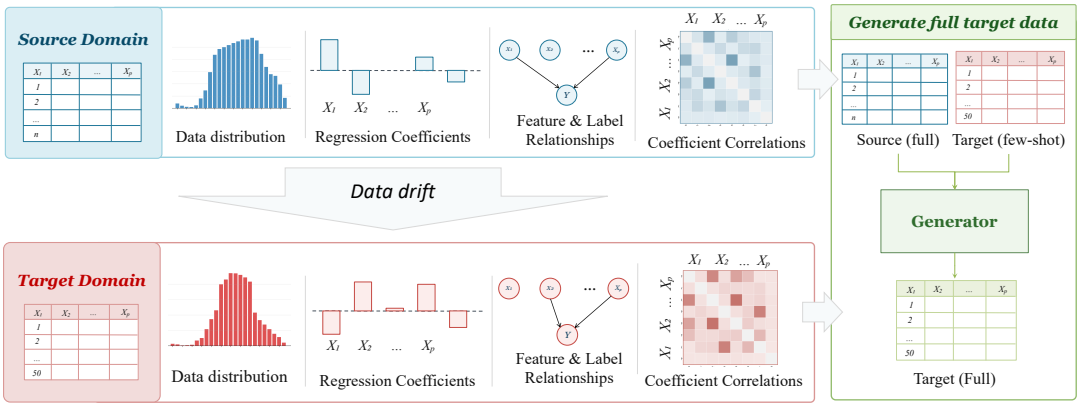

Figure 1: Source-aware few-shot tabular generation under distribution shift. A large source table and a small target subset are used to generate a full synthetic target table. The target domain may differ from the source in marginal distributions, predictive coefficients, feature–label relationships, and cross-feature associations.

Our contributions are summarized as follows:

- We introduce **LAB-Tab**, a source-aware few-shot tabular generation framework that combines a source-fitted BN, LLM-guided candidate-edge augmentation, and PPO-based dependency calibration.
- We formulate source-to-target adaptation as explicit BN edge control. The LLM expands the source-derived structural search space, deterministic validation enforces graph validity, and PPO adjusts edge activation, magnitude, and polarity using complementary global table-level and local dependency-level feedback.
- We evaluate LAB-Tab on six source–target shifts derived from three 2018 American Community Survey (ACS) prediction tasks. At the 10% target-data budget, LAB-Tab achieves the lowest macro Overall score, leads four of the six scenarios, and obtains the best macro JSD, WAPE, and UtilityGap. The ablations indicate complementary contributions from candidate-edge augmentation, PPO control, and local dependency feedback within the evaluated design.

2 Related Work

Tabular synthesis spans explicit probabilistic and high-capacity neural generators (Shi et al. 2025; Challagundla et al. 2025; Stoian, Giunchiglia, and Lukasiewicz 2025). Bayesian-network, SDV-style, and copula methods offer efficient sampling with inspectable dependency assumptions (Zhang et al. 2014; Patki, Wedge, and Veeramachaneni 2016; Asghar et al. 2019), whereas GAN and diffusion models increase expressiveness (Xu et al. 2019; Kotelnikov et al. 2023; Zhang et al. 2024). Most remain target-table driven. With few target records, neural models may fit unstable patterns, while probabilistic models must estimate local conditionals from sparse counts. When a related source table is available, the problem becomes generative domain adaptation. Conventional transfer learning improves target prediction through representation transfer, reweighting, distribution matching, or fine-tuning, but does not explicitly determine which dependencies a target generator should retain, introduce, or recalibrate (Zhuang et al. 2021).

LLM-based methods add schema semantics and in-context reasoning to tabular synthesis. GReaT, REaLTabFormer, and Tabula generate records from serialized rows or relations (Borisov et al. 2023; Solatorio and Dupriez 2023; Zhao, Birke, and Chen 2023), while subsequent methods improve example selection, efficiency, logical consistency, or dependency guidance (Xu et al. 2025; Fang et al. 2025; Nguyen et al. 2025; Long, Xu, and Brintrup 2025). In particular, SPADA uses an LLM-induced dependency graph to guide synthesis (Yang et al. 2025). Feedback-based learning and LLM-guided search further demonstrate the value of external evaluation for refining model outputs or proposed solutions (Christiano et al. 2017; Ouyang et al. 2022; Romera-Paredes et al. 2024; Ma et al. 2024). Taken together, prior work largely treats semantic structure induction, probabilistic generation, and feedback-driven adaptation as separate directions. Our proposed method bridges this gap by unifying these complementary capabilities in a framework for source-aware few-shot tabular generation.

3 Problem Formulation

Let $\mathcal{X} = \{X_1, \ldots, X_p\}$ denote the features, Y the task label, and $\mathcal{V} = \mathcal{X} \cup \{Y\} = \{V_1, \ldots, V_{p+1}\}$ the shared source–target schema. A record is $\mathbf{z} = (\mathbf{x}, y)$. We observe an abundant source table $\mathcal{D}_s = \{\mathbf{z}_i^s\}_{i=1}^{n_s}$ drawn i.i.d. from P_s and a few-shot target table $\mathcal{D}_t^{\text{fs}} = \{\mathbf{z}_i^t\}_{i=1}^{K}$ drawn i.i.d. from P_t, where $K \ll n_s$. The two distributions may differ in both marginals and conditional dependencies. During adaptation, P_t is unknown and held-out target records are used only for evaluation. Let $\mathcal{B}(\mathcal{V})$ denote the space of BN generators over $\mathcal{V}$. A source-aware adaptation rule produces

$$B^\star = \mathcal{A}\left(\mathcal{D}_s, \mathcal{D}_t^{\text{fs}}\right) \in \mathcal{B}(\mathcal{V}), \qquad \widehat{\mathcal{D}}_t = \{\widehat{\mathbf{z}}_i\}_{i=1}^{N}, \quad \widehat{\mathbf{z}}_i \overset{\text{i.i.d.}}{\sim} P_{B^\star}. \tag{1}$$

A BN is $B = (\mathcal{G}, \Theta)$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed acyclic graph and $\Theta = \{\theta_j\}_{j=1}^{p+1}$ contains its local conditional

distributions. For $V_j \in \mathcal{V}$, let $\mathrm{Pa}_{\mathcal{G}}(V_j) = \{V_i : (V_i, V_j) \in \mathcal{E}\}$ and $\theta_j(v_j \mid \mathbf{pa}_j) = P_B(V_j = v_j \mid \mathrm{Pa}_{\mathcal{G}}(V_j) = \mathbf{pa}_j)$. The induced joint distribution factorizes as

$$P_B(\mathbf{z}) = \prod_{j=1}^{p+1} \theta_j(v_j \mid \mathbf{pa}_j). \tag{2}$$

The target-aligned generator should balance distributional fidelity, predictive utility, and feature–label preservation. Let $D_{\mathrm{dist}}(P_B, P_t) \geq 0$ denote distributional discrepancy. For a fixed training size n_u, let $U_{n_u}(Q; P_t)$ denote the expected target performance of a fixed learner trained on samples from Q, and let $\mathbf{a}(Q)$ be the coefficient vector of a fixed linear probe under Q. Define

$$\begin{aligned} \Delta_U(B) &= \left| U_{n_u}(P_B; P_t) - U_{n_u}(P_t; P_t) \right|, \\ \Delta_A(B) &= 1 - \frac{\mathbf{a}(P_B)^\top \mathbf{a}(P_t)}{\|\mathbf{a}(P_B)\|_2 \|\mathbf{a}(P_t)\|_2}. \end{aligned} \tag{3}$$

The population objective and its ideal solution are

$$\begin{aligned} \mathcal{L}_t(B) &= \lambda_{\mathrm{d}} D_{\mathrm{dist}}(P_B, P_t) + \lambda_{\mathrm{u}} \Delta_U(B) + \lambda_{\mathrm{a}} \Delta_A(B), \\ B^\dagger &\in \arg\min_{B \in \mathcal{B}(\mathcal{V})} \mathcal{L}_t(B), \end{aligned} \tag{4}$$

where the nonnegative weights are not all zero. Since P_t is unknown, the practical objective is to construct $B^\star$ from $\mathcal{D}_s$ and $\mathcal{D}_t^{\mathrm{fs}}$ whose induced distribution approximates P_t under these three criteria.

4 Methodology

Figure 2 summarizes LAB-Tab, which adapts a source-fitted BN using the few-shot target set $\mathcal{D}_t^{\mathrm{fs}}$. The framework consists of source BN initialization, LLM-guided edge augmentation, and PPO-guided edge-control optimization. The adapted BN is then ancestrally sampled to generate the synthetic target table. The label variable is denoted by $Y \in \mathcal{V}$ throughout.

4.1 Source BN Initialization

We first learn a source BN $B_s = (\mathcal{G}_s, \Theta_s)$ from the source-domain table $\mathcal{D}_s$. Each variable is represented by a finite state space, with continuous variables discretized before structure and parameter learning.

Structure learning. Let $\mathcal{G}_s = (\mathcal{V}, \mathcal{E}_s)$ denote the source DAG. Starting from an empty graph, we perform greedy hill climbing under the Bayesian Information Criterion (BIC). At each step, the search considers single-edge additions, deletions, and reversals, and applies the valid operation that yields the largest positive BIC improvement. A candidate operation is valid only if it preserves acyclicity and keeps the in-degree of every node below $d_{\max}$. The search terminates when no valid operation improves the score. Since hill climbing is local,

$$\mathcal{G}_s \approx \arg\max_{\mathcal{G} \in \mathrm{DAG}_{d_{\max}}(\mathcal{V})} \mathrm{BIC}(\mathcal{G}; \mathcal{D}_s), \tag{5}$$

where $\mathrm{DAG}_{d_{\max}}(\mathcal{V})$ contains DAGs over $\mathcal{V}$ with maximum in-degree $d_{\max}$.

Parameter estimation. For each variable V_j, let $\mathcal{P}_j^s = \mathrm{Pa}_{\mathcal{G}_s}(V_j)$ denote its parent set, r_j the number of states of V_j, and q_j the number of joint configurations of $\mathcal{P}_j^s$. For a state $v \in \mathrm{Val}(V_j)$ and a parent configuration $\mathbf{pa}_j \in \mathrm{Val}(\mathcal{P}_j^s)$, we estimate the source CPT using a symmetric Dirichlet prior:

$$\widehat{\theta}^{(s)}_{j,v|\mathbf{pa}_j} = \frac{n_s(v, \mathbf{pa}_j) + \alpha/(r_j q_j)}{n_s(\mathbf{pa}_j) + \alpha/q_j}, \tag{6}$$

where $n_s(v, \mathbf{pa}_j)$ is the number of source records satisfying $V_j = v$ and $\mathcal{P}_j^s = \mathbf{pa}_j$, and $n_s(\mathbf{pa}_j) = \sum_v n_s(v, \mathbf{pa}_j)$. The hyperparameter α denotes the equivalent sample size. Collecting the CPTs of all variables gives Θ_s and hence the source BN is derived as $B_s = (\mathcal{G}_s, \Theta_s)$.

4.2 LLM-Guided BN Edge Augmentation

The source BN captures dependencies supported by $\mathcal{D}_s$, but target-relevant dependencies may be absent from $\mathcal{G}_s$ under domain shift. Relearning the entire DAG from the few-shot target set $\mathcal{D}_t^{\mathrm{fs}}$ is unreliable because many candidate relations receive limited empirical support. LAB-Tab therefore retains the source graph as a structural anchor and uses an LLM to propose a small set of candidate edge additions.

LLM-guided edge proposal. The augmentation context $\mathcal{C}_{\mathrm{LLM}}$ contains the source graph $\mathcal{G}_s$, the shared schema and variable descriptions, source-domain summaries $\mathcal{S}_s$, and few-shot target summaries $\mathcal{S}_t^{\mathrm{fs}}$. The summaries include category support, marginal frequencies, pairwise associations, and label-conditional contrasts. Given this context, the LLM returns

$$\widetilde{\mathcal{P}}_{\mathrm{LLM}} = \Phi_{\mathrm{LLM}}(\mathcal{C}_{\mathrm{LLM}}) = \left\{ p_m = \left(\widetilde{e}_m, c_m^{\mathrm{LLM}}, \ell_m \right) \right\}_{m=1}^{L}, \tag{7}$$

where $\widetilde{e}_m = (V_{u_m} \to V_{v_m})$ is a proposed directed edge, $c_m^{\mathrm{LLM}} \in [0, 1]$ is a self-assessed confidence score, and ℓ_m is a concise semantic and statistical rationale. The confidence score and rationale are retained as proposal metadata for subsequent calibration and auditability.

Graph validation. The proposed edges are processed by

$$\mathcal{E}^+ = \mathrm{ValidAdd}\left(\{\widetilde{e}_m\}_{m=1}^{L}, \mathcal{G}_s \right). \tag{8}$$

The validator rejects edges with endpoints outside $\mathcal{V}$, self-loops, duplicate or existing edges, and additions that would create a directed cycle or cause a node's in-degree to exceed $d_{\max}$. Each candidate is checked against the graph containing all previously accepted additions. The resulting augmented graph is

$$\begin{aligned} \mathcal{E}_{\mathrm{aug}} &= \mathcal{E}_s \cup \mathcal{E}^+, \\ \mathcal{G}_{\mathrm{aug}} &= (\mathcal{V}, \mathcal{E}_{\mathrm{aug}}). \end{aligned} \tag{9}$$

Because the accepted additions alter the parent sets of their child variables, we re-estimate the conditional distributions from $\mathcal{D}_s$ under $\mathcal{G}_{\mathrm{aug}}$ using the same Dirichlet estimator. Let $\bar{\theta}^{(s)}_{j,v|\mathbf{pa}_j}$ denote the resulting source conditional probability. We then combine this source estimate with the few-shot target counts:

$$\theta^{\mathrm{aug}}_{j,v|\mathbf{pa}_j} = \frac{n_t^{\mathrm{fs}}(v, \mathbf{pa}_j) + \kappa \bar{\theta}^{(s)}_{j,v|\mathbf{pa}_j}}{n_t^{\mathrm{fs}}(\mathbf{pa}_j) + \kappa}, \tag{10}$$

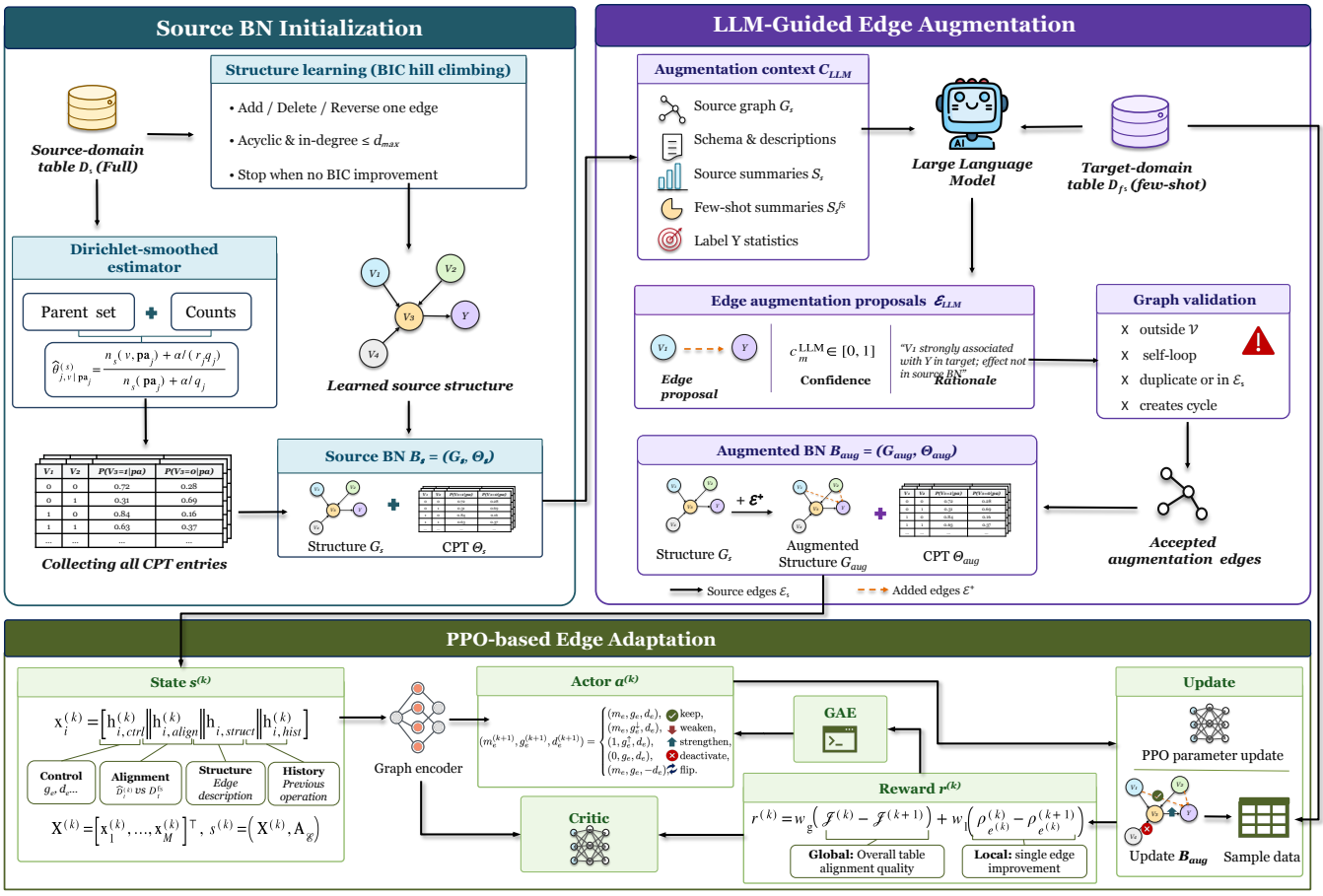

Figure 2: Overview of LAB-Tab: source BN learning and Conditional Probability Table (CPT) estimation, LLM-guided edge augmentation with graph validation, and PPO-based edge control with global–local rewards.

where $\kappa > 0$ is the source-informed prior concentration. Equivalently, $\bar{\theta}^{(s)}_{j,\cdot|\mathbf{pa}_j}$ specifies the prior mean and κ controls its strength relative to the few-shot target observations. Collecting these entries gives Θ_{aug} and the augmented BN

$$B^{\text{aug}} = (\mathcal{G}_{\text{aug}}, \Theta_{\text{aug}}), \tag{11}$$

which is subsequently calibrated by the edge-control policy.

4.3 PPO-Guided Edge-Control Optimization

The augmented BN favors structural coverage over exact target calibration. We therefore formulate edge control as a finite-horizon Markov decision process and optimize it with PPO (Schulman et al. 2017). For each edge $e \in \mathcal{E}_{\text{aug}}$, let $m_e \in \{0, 1\}$, $g_e \in [g_{\min}, g_{\max}]$, and $d_e \in \{-1, +1\}$ denote its activation, magnitude, and polarity, respectively, where $0 < g_{\min} < 1 \le g_{\max}$. Their combined control at step k is

$$\gamma_e^{(k)} = m_e^{(k)} d_e^{(k)} g_e^{(k)}. \tag{12}$$

Let $\mathcal{I}_j$ denote the incoming edges of V_j, and let x_e be the state of the parent endpoint of e. The controls parameterize the CPT as

$$\theta^{(k)}_{j,v|\mathbf{pa}_j} = \text{softmax}_v \left[b_j(v) + \sum_{e \in \mathcal{I}_j} \gamma_e^{(k)} \psi_e(v, x_e) \right], \tag{13}$$

where b_j is a base potential and ψ_e is the contribution of edge e. Collecting the CPTs gives $B^{(k)} = (\mathcal{G}_{\text{aug}}, \Theta^{(k)})$.

For an accepted LLM edge, define

$$\bar{g}_e = g_{\min} + (1 - g_{\min}) c_e^{\text{LLM}}. \tag{14}$$

The initial controls are

$$(m_e^{(0)}, g_e^{(0)}, d_e^{(0)}) = \begin{cases} (1, 1, 1), & e \in \mathcal{E}_s, \\ (1, \bar{g}_e, 1), & e \in \mathcal{E}^+. \end{cases} \tag{15}$$

We initialize the base and edge potentials in Eq. (13) to obtain the initial controlled BN $B^{(0)}$ by approximating B^{aug} under the controlled parameterization. Each episode starts from $B^{(0)}$ and runs for T steps, where the policy updates one edge control at each step and evaluates the resulting rollout; edge controls are reset between episodes, while the actor and critic parameters are retained.

State. Let $M = |\mathcal{E}_{\text{aug}}|$ and index the edges as $e_i = (V_{u_i} \to V_{v_i})$, $i = 1, \ldots, M$. At step k, the state representation associated with edge e_i is

$$\mathbf{x}_i^{(k)} = \left[\mathbf{h}^{(k)}_{i,\text{ctrl}} \| \mathbf{h}^{(k)}_{i,\text{align}} \| \mathbf{h}_{i,\text{struct}} \| \mathbf{h}^{(k)}_{i,\text{hist}} \right], \tag{16}$$

where the four blocks encode the current controls, generated–target discrepancies, structural information, and operation history, respectively. Let

$$\mathbf{X}^{(k)} = \left[\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_M^{(k)}\right]^\top, \quad s^{(k)} = \left(\mathbf{X}^{(k)}, \mathbf{A}_{\mathcal{E}}\right), \quad (17)$$

where $\mathbf{A}_{\mathcal{E}}$ connects two edge representations when their BN edges share a variable. A graph encoder aggregates information among connected edges; the actor scores valid edge–operation pairs, and the critic estimates the state value.

Action and transition. The policy samples

$$a^{(k)} = (e^{(k)}, o^{(k)}) \sim \pi_\phi(\cdot \mid s^{(k)}), \quad (18)$$

where

$$\mathcal{O} = \{\text{keep}, \text{weaken}, \text{strengthen}, \text{deactivate}, \text{flip}\}. \quad (19)$$

For $0 < \eta_- < 1 < \eta_+$, define

$$g_e^{\downarrow} = \max\{g_{\min}, \eta_- g_e\},\ g_e^{\uparrow} = \min\{g_{\max}, \eta_+ g_e\}. \quad (20)$$

For brevity, the quantities on the right-hand side below denote the controls at step k. The selected operation applies

$$(m_e^{(k+1)}, g_e^{(k+1)}, d_e^{(k+1)}) = \begin{cases} (m_e, g_e, d_e), & \text{keep}, \\ (m_e, g_e^{\downarrow}, d_e), & \text{weaken}, \\ (1, g_e^{\uparrow}, d_e), & \text{strengthen}, \\ (0, g_e, d_e), & \text{deactivate}, \\ (m_e, g_e, -d_e), & \text{flip}. \end{cases} \quad (21)$$

Deactivation preserves magnitude and polarity so that *strengthen* can reactivate the edge. The *flip* operation reverses the polarity of the edge-specific log-potential but does not change the graph direction $V_u \to V_v$. Operations that cannot alter the current controls are masked, while *keep* remains available. After each action, Eq. (13) yields $B^{(k+1)}$; we sample N_{roll} records and recompute the dynamic state features.

Reward. The global rollout loss combines distributional fidelity, downstream utility, and feature–label preservation:

$$\mathcal{J}^{(k)} = \lambda_{\text{d}} \widehat{D}_{\text{dist}}^{(k)} + \lambda_{\text{u}} \widehat{\Delta}_U^{(k)} + \lambda_{\text{a}} \widehat{\Delta}_A^{(k)}. \quad (22)$$

All terms compare $\widehat{\mathcal{D}}_t^{(k)}$ with $\mathcal{D}_t^{\text{fs}}$. For an edge $e = (V_u \to V_v)$, its local conditional discrepancy is

$$\rho_e^{(k)} = \frac{1}{2} \sum_{x \in \text{Val}(V_u)} \widehat{p}_t(x) \sum_{v \in \text{Val}(V_v)} \left| \widehat{p}_k(v \mid x) - \widehat{p}_t(v \mid x) \right|, \quad (23)$$

where $\widehat{p}_k$ and $\widehat{p}_t$ are smoothed estimates from the current rollout and few-shot target table. The step reward is

$$r^{(k)} = w_{\text{g}} \left(\mathcal{J}^{(k)} - \mathcal{J}^{(k+1)} \right) + w_{\text{l}} \left(\rho_{e^{(k)}}^{(k)} - \rho_{e^{(k)}}^{(k+1)} \right), \quad (24)$$

where the global term evaluates the complete generated table and the local term attributes improvement to the selected edge.

PPO update and synthesis. PPO uses generalized advantage estimation and the clipped actor objective, while the critic is trained on the corresponding returns. Entropy regularization encourages exploration, and a target KL threshold limits policy drift. Across all rollouts, we retain the control configuration $(\mathbf{m}^\star, \mathbf{g}^\star, \mathbf{d}^\star)$ with the lowest global loss. It defines

$$\mathcal{E}^\star = \{e \in \mathcal{E}_{\text{aug}} : m_e^\star = 1\}, \mathcal{G}^\star = (\mathcal{V}, \mathcal{E}^\star), \quad \Theta^\star = \Theta(\mathbf{m}^\star, \mathbf{g}^\star, \mathbf{d}^\star). \quad (25)$$

The resulting BN $B^\star = (\mathcal{G}^\star, \Theta^\star)$ is used to independently sample the N synthetic target records specified in the problem formulation.

5 Experiments

5.1 Experimental Setup

Data and scenarios. We construct six source–target shifts from three 2018 ACS prediction tasks in Folktables (Ding et al. 2021). The regression scenarios are INC-PRSD (ACSIncome, Puerto Rico → South Dakota) and INC-EDU (ACSIncome, non-bachelor → bachelor-or-above in California). The binary-classification scenarios are COV-TXCA (ACSPublicCoverage, Texas → California), COV-EDU (ACSPublicCoverage, non-bachelor → bachelor-or-above in California), MOB-PRCA (ACSMobility, Puerto Rico → California), and MOB-SEC (ACSMobility, private-sector → government workers in California). For each scenario, we reserve 20% of the target records for testing. The remaining records form the target training set used for the main experiment and the sensitivity analysis as provided in Appendix A.

Metrics. All metrics are computed against held-out real target data. **JSD** is the mean marginal Jensen–Shannon divergence, and define

$$\textbf{WAPE} = \frac{\sum_i |y_i - \hat{y}_i|}{\sum_i |y_i|}, \quad (26)$$

where y_i is the held-out target label and $\hat{y}_i$ is the predicted income for INC-PRSD and INC-EDU, or the predicted positive-class probability for the four classification scenarios. Let f_r and f_s denote identical predictors trained on the real few-shot and synthetic target data, respectively. We define

$$\textbf{UtilityGap} = |U(f_r; \mathcal{D}_t^{\text{test}}) - U(f_s; \mathcal{D}_t^{\text{test}})|, \quad (27)$$

where U is R^2 for regression and ROC-AUC for classification. Because the predictor architecture and test set are fixed, UtilityGap measures how closely synthetic training data reproduce the predictive utility of real few-shot target data. After applying a shared one-hot encoding to categorical features and standardizing numerical features, we compute

$$\textbf{Coef–Cos} = \frac{\boldsymbol{\beta}_r^\top \boldsymbol{\beta}_s}{\|\boldsymbol{\beta}_r\|_2 \, \|\boldsymbol{\beta}_s\|_2}, \quad (28)$$

where $\boldsymbol{\beta}_r$ and $\boldsymbol{\beta}_s$ are the coefficient vectors of linear probes fitted to real and synthetic target data. JSD, WAPE, and UtilityGap are lower-is-better, whereas Coef–Cos is higher-is-better. Let J, W, G, and C denote the four metrics for one scenario and seed. We define

$$\textbf{Overall} = \frac{1}{4}\left(\frac{J}{\ln 2} + \frac{W}{1+W} + \frac{G}{1+G} + \frac{1-C}{2} \right). \quad (29)$$

(a) Overall by scenario (↓)

Method	INC-PRSD	INC-EDU	COV-TXCA	COV-EDU	MOB-PRCA	MOB-SEC	Avg.
Gaussian Copula	0.3200	0.3364	0.0729	0.0858	0.1096	0.2584	0.1972 ± 0.0082
MTabGen	0.1483	**0.1069**	0.0608	0.0599	0.2619	0.2231	0.1435 ± 0.0087
CTGAN	0.2200	0.1578	0.0767	0.0862	0.1318	0.1573	0.1383 ± 0.0069
TVAE	0.1770	0.2750	0.1082	0.0809	0.1743	**0.0882**	0.1506 ± 0.0073
CopulaGAN	0.2149	0.1311	0.0682	0.0693	0.3094	0.1189	0.1520 ± 0.0079
SPADA	0.1673	0.1230	0.2131	0.2056	0.1143	0.2059	0.1715 ± 0.0170
LAB-Tab	**0.1367**	0.1184	**0.0243**	**0.0430**	**0.0837**	0.1437	$\mathbf{0.0916 \pm 0.0058}$

(b) Macro-averaged component metrics over 6 scenarios

Method	JSD ↓	WAPE ↓	Utility Gap ↓	Coef–Cos ↑
Gaussian Copula	0.0195 ± 0.0003	0.3061 ± 0.0024	0.3986 ± 0.0120	0.3783 ± 0.0065
MTabGen	0.0213 ± 0.0006	0.2753 ± 0.0039	0.0751 ± 0.0045	0.4740 ± 0.0069
CTGAN	0.0265 ± 0.0007	0.3097 ± 0.0075	0.1458 ± 0.0074	0.6816 ± 0.0053
TVAE	0.0492 ± 0.0034	0.3568 ± 0.0097	0.1556 ± 0.0351	$\mathbf{0.6919 \pm 0.0147}$
CopulaGAN	0.0332 ± 0.0017	0.3259 ± 0.0058	0.1185 ± 0.0076	0.5620 ± 0.0062
SPADA	0.0167 ± 0.0006	0.2317 ± 0.0040	0.1381 ± 0.0074	0.2747 ± 0.0154
LAB-Tab	$\mathbf{0.0016 \pm 0.0003}$	$\mathbf{0.1600 \pm 0.0024}$	$\mathbf{0.0728 \pm 0.0037}$	0.6464 ± 0.0458

Table 1: Results at the 10% target-data budget. Panel (a) reports scenario-level Overall scores, and Panel (b) reports macro-averaged component metrics. Values are averaged over five seeds; macro results are mean ± standard deviation. Scenario-level standard deviations are provided in Appendix F. Best values are bold.

Overall is bounded in $[0, 1]$ and is lower-is-better.

Baselines and implementation. We compare CopulaGAN (Patki, Wedge, and Veeramachaneni 2016), CTGAN (Xu et al. 2019), TVAE (Xu et al. 2019), Gaussian Copula (Asghar et al. 2019), MTabGen (Villaizán-Vallelado et al. 2025), and SPADA (Yang et al. 2025). All methods receive the same source data and few-shot target subset and are evaluated on the same held-out target split. The BN is learned by BIC hill climbing with $d_{\max} = 4$, $\alpha = 10$, and $\kappa = 1$. PPO runs for 150 episodes on regression scenarios and 250 episodes on classification scenarios, with 20 steps per episode, a clip ratio of 0.2, and a learning rate of 3×10^{-4}. Candidate edges are proposed using *gpt-4o*. The 10% main-table results are averaged over five random seeds, and uncertainty denotes the empirical standard deviation across seeds, more implementation details are given in Appendix B and C.

5.2 Main Results

Comparison at the 10% few-shot budget. Table 1 shows that LAB-Tab achieves the lowest macro Overall score, 0.0916 ± 0.0058, which is 33.8% lower than CTGAN, the strongest baseline at 0.1383 ± 0.0069. LAB-Tab ranks first in four of the six scenarios, while MTabGen and TVAE obtain the lowest Overall scores on INC-EDU and MOB-SEC, respectively. This advantage is supported by the best macro JSD, WAPE, and UtilityGap, together with competitive Coef–Cos. The baselines exhibit sharper trade-offs: SPADA provides the strongest baseline distributional alignment but preserves less downstream utility and feature–label association; MTabGen approaches LAB-Tab in UtilityGap but remains weaker in distributional fidelity and task-specific error; and TVAE and CTGAN obtain higher Coef–Cos but incur larger discrepancies on the other components. LAB-Tab's leading Overall score therefore reflects consistent performance across different objectives.

Method	JSD ↓	WAPE ↓	UGap ↓	Coef. ↑	Overall ↓
Source BN	0.0318	0.3137	0.1861	0.5280	0.1633
Target BN	**0.0011**	0.1815	0.1550	0.4137	0.1382
Source BN + MAP	**0.0011**	0.1641	0.0923	0.6082	0.1003
w/o LLM	0.0015	0.1659	0.0929	0.4478	0.1211
w/o PPO	0.0013	0.1626	0.0766	0.5803	0.0997
Greedy edits	0.0015	0.1707	0.0995	0.5536	0.1087
w/o local reward	0.0016	**0.1584**	0.0865	0.4892	0.1137
LAB-Tab	0.0016	0.1600	**0.0728**	**0.6464**	**0.0916**

UGap denotes Utility Gap and Coef. denotes Coef–Cos.

Table 2: Ablation results averaged over five random seeds and six drift scenarios.

Sensitivity to the target-data fraction. Figure 3 shows that LAB-Tab has the lowest macro Overall at seven of the eleven target-data fractions and the lowest average score across the sweep. Its advantage is concentrated in the low-data regime: at 1%, LAB-Tab scores 0.1499 versus 0.3299 for CTGAN, a 54.6% reduction. The gap narrows as target evidence grows, and MTabGen leads at 40%, 60%, 90%, and 100%; even at the full-data endpoint, however, LAB-Tab remains close (0.1751 versus 0.1650). This crossover is consistent with source-informed structural adaptation being most valuable when target observations are scarce, whereas MTabGen benefits more from direct target evidence. LAB-Tab's stable trajectory further shows that its few-shot advantage does not come with a sharp loss of competitiveness at larger budgets.

5.3 Ablation Study

Ablation settings. All variants use identical 10% target splits, seeds, and evaluation protocol. Source BN reuses the source-fitted graph and parameters, Target BN is fitted en-

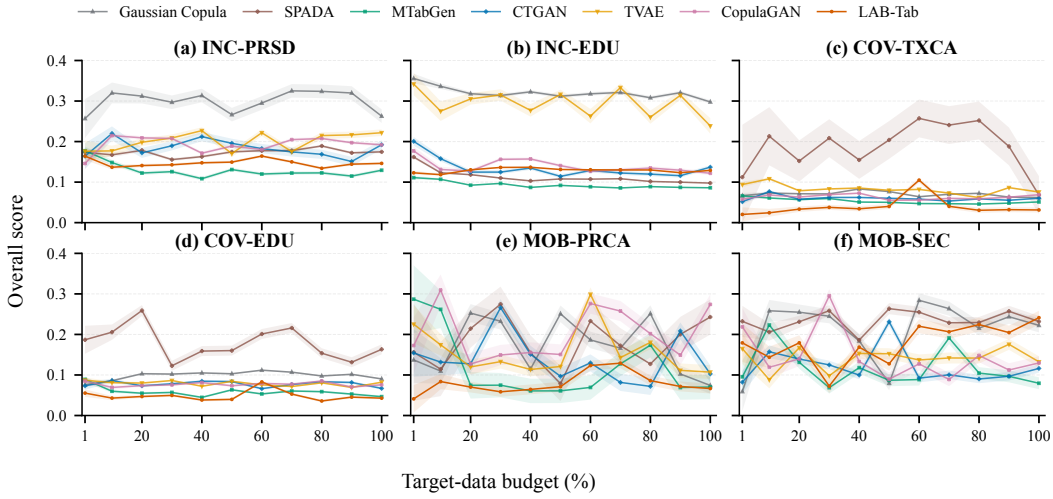

Figure 3: Macro Overall as the target-data fraction increases from 1% to 100%. Each method is retrained at every fraction, and lower values indicate better joint performance. Shading denotes variation across five random seeds. The structural adaptation cases and policy-ablation training dynamics are available in Appendices D and E.

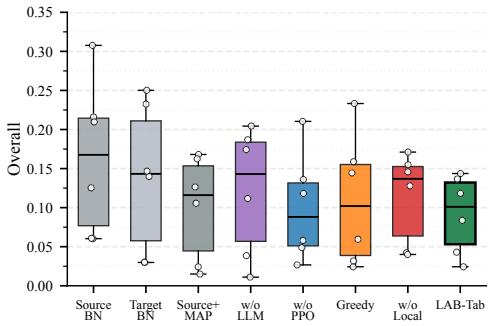

Figure 4: Scenario-wise Overall distributions for the ablations. Each circle is the seed-averaged score for one scenario; boxes show the interquartile range and horizontal lines show the median.

tirely on few-shot target data, and Source BN + MAP retains the source graph while updating its conditional probabilities with target counts. The w/o LLM, w/o PPO, and w/o local reward variants remove proposed edges, policy control, and edge-level feedback, respectively; Greedy edits replaces PPO with locally selected one-step actions.

Component contributions. Table 2 reveals distinct roles for the components of LAB-Tab. Direct source reuse performs poorly under target shift, while the target-only and MAP variants show that fitting sparse target data or they can match target marginals yet fail to preserve predictive and feature–label relationships. Removing LLM augmentation leaves JSD nearly unchanged but substantially reduces Coef–Cos, indicating that the proposed edges mainly contribute target-specific conditional structure. PPO then coordinates these dependencies: disabling it or replacing it with greedy edits produces less stable performance across scenarios, as reflected by the higher upper tails in Figure 4. Removing the local reward slightly improves WAPE because the policy can focus more directly on the global prediction objective, but this also weakens conditional and feature–label alignment. Similarly, the full model's slightly higher JSD reflects a small marginal-fidelity trade-off for better dependency preservation and downstream utility. The results show that LLM augmentation expands the target dependency space, while PPO coordination prevents optimization of aggregate errors from distorting conditional relationships.

6 Conclusion

Few-shot target-domain synthesis must reuse transferable source dependencies without retaining relationships that shift across domains. **LAB-Tab** addresses this challenge by adapting an explicit BN, it learns a source structure, introduces target-specific candidate edges under graph-validity constraints, recalibrates the augmented BN with few-shot target counts, and applies PPO to control edge activation, magnitude, and polarity through global and local feedback. Across six ACS source–target shifts at the 10% target-data budget, LAB-Tab achieves the lowest Overall score in four scenarios and reduces macro Overall by 33.8% relative to the strongest baseline, while obtaining the best macro JSD, WAPE, and Utility Gap and remaining competitive in feature–label preservation. It also leads at seven of eleven target-data fractions, with its largest gains at 1% and 10%. Ablations show that LLM augmentation, PPO-based edge control, and local dependency feedback provide complementary benefits beyond source reuse and MAP recalibration. Overall, LAB-Tab provides an effective and inspectable approach to few-shot tabular generation by combining explicit structural hypotheses with feedback-driven source-to-target adaptation.

References

Asghar, H. J.; Ding, M.; Rakotoarivelo, T.; Mrabet, S.; and Kaafar, M. A. 2019. Differentially Private Release of High-Dimensional Datasets Using the Gaussian Copula. *arXiv preprint arXiv:1902.01499*.

Borisov, V.; Sessler, K.; Leemann, T.; Pawelczyk, M.; and Kasneci, G. 2023. Language Models are Realistic Tabular Data Generators. In *The Eleventh International Conference on Learning Representations*.

Challagundla, R.; Dorodchi, M.; Wang, P.; and Lee, M. 2025. Synthetic Tabular Data Generation: A Comparative Survey for Modern Techniques. *arXiv preprint arXiv:2507.11590*.

Chen, T.; Shen, Z.; Zhou, B.; Liu, Y.; Wang, S.; and Ke, J. 2026. Addressing the online incremental transport mode choice prediction problem with an LLM-augmented class-incremental learning approach. *Transportation Research Part C: Emerging Technologies*, 188: 105709.

Christiano, P. F.; Leike, J.; Brown, T. B.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, volume 30.

Ding, F.; Hardt, M.; Miller, J. P.; and Schmidt, L. 2021. Retiring Adult: New Datasets for Fair Machine Learning. In *Advances in Neural Information Processing Systems*, volume 34, 6478–6490.

Dong, Y.; and Peng, C.-Y. J. 2013. Principled Missing Data Methods for Researchers. *SpringerPlus*, 2(1): 222.

Emmanuel, T.; Maupong, T.; Mpoeleng, D.; Semong, T.; Mphago, B.; and Tabona, O. 2021. A Survey on Missing Data in Machine Learning. *Journal of Big Data*, 8(1): 140.

Fang, L.; Liu, A.; Zhang, H.; Zou, H. P.; Zhang, W.; and Yu, P. S. 2025. TabGen-ICL: Residual-Aware In-Context Example Selection for Tabular Data Generation. *arXiv preprint arXiv:2502.16414*.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; et al. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 5637–5664.

Kotelnikov, A.; Baranchuk, D.; Rubachev, I.; and Babenko, A. 2023. TabDDPM: Modelling Tabular Data with Diffusion Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 17564–17579. PMLR.

Long, Y.; Xu, L.; and Brintrup, A. 2025. LLM-TabLogic: Preserving Inter-Column Logical Relationships in Synthetic Tabular Data via Prompt-Guided Latent Diffusion. *arXiv preprint arXiv:2503.02161*.

Ma, Y. J.; Liang, W.; Wang, G.; Huang, D.-A.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2024. Eureka: Human-Level Reward Design via Coding Large Language Models. In *International Conference on Learning Representations*.

Nguyen, A.; Schafft, S.; Hale, N.; and Alfaro, J. 2025. FASTGEN: Fast and Cost-Effective Synthetic Tabular Data Generation with LLMs. *arXiv preprint arXiv:2507.15839*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*, volume 35, 27730–27744.

Patki, N.; Wedge, R.; and Veeramachaneni, K. 2016. The Synthetic Data Vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics*, 399–410. IEEE.

Quiñonero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D., eds. 2009. *Dataset Shift in Machine Learning*. MIT Press.

Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; Kumar, M. P.; Dupont, E.; Ruiz, F. J. R.; Ellenberg, J. S.; Wang, P.; Fawzi, O.; Kohli, P.; and Fawzi, A. 2024. Mathematical Discoveries from Program Search with Large Language Models. *Nature*, 625(7995): 468–475.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Shi, R.; Wang, Y.; Du, M.; Shen, X.; and Wang, X. 2025. A Comprehensive Survey of Synthetic Tabular Data Generation. *arXiv preprint arXiv:2504.16506*.

Solatorio, A. V.; and Dupriez, O. 2023. REaLTabFormer: Generating Realistic Relational and Tabular Data Using Transformers. *arXiv preprint arXiv:2302.02041*.

Stoian, M. C.; Giunchiglia, E.; and Lukasiewicz, T. 2025. A Survey on Tabular Data Generation: Utility, Alignment, Fidelity, Privacy, and Beyond. *arXiv preprint arXiv:2503.05954*.

Villaizán-Vallelado, M.; Salvatori, M.; Segura, C.; and Arapakis, I. 2025. Diffusion Models for Tabular Data Imputation and Synthetic Data Generation. *ACM Transactions on Knowledge Discovery from Data*, 19(6): 1–32.

Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; and Veeramachaneni, K. 2019. Modeling Tabular Data Using Conditional GAN. In *Advances in Neural Information Processing Systems*, volume 32.

Xu, S.; Lee, C.-T.; Sharma, M.; Yousuf, R. B.; Muralidhar, N.; and Ramakrishnan, N. 2025. Why LLMs Are Bad at Synthetic Table Generation (and what to do about it). arXiv:2406.14541.

Yang, S.; Zhang, Z.; Prenkaj, B.; and Kasneci, G. 2025. Doubling Your Data in Minutes: Ultra-fast Tabular Data Generation via LLM-Induced Dependency Graphs. *arXiv preprint arXiv:2507.19334*.

Zhang, H.; Zhang, J.; Srinivasan, B.; Shen, Z.; Qin, X.; Faloutsos, C.; Rangwala, H.; and Karypis, G. 2024. Mixed-Type Tabular Data Synthesis with Score-based Diffusion in Latent Space. *arXiv preprint arXiv:2310.09656*.

Zhang, J.; Cormode, G.; Procopiuc, C. M.; Srivastava, D.; and Xiao, X. 2014. PrivBayes: Private Data Release via Bayesian Networks. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 1423–1434. ACM.

Zhao, Z.; Birke, R.; and Chen, L. Y. 2023. TabuLa: Harnessing Language Models for Tabular Data Synthesis. *arXiv preprint arXiv:2310.12746*.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2021. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1): 43–76.

A Dataset and Scenario Details

A.1 ACS Task Schemas

All experiments use person-level records from the 2018 ACS distributed through Folktables. Income retains working-age individuals with positive personal income and weekly working hours. Coverage focuses on individuals younger than 65 with annual income no greater than \$30,000, and Mobility focuses on adults aged 19–34. The targets are personal income, public health insurance coverage, and residential mobility, respectively.

A.2 Source–Target Scenario Construction

For each task, we construct one spatial shift and one population-group shift. The spatial scenarios are Puerto Rico to South Dakota for Income, Texas to California for Coverage, and Puerto Rico to California for Mobility. The population-group scenarios are constructed within California: the education shifts compare non-bachelor and bachelor-or-above groups, and the sector shift compares private-sector and government workers.

A.3 Data Splits and Target-Data Budgets

For each reporting seed, a stratified split reserves 20% of the target domain for held-out testing. A scenario-specific adaptation set is then sampled from the remaining records, with the sizes reported in Table 4. All methods use the same test and adaptation records within a scenario and seed.

A seeded permutation produces nested adaptation subsets containing $\{1, 10, 20, \ldots, 100\}\%$ of the adaptation set. The default experiment and the 10% sensitivity point use the same subset and run configuration.

A.4 Preprocessing and Discretization

The BN operates on finite state spaces. Each non-label numerical variable is discretized into at most eight source-domain quantile bins, and the same boundaries are applied to the target data. Repeated boundaries are merged, and missing feature values form an explicit category.

For Income, the label is transformed by $\log(1 + \text{income})$ and discretized into ten source-defined ordinal states for BN modeling. Each state is decoded through its source-bin median when regression metrics are computed in dollars. Coverage and Mobility use binary labels.

B Evaluation Metrics and Aggregation

Let $\mathcal{D}_t^{\text{test}}$ denote the held-out real target table and $\widehat{\mathcal{D}}_t$ a synthetic target table. All headline metrics use the same processed variables and are evaluated against $\mathcal{D}_t^{\text{test}}$.

B.1 Distributional Fidelity and WAPE

Mean marginal JSD. For a variable V_j with state space $\mathcal{A}_j$, let $p_j(a)$ and $\widehat{p}_j(a)$ denote the empirical probabilities of state a in $\mathcal{D}_t^{\text{test}}$ and $\widehat{\mathcal{D}}_t$, respectively, and let $m_j = (p_j + \widehat{p}_j)/2$. The marginal Jensen–Shannon divergence is

$$\text{JSD}_j = \frac{1}{2}\text{KL}(p_j \| m_j) + \frac{1}{2}\text{KL}(\widehat{p}_j \| m_j), \tag{30}$$

where natural logarithms are used and zero-probability terms contribute zero. We average this quantity over all modeled variables, including the label:

$$\text{JSD} = \frac{1}{|\mathcal{V}|} \sum_{V_j \in \mathcal{V}} \text{JSD}_j. \tag{31}$$

All variables are evaluated in their discrete BN representations; the Income label is not decoded for JSD.

WAPE. For each scenario, a task predictor is trained on the synthetic target table and evaluated on the held-out real target records. We compute

$$\text{WAPE} = \frac{\sum_{i=1}^{n_{\text{test}}} |y_i - \widehat{y}_i|}{\sum_{i=1}^{n_{\text{test}}} |y_i|}, \tag{32}$$

where y_i is the held-out target label. For INC-PRSD and INC-EDU, $\widehat{y}_i$ is the predicted income after decoding the ordinal income states to their source-bin medians. For COV-TXCA, COV-EDU, MOB-PRCA, and MOB-SEC, $\widehat{y}_i$ is the predicted positive-class probability. The same WAPE definition is used in all six scenarios.

B.2 Downstream Utility Gap

We evaluate downstream utility using Train on Synthetic, Test on Real (TSTR) and Train on Real, Test on Real (TRTR) under a matched target-test protocol. The TSTR predictor is trained on $\widehat{\mathcal{D}}_t$, while the TRTR reference predictor is trained on the same few-shot target subset available to the generator. Both predictors use the same model family and are evaluated on the same fixed $\mathcal{D}_t^{\text{test}}$. With U_{TSTR} and U_{TRTR} denoting the corresponding test utilities, we report

$$\text{UtilityGap} = |U_{\text{TRTR}} - U_{\text{TSTR}}|. \tag{33}$$

For Income, U is the coefficient of determination R^2 after decoding the ordinal income states. For Coverage and Mobility, U is ROC-AUC computed from positive-class probabilities. A smaller UtilityGap indicates that training on synthetic data more closely reproduces the predictive utility of the real few-shot target reference.

For each task, discrete feature states are ordinal-encoded. To keep feature codes aligned across real and synthetic evaluation tables, the encoder vocabulary is formed from the observed training and held-out feature states without using held-out labels. Any state outside this vocabulary is mapped to -1. Hyperparameters are fixed across methods, all predictor fits use fixed seed 0, and no predictor is tuned separately for a generator. Table 5 summarizes the configurations.

B.3 Feature–Label Coefficient Similarity

Coef–Cos evaluates whether a synthetic table preserves feature–label associations captured by matched linear probes. We apply a shared one-hot encoding to categorical features and standardize numerical features using a common preprocessing map. Two probes with the same specification are then fitted separately on the real and synthetic target tables, yielding coefficient vectors $\boldsymbol{\beta}_{\text{real}}$ and $\boldsymbol{\beta}_{\text{syn}}$. We report

$$\text{Coef-Cos} = \frac{\boldsymbol{\beta}_{\text{real}}^{\top} \boldsymbol{\beta}_{\text{syn}}}{\|\boldsymbol{\beta}_{\text{real}}\|_2 \, \|\boldsymbol{\beta}_{\text{syn}}\|_2}. \tag{34}$$

Table 3: Modeled variables for the three ACS tasks. Cat. denotes categorical, Q-num. quantile-binned numeric, Q-code quantile-binned code, and Ord. source-defined ordinal bins. Bold variables are labels; all remaining variables are features. State counts can be smaller when quantile boundaries coincide.

Income			Coverage			Mobility		
Variable	Type	States	Variable	Type	States	Variable	Type	States
Age	Q-num.	8	Age	Q-num.	8	Age	Q-num.	8
Class of worker	Cat.	8	Education	Cat.	24	Education	Cat.	24
Education	Cat.	24	Marital status	Cat.	5	Marital status	Cat.	5
Marital status	Cat.	5	Sex	Cat.	2	Sex	Cat.	2
Occupation	Q-code	8	Disability	Cat.	2	Disability	Cat.	2
Household relationship	Cat.	17–18	Citizenship	Cat.	5	Citizenship	Cat.	5
Weekly working hours	Q-num.	5	Prior-year mobility	Cat.	3	Class of worker	Cat.	10
Sex	Cat.	2	Personal income	Q-num.	7	Weekly working hours	Q-num.	6
Race	Cat.	8–9	Employment status	Cat.	7	Personal income	Q-num.	6–8
Personal income	Ord.	10	Race	Cat.	9	Race	Cat.	9
			Public coverage	Binary	2	**Residential mobility**	Binary	2

Table 4: Dataset sizes for the six source–target scenarios. n_{adapt} is the target set available for adaptation, and d_x excludes the label.

Scenario	n_s	n_{adapt}	n_{test}	d_x
INC-PRSD	9,071	1,595	980	9
INC-EDU	120,016	6,083	15,128	9
COV-TXCA	98,928	7,841	27,711	10
COV-EDU	115,947	3,416	4,521	9
MOB-PRCA	4,730	1,707	16,066	10
MOB-SEC	56,723	2,064	1,678	9

Higher values indicate closer agreement in coefficient direction and stronger preservation of feature–label associations.

B.4 Overall Score

Overall combines JSD, WAPE, Utility Gap, and Coef–Cos on a fixed loss-oriented scale. For scenario s, method m, and seed r, let $J_{s,m}^{(r)}$, $W_{s,m}^{(r)}$, $G_{s,m}^{(r)}$, and $C_{s,m}^{(r)}$ denote the four component metrics. We define

$$O_{s,m}^{(r)} = \frac{1}{4}\left(\frac{J_{s,m}^{(r)}}{\ln 2} + \frac{W_{s,m}^{(r)}}{1+W_{s,m}^{(r)}} + \frac{G_{s,m}^{(r)}}{1+G_{s,m}^{(r)}} + \frac{1-C_{s,m}^{(r)}}{2}\right). \tag{35}$$

Each term is bounded in $[0,1]$, and lower values indicate better performance. The JSD term uses its theoretical upper bound under natural logarithms; the WAPE and Utility Gap transformations preserve their ordering while bounding potentially unbounded values; and the Coef–Cos term converts similarity into a loss.

For each seed, the six-scenario macro score is

$$O_m^{(r)} = \frac{1}{6}\sum_{s=1}^{6} O_{s,m}^{(r)}. \tag{36}$$

Because the transformation is fixed, Overall is independent of the methods included in a table and is directly comparable across the headline, target-budget, and ablation experiments.

B.5 Aggregation and Uncertainty

All headline experiments are repeated with $R = 5$ independent reporting seeds. For a metric Q, the scenario-level mean and sample standard deviation are

$$\overline{Q}_{s,m} = \frac{1}{R}\sum_{r=1}^{R} Q_{s,m}^{(r)}, \qquad R = 5, \tag{37}$$

$$S_{s,m} = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}\left(Q_{s,m}^{(r)} - \overline{Q}_{s,m}\right)^2}. \tag{38}$$

For Overall, the fixed transformation is applied first within each seed and scenario. The six-scenario average is then computed within each seed, followed by the mean and empirical standard deviation across five seeds. Scenario entries without uncertainty are five-seed means. The 10% headline table, 10% sensitivity point, and 10% ablations follow this same order. Appendix E presents representative single-seed training traces, while cross-seed uncertainty comes from the five reporting seeds.

C Implementation and Reproducibility Details

C.1 Bayesian-Network Learning and Calibration

The source Bayesian network is learned independently for each scenario. Structure learning starts from an empty directed acyclic graph and applies greedy hill climbing with the Bayesian information criterion. To bound the cost of structure search, at most 10,000 source records are selected; parameter estimation still uses the complete source training table. The search evaluates single-edge addition, deletion, and reversal operations, rejects operations that introduce a cycle or violate the maximum in-degree, and terminates after 200 iterations or when no admissible operation improves the score. No random restart is used. A fixed column order and deterministic operator enumeration are retained throughout a run.

After structure learning, every conditional probability table (CPT) is estimated on the complete source table with a

Table 5: Downstream predictors used for TSTR and TRTR evaluation.

Task	Predictor and input	Configuration
Income	Histogram gradient-boosting regressor; ordinal feature states; income states decoded to source-bin medians	150 iterations; fixed seed 0; utility: R^2
Coverage	Histogram gradient-boosting classifier; ordinal feature states; unseen states mapped to -1	200 iterations; depth 6; learning rate 0.1; fixed seed 0; utility: ROC-AUC
Mobility	Histogram gradient-boosting classifier; ordinal feature states; unseen states mapped to -1	200 iterations; depth 6; learning rate 0.1; fixed seed 0; utility: ROC-AUC

Table 6: Bayesian-network and sampling settings.

Component	Setting
Discretization	At most 8 source-quantile states per numerical variable
Structure score	BIC greedy hill climbing from an empty DAG
Structure rows	$\min(\lvert\mathcal{D}_s\rvert, 10{,}000)$; fixed seed 0
Search limit	200 operations; no random restart
Maximum in-degree	$d_{\max} = 4$
Source CPT prior	BDeu, $\alpha = 10$
Target MAP prior	Source-centered Dirichlet, $\kappa = 1$
PPO rollout size	800 rows for Income; 1,500 otherwise

symmetric BDeu prior of equivalent sample size $\alpha = 10$. Following LLM augmentation, the source CPTs are re-estimated on the augmented parent sets and calibrated with the few-shot target counts using $\kappa = 1$. Unobserved parent configurations retain the source conditional distribution; observed configurations combine the source prior with target counts. Table 6 summarizes the settings used in all six scenarios.

For each action comparison, the pre-action and post-action samplers use common random numbers. This holds the underlying draws fixed while changing the selected edge control and reduces Monte Carlo noise in the reward difference. The final synthetic table is generated with a separate fixed seed after selecting the best edge-control configuration.

C.2 Edge-Control Parameterization

The base potential b_j and edge potentials ψ_e are initialized from Θ_{aug} to obtain the controlled BN $B^{(0)}$ used at the start of each episode. These potentials remain fixed during PPO, while the policy updates the activation m_e, magnitude g_e, and polarity d_e of every edge in $\mathcal{E}_{\text{aug}}$. Thus, each action changes the contribution of a validated dependency without altering its endpoints or parent–child orientation.

During optimization, $\mathcal{G}_{\text{aug}}$ defines the editable edge set. Deactivation sets an edge contribution to zero and allows later reactivation through *strengthen*, while *flip* changes its polarity. The retained activation variables define $\mathcal{G}^\star$, and the retained controls determine $\Theta^\star$ for final sampling.

C.3 PPO Architecture, Reward, and Selection

Each augmented BN edge is represented by four feature blocks: current controls, target-alignment statistics, local graph context, and adjustment history. The alignment block contains edge-conditional and endpoint-level generated–target discrepancies. Two edge representations are adjacent when their BN edges share an endpoint. A two-layer GraphSAGE encoder maps the normalized coordinates to 64-dimensional edge embeddings. The actor applies a 128-unit multilayer perceptron to each edge embedding; the critic applies a separate 128-unit multilayer perceptron to mean- and max-pooled graph embeddings.

At each step, the actor samples one valid edge–operation pair. Keep, weaken, strengthen, and deactivate control the selected contribution, and strengthen can reactivate a deactivated edge. The *flip* action changes its polarity. All actions preserve the validated endpoints and parent–child orientation. Ineffective operations are masked, and keep remains valid for every edge.

Let $\widehat{A}_k$ and R_k denote the generalized-advantage estimate and return for transition k. The actor and critic minimize

$$\begin{aligned}\mathcal{L}_{\text{PPO}} = & -\mathcal{L}_{\text{clip}} + c_V \mathbb{E}_k\big[(V(s_k) - R_k)^2\big] \\ & - c_H \mathbb{E}_k[H(\pi(\cdot \mid s_k))]\,,\end{aligned} \tag{39}$$

where c_V is the value-loss coefficient and c_H the episode-dependent entropy coefficient. Advantages are standardized within an episode. State coordinates are normalized online with Welford statistics and clipped to $[-10, 10]$; rewards are divided by their running standard deviation and clipped to the same interval. Gradient norms are clipped at 0.5.

For each run, the state with the smallest global alignment loss over all episodes and steps is retained. The held-out target set is excluded from both policy optimization and configuration selection. Across the five reporting seeds, the target split and few-shot subset are sampled independently, so the reported variation reflects different target draws as well as model randomness.

C.4 Baseline Configurations

Every headline baseline receives the same source table and few-shot target subset as LAB-Tab. The two tables are concatenated without a domain indicator, target oversampling, or scenario-specific reweighting. The Source BN, Target BN, and Source BN + MAP variants in Table 12 provide complementary BN references. All methods use the same source-fitted discretization and are evaluated on the same target test set within each seed.

MTabGen treats each discretized column as a categorical token with a column-specific vocabulary and optimizes masked-cell reconstruction using AdamW with weight decay 10^{-4}. Sampling starts from empirical marginal draws

Table 7: PPO settings for the regression and classification scenarios.

Parameter	Income	Coverage/Mobility
Episodes	150	250
Steps per episode	20	20
Rollout rows	800	1,500
Optimizer	Adam	Adam
Learning rate	3×10^{-4}	3×10^{-4}
PPO epochs per episode	4	4
Clip ratio ϵ	0.20	0.20
Discount factor γ	0.99	0.99
GAE parameter λ	0.95	0.95
Value coefficient c_V	0.50	0.50
Entropy coefficient c_H	$0.020 \rightarrow 0.003$	$0.020 \rightarrow 0.003$
Target KL threshold	0.03	0.03
Weaken factor η_-	0.85	0.85
Strengthen factor η_+	1.20	1.20
Global reward weight	0.20	0.20
Local reward weight	1.00	1.00

and iteratively denoises the table at temperature 0.9. CopulaGAN maps quantile labels to ordinal numbers only for its internal copula transformation and maps generated values back to the nearest valid quantile state. SPADA constructs its LLM dependency graph separately for each scenario, target-data budget, and reporting seed, followed by graph validation before sampling.

C.5 Computational Environment

Experiments were executed on a Mac Studio with an Apple M2 Ultra processor (24 CPU cores) and 192 GB of unified memory. The archived environment uses Python 3.10.11, NumPy 2.0.2, pandas 2.3.3, SciPy 1.15.3, scikit-learn 1.7.2, pgmpy 1.1.2, Folktables 0.0.12, SDV 1.37.1, NetworkX 3.4.2, PyTorch 2.12.0, and PyTorch Geometric 2.8.0. LAB-Tab and the SDV baselines run on CPU in this environment. The scenario–seed LAB-Tab jobs were launched independently; measured wall-clock times from the training-event and result timestamps range from approximately 5.8 to 16.8 minutes per scenario.

Python, NumPy, and PyTorch are initialized separately for each of the five reporting seeds. The target permutation, model initialization, policy sampling, and synthetic sampling are derived deterministically from the corresponding seed.

LLM augmentation uses *gpt-4o* with temperature 0.2 and a maximum output length of 4,096 tokens. A proposal is generated for each scenario, target-data budget, and reporting seed; validated proposals are reused across ablations with the same split and budget.

D LLM Edge Proposals and Structural Adaptation

D.1 Prompt Construction

The LLM receives aggregate context containing semantic variable names, state domains, the source BN, source and target marginal summaries, label-conditional summaries, and selected pairwise conditional contrasts. The context is constructed from the source table and few-shot target subset.

The prompt requests a short ranked list of missing directed edges. Each proposal contains a parent, child, confidence score, and concise rationale. The LLM stage proposes structure; CPT estimation and row generation remain in the BN pipeline.

D.2 Proposal Validation

Proposals are sorted by confidence and processed sequentially against the updated graph. Proposals are validated to ensure valid variables, graph consistency, and DAG constraints before being added to the source graph. Accepted proposals retain their confidence and rationale for initialization and analysis. Malformed responses are retried, and the source BN is used when no valid addition is returned.

D.3 Proposal Statistics and Structural Cases

Table 9 reports the accepted additions shown in Figure 5 and representative PPO outcomes at the 10% target-data budget. For an accepted edge e, the LLM confidence initializes its magnitude as $g_e^{(0)} = g_{\min} + (1 - g_{\min})c_e^{\mathrm{LLM}}$.

LLM-proposed additions receive different controls across scenarios. PPO strengthens, weakens, flips, or deactivates their contributions according to the generated–target feedback.

D.4 Scenario-Level BN Adaptation Patterns

Figure 5 shows that the source BN provides the structural backbone in every scenario. LLM additions expand the editable edge set, and PPO assigns the final controls to inherited and added dependencies. The selected edits concentrate around label-related paths and differ across the Income, Coverage, and Mobility shifts.

E PPO Optimization Dynamics

E.1 Evolution of Edge-Control Actions

For an episode e containing $T = 20$ edge-control steps, we summarize the policy behavior by the fraction of steps assigned to action o:

$$p_e(o) = \frac{1}{T} \sum_{k=1}^{T} \mathbf{1}\left[o^{(e,k)} = o\right]. \tag{40}$$

The five logged actions are keep, weaken, strengthen, deactivate, and flip. Deactivate sets the selected edge contribution to zero while leaving the augmented DAG unchanged. Flip changes the sign of the selected dependency contribution while preserving the parent–child orientation. These names follow the terminology used in the main paper.

Figure 6 shows a common transition from exploratory edge adjustment to a keep-dominated policy. Over the final fifth of training, the mean keep fraction is 0.92 or higher in five scenarios and reaches 1.00 in MOB-SEC. The exception is MOB-PRCA, where the final-fifth keep fraction is 0.73 and weaken and strengthen remain active at 0.08 and 0.14, respectively. This longer adjustment phase is consistent with a more heterogeneous policy search in that scenario. The rising

Table 8: Headline baseline configurations. All methods use the same source table and few-shot target subset.

Method	Training data	Configuration
Gaussian Copula	Source + few-shot target	SDV Gaussian-copula synthesizer with all processed columns treated as categorical
SPADA	Source + few-shot target	Categorical specialization of dependency-aware sampling; LLM dependency DAG; temperature 1.0
MTabGen	Source + few-shot target	80 epochs; batch 1,024; 3 transformer layers; 4 heads; width 96; learning rate 10^{-3}; mask probability 0.35; 24 denoising steps
CTGAN	Source + few-shot target	SDV implementation; 100 epochs; package-default architecture and batch size; CPU training
TVAE	Source + few-shot target	SDV implementation; 100 epochs; package-default architecture and batch size; CPU training
CopulaGAN	Source + few-shot target	SDV implementation; 300 epochs; quantile-bin variables encoded ordinally during fitting and decoded after sampling

Table 9: Structural adaptation statistics at the 10% target-data budget. Source edges denote dependencies in the source BN, LLM additions denote new edges proposed by the LLM, and active additions denote proposed edges retained after PPO calibration.

Scenario	Source edges	LLM additions	Active additions	LLM-proposed additions
INC-PRSD	8	3	3	Age→Income; Education→Income; Marital status→Income
INC-EDU	8	3	3	Class of worker→Income; Education→Income; Occupation→Income
COV-TXCA	11	3	3	Citizenship→Coverage; Coverage→Employment; Income→Sex
COV-EDU	12	3	2	Coverage→Age; Coverage→Sex; Citizenship→Coverage
MOB-PRCA	8	5	4	Age→Mobility; Education→Mobility; Marital status→Mobility; Race→Mobility; Citizenship→Mobility
MOB-SEC	9	3	2	Race→Mobility; Mobility→Disability; Sex→Mobility

(b) Representative structural cases

Scenario	Accepted edge	Rationale	PPO outcome	Multiplier
INC-PRSD	Age→Income	Age adds direct target-domain information for the income distribution.	Strengthened	1.20
INC-EDU	Class of worker→Income	Class of worker captures income differences within the target education group.	Weakened	0.85
COV-TXCA	Coverage→Employment	Coverage contributes to target-specific employment patterns in the target domain.	Flipped	−1.00
COV-EDU	Citizenship→Coverage	Citizenship provides a plausible coverage-related dependency in the target group.	Deactivated	0.00
MOB-PRCA	Marital status→Mobility	Marital status contributes to residential mobility under the geographic shift.	Strengthened	1.44
MOB-SEC	Race→Mobility	Race captures a target-specific mobility association under the sector shift.	Weakened	0.85

keep fraction shows that later episodes make progressively fewer changes to the retained controls, consistent with the stable configurations evaluated in Appendix F.

E.2 Episode-Return Dynamics

Let $r^{(e,k)}$ denote the reward at step k of episode e. The episode return is

$$R_e = \sum_{k=1}^{20} r^{(e,k)}. \tag{41}$$

Figure 7 reports this quantity for LAB-Tab, the variant without the local reward, and the variant without LLM edge proposals. The solid curves are 10-episode rolling means, and the shaded regions show pointwise 95% smoothing bands computed within the same rolling window. Cross-seed variation is reported separately in the result tables.

The magnitude of the return decreases over training in all six scenarios, showing that later edge operations produce progressively smaller changes under each variant's objective. This behavior is most pronounced in INC-PRSD, COV-TXCA, and MOB-SEC, where the rolling means approach zero near the end of training. The MOB-PRCA traces remain more variable, consistent with the longer weaken–strengthen phase in Figure 6. The variants use different reward definitions, and their final quality is summarized by UtilityGap, Coef–Cos, and Overall in the ablation results.

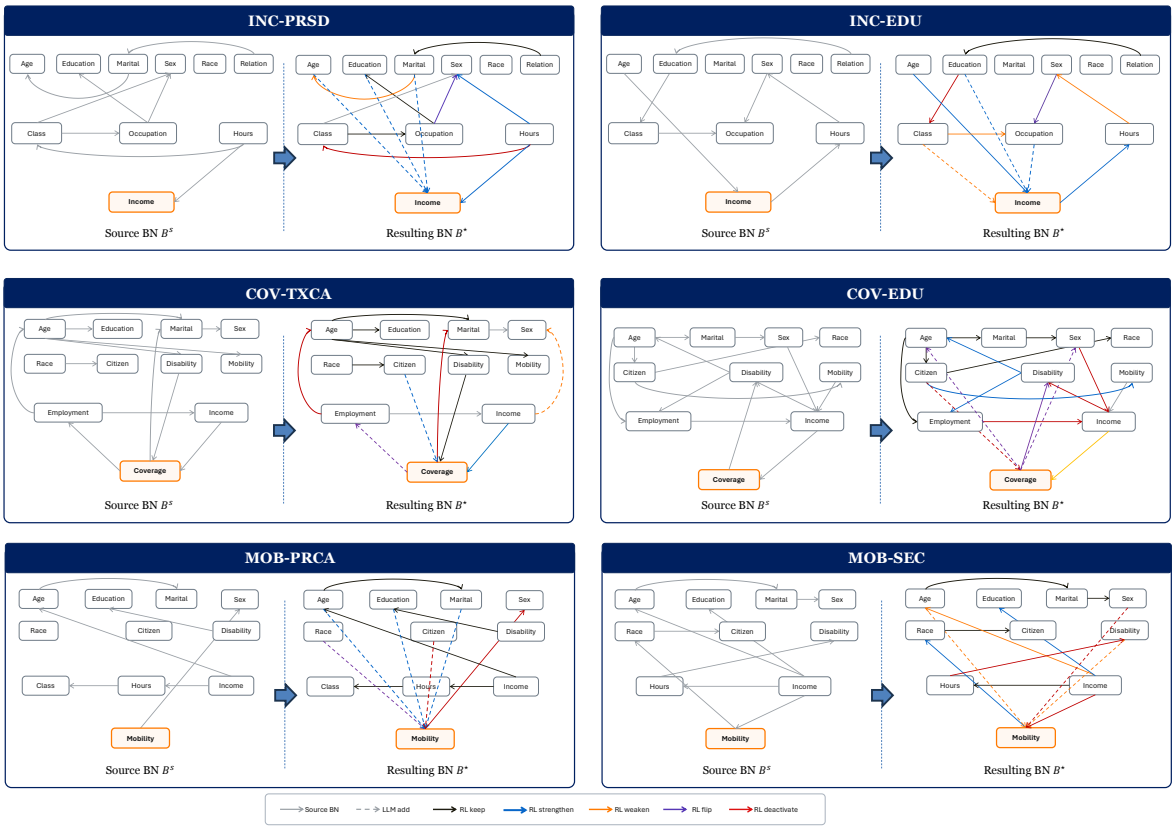

Figure 5: Source and selected target-domain BN configurations for the six shifts. Gray solid and dashed arrows denote source and LLM-added edges. Black, blue, orange, purple, and red denote keep, strengthen, weaken, flip, and deactivate outcomes. Deactivated edges are omitted from the final active graph.

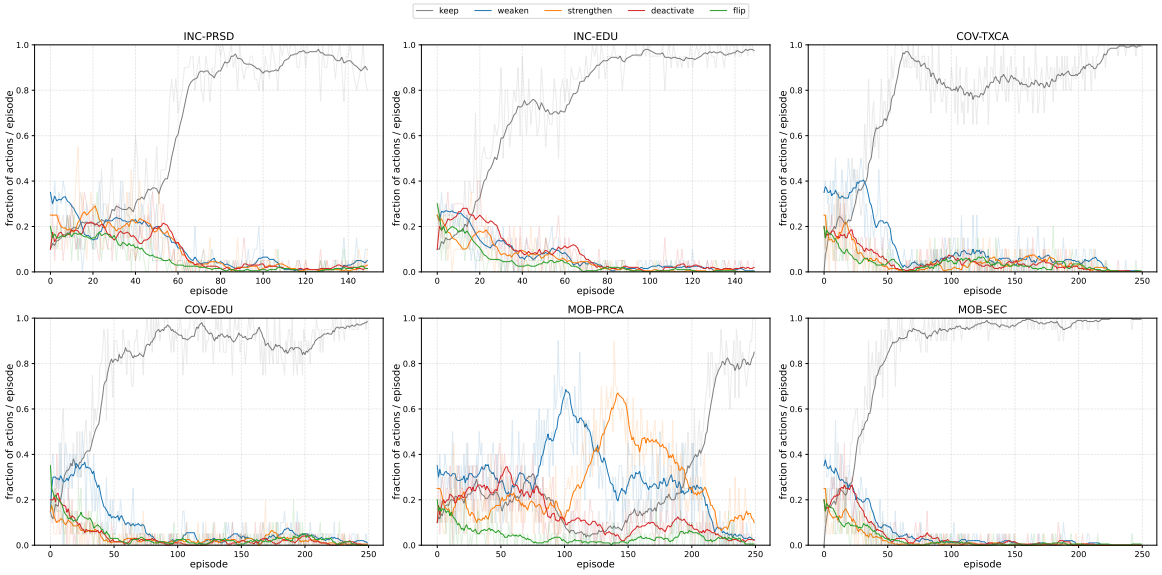

Figure 6: Episode-wise edge-control action fractions for the six scenarios. Light traces show episode values, and dark traces show 10-episode moving averages. Each episode contains 20 actions.

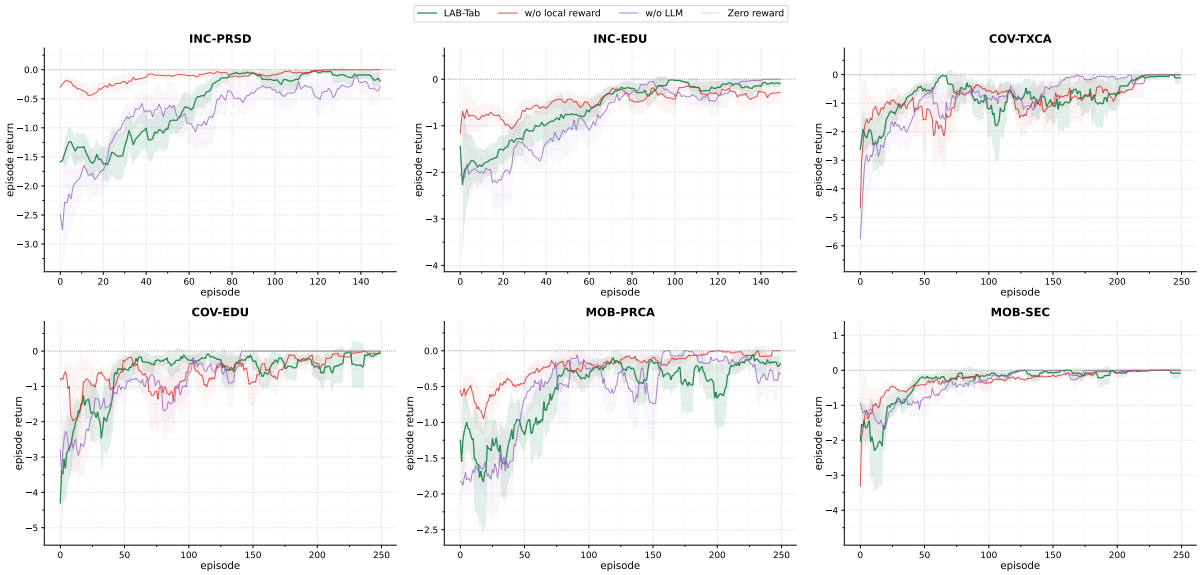

Figure 7: Episode-return dynamics for LAB-Tab and two policy ablations. Solid curves are 10-episode rolling means, shaded regions are within-window 95% smoothing bands, and the dotted line marks zero return.

F Complete Experimental Results

All results use the six scenarios in Table 4, five reporting seeds, and the fixed Overall definition in Appendix B. The 10% headline and sensitivity entries use the same runs.

F.1 Scenario-Level Main Results

F.2 Results across Target-Data Budgets

Table 11 summarizes the main target-budget anchors. LAB-Tab leads seven of eleven budgets, with its largest gains in the low-data regime. At 1%, it reduces Overall by 54.6% relative to CTGAN. MTabGen leads at 40%, 60%, 90%, and 100%, and the two methods remain close at the full-data endpoint. This crossover is consistent with source-informed adaptation being most useful when target evidence is scarce.

Table 11: Selected anchors from the target-data sensitivity analysis.

Setting	LAB-Tab	Reference
1% Overall	**0.1499**	CTGAN: 0.3299
10% Overall	**0.0916**	CTGAN: 0.1383
100% Overall	0.1751	**MTabGen: 0.1650**
Budget wins	**7/11**	MTabGen: 4/11

F.3 Ablation Summary

Table 12 reports the fixed-scale ablation averages. The results show complementary roles across the components. LLM augmentation mainly improves conditional and feature–label structure, PPO coordinates inherited and added edges, and local feedback improves the balance of utility and association preservation.

Table 12: Ablation results averaged over five seeds and six scenarios.

Method	JSD↓	WAPE↓	UGap↓	Coef.↑	Overall↓
Source BN	0.0318	0.3137	0.1861	0.5280	0.1633
Target BN	**0.0011**	0.1815	0.1550	0.4137	0.1382
Source BN + MAP	**0.0011**	0.1641	0.0923	0.6082	0.1003
w/o LLM	0.0015	0.1659	0.0929	0.4478	0.1211
w/o PPO	0.0013	0.1626	0.0766	0.5803	0.0997
Greedy edits	0.0015	0.1707	0.0995	0.5536	0.1087
w/o local reward	0.0016	**0.1584**	0.0865	0.4892	0.1137
LAB-Tab	0.0016	0.1600	**0.0728**	**0.6464**	**0.0916**

Table 10: Complete five-seed results for all six scenarios at the 10% target-data budget. Component entries are mean ± empirical standard deviation; Overall entries are fixed-score means. Best means within each scenario are bold.

Method	JSD ↓	WAPE ↓	UGap ↓	Coef–Cos ↑	Overall ↓
INC-PRSD					
Gaussian Copula	0.0318 ± 0.0005	0.5878 ± 0.0091	0.7089 ± 0.0385	0.1016 ± 0.0084	0.3200
MTabGen	0.0402 ± 0.0027	0.4268 ± 0.0055	0.1567 ± 0.0211	$\mathbf{0.7990 \pm 0.0032}$	0.1483
CTGAN	0.0424 ± 0.0057	0.4642 ± 0.0128	0.2670 ± 0.0408	0.4177 ± 0.0059	0.2200
TVAE	0.0456 ± 0.0118	0.4525 ± 0.0134	0.2383 ± 0.0630	0.7235 ± 0.0072	0.1770
CopulaGAN	0.0451 ± 0.0043	0.4788 ± 0.0074	0.3409 ± 0.0300	0.5674 ± 0.0066	0.2149
SPADA	0.0328 ± 0.0001	0.4374 ± 0.0069	0.1524 ± 0.0256	0.6296 ± 0.0027	0.1673
LAB-Tab	$\mathbf{0.0055 \pm 0.0002}$	$\mathbf{0.4258 \pm 0.0067}$	$\mathbf{0.1093 \pm 0.0171}$	0.7166 ± 0.0024	**0.1367**
INC-EDU					
Gaussian Copula	0.0379 ± 0.0003	0.6026 ± 0.0094	1.1755 ± 0.0505	0.2508 ± 0.0047	0.3364
MTabGen	0.0220 ± 0.0029	$\mathbf{0.3622 \pm 0.0054}$	$\mathbf{0.1014 \pm 0.0083}$	0.9244 ± 0.0012	**0.1069**
CTGAN	0.0421 ± 0.0025	0.4076 ± 0.0122	0.2450 ± 0.0346	0.8317 ± 0.0035	0.1578
TVAE	0.0921 ± 0.0086	0.4724 ± 0.0622	0.5107 ± 0.0512	0.3832 ± 0.0081	0.2750
CopulaGAN	0.0448 ± 0.0030	0.3701 ± 0.0119	0.1428 ± 0.0361	0.8710 ± 0.0043	0.1311
SPADA	0.0264 ± 0.0002	0.3842 ± 0.0047	0.1704 ± 0.0089	$\mathbf{0.9386 \pm 0.0007}$	0.1230
LAB-Tab	$\mathbf{0.0008 \pm 0.0004}$	0.3639 ± 0.0065	0.1425 ± 0.0199	0.8385 ± 0.0008	0.1184
COV-TXCA					
Gaussian Copula	0.0054 ± 0.0002	0.1286 ± 0.0031	0.1877 ± 0.0179	0.9761 ± 0.0019	0.0729
MTabGen	0.0100 ± 0.0009	0.1831 ± 0.0116	$\mathbf{0.0705 \pm 0.0040}$	0.9833 ± 0.0002	0.0608
CTGAN	0.0090 ± 0.0012	0.2002 ± 0.0136	0.1021 ± 0.0149	0.9315 ± 0.0014	0.0767
TVAE	0.0538 ± 0.0059	0.3618 ± 0.0268	0.0948 ± 0.0257	$\mathbf{0.9944 \pm 0.0007}$	0.1082
CopulaGAN	0.0105 ± 0.0012	0.1948 ± 0.0127	0.0927 ± 0.0110	0.9807 ± 0.0016	0.0682
SPADA	$0.0026 \pm <0.0001$	0.0858 ± 0.0013	0.1863 ± 0.0070	-0.2254 ± 0.0231	0.2131
LAB-Tab	$\mathbf{0.0002 \pm 0.0001}$	$\mathbf{0.0198 \pm 0.0020}$	0.0792 ± 0.0171	0.9918 ± 0.0011	**0.0243**
COV-EDU					
Gaussian Copula	0.0207 ± 0.0003	0.2297 ± 0.0027	0.1202 ± 0.0132	0.9618 ± 0.0026	0.0858
MTabGen	0.0220 ± 0.0036	0.2343 ± 0.0369	0.0046 ± 0.0115	0.9733 ± 0.0006	0.0599
CTGAN	0.0253 ± 0.0025	0.2869 ± 0.0210	0.0708 ± 0.0108	0.9618 ± 0.0024	0.0862
TVAE	0.0426 ± 0.0038	0.3052 ± 0.0050	0.0211 ± 0.0332	0.9845 ± 0.0007	0.0809
CopulaGAN	0.0285 ± 0.0022	0.2946 ± 0.0310	$\mathbf{0.0012 \pm 0.0127}$	$\mathbf{0.9853 \pm 0.0013}$	0.0693
SPADA	0.0179 ± 0.0001	0.2079 ± 0.0014	0.1336 ± 0.0186	-0.0132 ± 0.0264	0.2056
LAB-Tab	$\mathbf{0.0006 \pm 0.0002}$	$\mathbf{0.0479 \pm 0.0086}$	0.0048 ± 0.0083	0.7590 ± 0.0078	**0.0430**
MOB-PRCA					
Gaussian Copula	0.0161 ± 0.0004	0.1720 ± 0.0036	0.1043 ± 0.0124	0.6518 ± 0.0472	0.1096
MTabGen	0.0241 ± 0.0013	0.2587 ± 0.0127	$\mathbf{0.0507 \pm 0.0072}$	-0.5180 ± 0.0246	0.2619
CTGAN	0.0288 ± 0.0014	0.2915 ± 0.0194	0.1046 ± 0.0098	$\mathbf{0.6697 \pm 0.0431}$	0.1318
TVAE	0.0349 ± 0.0077	0.2950 ± 0.0203	0.0655 ± 0.0145	0.2847 ± 0.0442	0.1743
CopulaGAN	0.0362 ± 0.0017	0.3187 ± 0.0244	0.0755 ± 0.0110	-0.7473 ± 0.0338	0.3094
SPADA	0.0158 ± 0.0003	0.1624 ± 0.0020	0.0946 ± 0.0107	0.5833 ± 0.0390	0.1143
LAB-Tab	$\mathbf{0.0004 \pm 0.0001}$	$\mathbf{0.0319 \pm 0.0048}$	0.0643 ± 0.0049	0.5140 ± 0.0092	**0.0837**
MOB-SEC					
Gaussian Copula	0.0049 ± 0.0002	0.1157 ± 0.0019	0.0948 ± 0.0278	-0.6725 ± 0.0418	0.2584
MTabGen	0.0093 ± 0.0009	0.1870 ± 0.0131	0.0668 ± 0.0157	-0.3176 ± 0.0243	0.2231
CTGAN	0.0117 ± 0.0030	0.2081 ± 0.0328	0.0852 ± 0.0240	0.2771 ± 0.0302	0.1573
TVAE	0.0263 ± 0.0096	0.2540 ± 0.0438	$\mathbf{0.0029 \pm 0.0258}$	$\mathbf{0.7809 \pm 0.0191}$	**0.0882**
CopulaGAN	0.0339 ± 0.0041	0.2982 ± 0.0318	0.0579 ± 0.0254	0.7151 ± 0.0399	0.1189
SPADA	0.0047 ± 0.0001	0.1123 ± 0.0009	0.0912 ± 0.0226	-0.2649 ± 0.0186	0.2059
LAB-Tab	$\mathbf{0.0018 \pm 0.0002}$	$\mathbf{0.0707 \pm 0.0082}$	0.0367 ± 0.0159	0.0587 ± 0.0046	0.1437