

Fair Data Pre-Processing with Imperfect Attribute Space

YING ZHENG, National University of Singapore, Singapore

YANGFAN JIANG, National University of Singapore, Singapore

KIAN-LEE TAN, National University of Singapore, Singapore

Fair data pre-processing is a widely used strategy for mitigating bias in machine learning. A promising line of research focuses on calibrating datasets to satisfy a designed fairness policy so that sensitive attributes influence outcomes only through clearly specified legitimate causal pathways. While effective on clean and information-rich data, these methods often break down in real-world scenarios with imperfect attribute spaces, where decision-relevant factors may be deemed unusable or even missing. To address this gap, we propose LatentPre, a novel framework that enables principled and robust fair data processing in practical settings. Instead of relying solely on observed attributes, LatentPre augments the fairness policy with latent attributes that capture essential but subtle signals, enabling the framework to operate as if the attribute space were perfect. These latent attributes are strategically introduced to guarantee identifiability and are estimated using a tailored expectation-maximization paradigm. The raw data is then carefully refined to conform to this latent-augmented policy, effectively removing biased patterns while preserving justifiable ones. Extensive experiments demonstrate that LatentPre consistently achieves strong fairness-utility trade-offs across diverse scenarios, advancing practical fairness-aware data management.

CCS Concepts: • **Information systems** → **Data management systems**; • **Computing methodologies** → **Machine learning**.

Additional Key Words and Phrases: Data pre-processing; Causal fairness

ACM Reference Format:

Ying Zheng, Yangfan Jiang, and Kian-Lee Tan. 2026. Fair Data Pre-Processing with Imperfect Attribute Space. *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 180 (June 2026), 26 pages. <https://doi.org/10.1145/3802057>

1 Introduction

Machine learning (ML) has become a core component of modern data-driven systems, supporting decision-making in domains such as finance [23], education [5, 27], healthcare [26], and public administration [6, 31, 38]. However, these advances also raise growing concerns about fairness. In practice, ML models often unintentionally learn biased patterns from the raw data maintained by database management systems (DBMSs), and these patterns may be further amplified during training, resulting in unfair or discriminatory predictions. [20, 32, 40, 41, 44, 49, 53, 54].

A promising way to mitigate this issue is to apply fairness-aware data pre-processing before the data enters ML training. This ensures that models are trained on data as if collected in a fair world, rather than drawn from potentially biased distributions stored in DBMSs. Substantial research efforts have been devoted to exploring fairness criteria and developing fair data pre-processing solutions [7, 11–13, 18, 25, 30, 36, 37, 50, 52, 55]. Among them, methods based on *justifiable fairness* [37] have received considerable attention [12, 30, 33, 36, 55, 56], due to their favorable fairness-utility trade-offs, interpretability, and practical applicability.

Authors' Contact Information: Ying Zheng, National University of Singapore, Singapore, Singapore, zheng.ying@u.nus.edu; Yangfan Jiang, National University of Singapore, Singapore, Singapore, jyangfan@u.nus.edu; Kian-Lee Tan, National University of Singapore, Singapore, Singapore, tanl@comp.nus.edu.sg.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).

ACM 2836-6573/2026/6-ART180

<https://doi.org/10.1145/3802057>

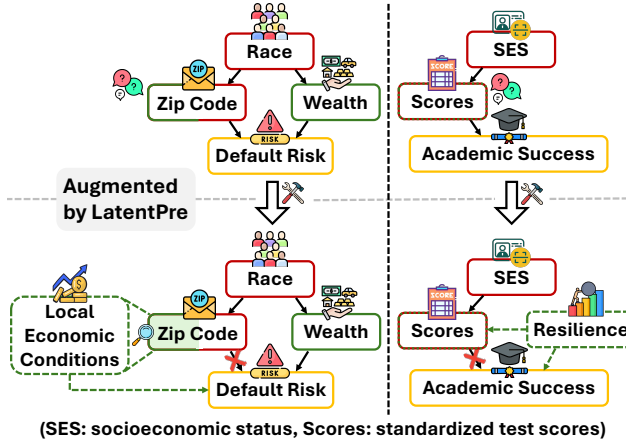


Fig. 1. Attribute graphs of real-world examples.

Justifiable fairness restricts sensitive attributes, such as gender or race, to influencing decisions only through ethically acceptable attributes, known as admissible attributes, even if such attributes carry biased information. In contrast, other attributes that encode bias, termed inadmissible attributes, are not allowed to affect decisions. To instantiate this notion in data pre-processing, a standard recipe is to define a *fairness policy* that models a hypothetically fair world, in which only justifiably fair causal relationships remain. The raw data is then modified so that its empirical distribution conforms to this policy, as if collected from such a world. Existing methods [37, 55] typically construct this policy within an attribute space that mirrors the raw DBMS schema, relying on straightforward fair representations. When applied to clean and information-rich datasets with clearly specified admissible and inadmissible attributes, these frameworks achieve desirable fairness-utility trade-offs.

1.1 Fair Data Pre-Processing in the Wild

In real-world DBMSs, however, raw data often comes with an imperfect attribute space. One common challenge in such settings is **attribute ambiguity**, where some attributes have unclear roles, making it difficult even for domain experts to decide whether they should be treated as admissible or inadmissible. When such attributes are conservatively treated as inadmissible to avoid potential bias, existing methods typically block their influence on the decision. This operation can substantially undermine data utility, as valuable predictive signals are removed during processing. Another challenge is **attribute absence**, where certain utility-relevant factors are not recorded in the schema due to collection constraints or their abstract nature. Although the effect of such factors may be partially reflected in observed attributes, the factors themselves are not explicitly present. When the corresponding information is entangled with inadmissible attributes, existing methods may also suppress the resulting signals, even when these signals are legitimate. As a result, the already limited data utility can be further compromised after fairness-aware processing, and the processed data may no longer support effective decision-making. To make these challenges concrete, we illustrate both sources of imperfection with the following examples.

EXAMPLE 1 (ATTRIBUTE AMBIGUITY). Consider a loan default prediction task. As shown in Figure 1 (top left), race is sensitive, wealth is admissible, and default risk is the outcome. The role of zip code, however, is ambiguous. It correlates with race and may therefore reflect historical or structural bias,

while it can also serve as a proxy for local economic conditions relevant to an applicant's financial stability. For example, in a town whose economy depends on a single large factory, news of the factory's impending shutdown can sharply increase residents' future financial uncertainty. In this case, zip code becomes a forward-looking risk indicator beyond what current wealth alone can reveal. If practitioners conservatively treat zip code as inadmissible, existing fairness-aware methods typically block their influence on the decision, which can reduce data utility.

EXAMPLE 2 (ATTRIBUTE ABSENCE). *Consider a university scholarship allocation task that aims to predict academic success. As shown in Figure 1 (top right), socioeconomic status (SES) is sensitive, academic success is the outcome, and standardized test scores (scores) are often deemed inadmissible because they are heavily influenced by SES. Students from high-SES backgrounds typically benefit from better preparation resources and repeated test attempts, so high scores do not always reflect ability alone. For students from low-SES backgrounds, a high score is often achieved under substantial challenges and may therefore reflect resilience, an unobserved trait that is highly predictive of academic success. Because resilience is not recorded in the data, existing methods cannot capture it, thereby overlooking such students' potential and weakening prediction.*

As these examples show, imperfections are inherent in real-world DBMS data. Existing fair data pre-processing methods are not designed for such cases and can fail when applied in practice, as shown in Section 6.2. Simple remedies such as imputing missing values or applying data augmentation do not account for fairness and may even amplify existing biases [29]. These limitations reveal a critical barrier to practically deploying fair data pre-processing in DBMSs and motivate the need for robust approaches that can operate under imperfect attribute spaces while still ensuring justifiable fairness and preserving data utility.

1.2 Contributions

Motivated by this, we present LatentPre, a justifiably fair data pre-processing framework that can robustly capture informative signals from raw data to support reliable and effective downstream decision-making, even in DBMS environments with an imperfect attribute space. To achieve this, LatentPre follows the standard *policy-then-adjust* paradigm, while departing from prior solutions through a fundamentally different policy formulation and a corresponding adjustment procedure. The key idea is to augment the hypothetically fair world with carefully introduced *latent attributes* that represent essential but unrecorded factors, rather than constructing it solely from the raw DBMS attribute space. This latent augmentation compensates for practical imperfections in the observed schema and enables LatentPre to recover and leverage predictive signals in a manner consistent with justifiable fairness.

We revisit Examples 1 and 2 to illustrate this key idea. In Example 1, LatentPre introduces a latent attribute, as shown in the bottom-left panel of Figure 1, to represent local economic conditions associated with zip code, thereby retaining forward-looking information about financial uncertainty. In Example 2, LatentPre similarly introduces a latent attribute, as shown in the bottom-right panel of Figure 1, to represent resilience, a legitimate factor underlying test scores that is otherwise absent from the schema. In both cases, LatentPre blocks the unfair pathway from the inadmissible attribute to the label and preserves utility through a latent attribute that is independent of sensitive attributes. As a result, it can reliably use potential decision-relevant information even when the attribute space is imperfect.

Challenges. While latent attributes are a well-established modeling tool, incorporating them into fair data pre-processing, especially under an imperfect attribute space, presents two significant challenges. First, determining how to introduce latent attributes and model their relationships

with observed attributes is non-trivial, as they must reliably capture fair yet decision-relevant signals that are often only implicitly encoded in the raw data via inadmissible attributes. Second, once the policy includes unobserved latent attributes, existing fair data adjustment techniques no longer apply. To our knowledge, no prior work has formally addressed these challenges in the pre-processing setting.

Technical overview. LatentPre is built on a set of tightly integrated technical components designed to address the above challenges. At the policy level, it augments the fair world with latent attributes L that influence a strategically selected subset of inadmissible attributes $\mathcal{I}_c$ and the label Y , while remaining independent of sensitive attributes $\mathcal{S}$. Intuitively, L models valuable predictive factors that are not recorded in the explicit schema but are indirectly reflected in the raw data through inadmissible attributes, which prior solutions cannot fully exploit without violating fairness constraints. This policy design enables LatentPre to uncover and leverage such legitimate signals while maintaining justifiable fairness.

However, a naive approach may yield little benefit if L is poorly placed or modeled. Without careful design, the policy formulation may not be provably effective, and the latent attributes could remain uninformative, introducing arbitrary or redundant structure. In essence, modeling L requires enough observable constraints to uniquely determine its effect, analogous to solving for unknowns requiring enough independent equations. In latent causal models, this requirement is captured by *identifiability* [2]. To avoid such degeneracy, we enforce identifiability through a principled placement strategy, which provably guarantees that the inserted latent attributes capture genuine decision-relevant signals. Specifically, our strategy positions L to intervene on $\mathcal{I}_c$ and Y , combined with a local pruning mechanism that minimally reshapes the causal structure surrounding $\mathcal{I}_c$ to meet identifiability conditions.

A remaining practical concern is that the augmented policy contains unobserved attributes, making direct data adjustment infeasible. To address this, we formulate the data adjustment process as a parameter estimation problem involving latent attributes, and efficiently solve it using an expectation-maximization (EM) paradigm tailored to the structure of our policy. Once the parameters are estimated, the optimal empirical distribution conforming to the policy is determined, and the raw data can then be adjusted accordingly to complete the fairness-aware processing.

Extensive experiments verify that LatentPre consistently achieves strong performance across diverse scenarios, demonstrating robustness to imperfect and complex real-world data environments.

Roadmap. The remainder of this paper is organized as follows. Section 2 introduces the required notions and background. Section 3 formalizes the fair data pre-processing problem under an imperfect attribute space. Sections 4 and 5 then present our proposed solution, LatentPre. Section 6 provides a comprehensive evaluation, examining the behavior of LatentPre across various scenarios. Finally, Section 7 reviews related work, and Section 8 concludes the paper.

2 Preliminaries

2.1 Causal DAGs

A causal DAG $\mathcal{G}$ is a directed acyclic graph over attributes $\mathcal{V} = \{V_1, \dots, V_d\}$ that encodes direct causal relationships among them. Each node corresponds to an attribute (viewed as a random variable in the underlying distribution), and each directed edge represents a direct causal effect. For each V_i , let Π_i denote its parent set in $\mathcal{G}$. The model associates V_i with a conditional distribution $\mathbb{P}[V_i \mid \Pi_i; \theta_i]$, where θ_i parameterizes the conditional probability table (CPT) of V_i given Π_i . The

joint distribution over $\mathcal{V}$ then admits the standard Bayesian network factorization [28]:

$$\mathbb{P}[\mathcal{V}; \theta] = \mathbb{P}[V_1, \dots, V_d; \theta] = \prod_{i=1}^d \mathbb{P}[V_i \mid \Pi_i; \theta_i], \quad (1)$$

where $\theta = (\theta_1, \dots, \theta_d)$ collects the parameters for all CPTs.

Beyond factorization, causal DAGs encode conditional independencies through d-separation [28]. For disjoint node sets $X, Y, Z \subseteq \mathcal{V}$, X and Y are *d-separated* by Z in $\mathcal{G}$ if every path between X and Y is blocked when conditioning on Z . A distribution $\mathbb{P}$ over $\mathcal{V}$ is *Markov compatible* with $\mathcal{G}$ if every d-separation in $\mathcal{G}$ implies the corresponding conditional independence in $\mathbb{P}$. Conversely, $\mathbb{P}$ is *faithful* to $\mathcal{G}$ if every conditional independence in $\mathbb{P}$ is implied by a d-separation in $\mathcal{G}$. Following prior work [12, 33, 37], we assume that the observed distribution $\mathbb{P}$ is Markov compatible with and faithful to the underlying causal DAG.

Causal DAGs support the definition of justifiable fairness [37] (introduced in the next section) through interventions, represented by the *do* operator [28]. For an attribute X and value x , the intervention $do(X = x)$ forces X to take value x and severs all incoming causal influences into X . The resulting interventional distribution $\mathbb{P}[O \mid do(X = x)]$ characterizes the causal effect of X on an outcome O . Graphically, $do(X = x)$ corresponds to removing all incoming edges to X and fixing its value.

2.2 Justifiable Fairness

Consider a classifier $\mathcal{M} : \text{Dom}(\mathcal{X}) \rightarrow \text{Dom}(O)$ that maps input features $\mathcal{X}$ to an outcome variable O , where $\text{Dom}(\cdot)$ denotes the domain. Let $\mathcal{S} \subseteq \mathcal{X}$ represent the set of sensitive attributes; then, justifiable fairness is defined as follows.

DEFINITION 1 ($\mathcal{K}$ -FAIRNESS [37]). For a subset $\mathcal{K} \subseteq \mathcal{X} \setminus \mathcal{S}$, we say that $\mathcal{M}$ is $\mathcal{K}$ -fair with respect to the sensitive set $\mathcal{S}$ if, for every assignment $\mathcal{K} = \kappa$, intervening on $\mathcal{S}$ does not affect the distribution of the outcome once $\mathcal{K}$ is fixed:

$$\mathbb{P}[O = o \mid do(\mathcal{S} = s_0), do(\mathcal{K} = \kappa)] = \mathbb{P}[O = o \mid do(\mathcal{S} = s_1), do(\mathcal{K} = \kappa)].$$

DEFINITION 2 (JUSTIFIABLE FAIRNESS [37]). Let $\mathcal{A} \subseteq \mathcal{X}$ denote the collection of admissible attributes, i.e., features that are permitted to influence O even if they are themselves affected by sensitive attributes. A classifier $\mathcal{M}$ is said to satisfy justifiable fairness if it is $\mathcal{K}$ -fair for every $\mathcal{K}$ with $\mathcal{A} \subseteq \mathcal{K} \subseteq \mathcal{X}$.

To isolate the effect of data pre-processing from that of model training, we follow prior work [12, 37, 55] and assume a *reasonable* classifier, i.e., one that closely approximates the data distribution on which it is trained. The following corollary and proposition provide sufficient structural conditions on the data under which any reasonable classifier satisfies justifiable fairness.

COROLLARY 1 ([37]). Let $\mathcal{G}$ be the attribute graph derived from a dataset. If all causal pathways from sensitive attributes to the label pass through at least one admissible attribute, then any reasonable classifier trained on this dataset is regarded as justifiably fair.

PROPOSITION 1 ([37, 55]). Let $\mathcal{D}$ be a database instance and $\mathcal{G}$ its attribute graph over the attribute set $\mathcal{V}$. Under attribute specification, $\mathcal{V}$ is fully partitioned into five disjoint subsets: $\mathcal{S}$ (sensitive), $\mathcal{I}$ (inadmissible), $\mathcal{A}$ (admissible), $\mathcal{W}$ (additional), and $\mathcal{Y}$ (label). Here $\mathcal{W}$ refers to additional attributes that are neither sensitive nor causally relevant to fairness. If, in $\mathcal{G}$, every edge directed toward the label attribute in $\mathcal{Y}$ originates exclusively from attributes in $\mathcal{A} \cup \mathcal{W}$, then any reasonable classifier trained on $\mathcal{D}$ respects justifiable fairness. Equivalently, the condition holds whenever the parent set Π of the label attribute is fully contained in $\mathcal{A} \cup \mathcal{W}$, i.e., $\Pi \subseteq \mathcal{A} \cup \mathcal{W}$.

2.3 Latent Model and Identifiability

Causal DAGs offer a principled way to model dependencies among observed attributes, but in many cases, there exist hidden or unmeasured factors that also affect the outcomes. To consider such factors, let L denote an unobserved latent attribute in addition to the observed attributes $\mathcal{V}$. A latent causal model then specifies a joint distribution over $(\mathcal{V}, L)$, written as $\mathbb{P}[\mathcal{V}, L; \theta]$, where $\theta = (\theta_1, \dots, \theta_d, \theta_l)$ denotes the model parameters. The observed distribution is obtained by marginalizing out L :

$$\mathbb{P}[\mathcal{V}; \theta] = \sum_L \mathbb{P}[\mathcal{V}, L; \theta],$$

where the summation is taken over all possible states of L .

Since L is unobserved, its role must be inferred from the observed data. This naturally raises the question of whether the model parameters can be reasonably determined from the observed data without ambiguity, a property known as *identifiability*. When identifiability holds, the estimated parameter θ faithfully represents the true data-generating process, enabling reliable causal reasoning. The concept is formally defined as follows.

DEFINITION 3 (IDENTIFIABILITY [2]). Consider a latent causal model parameterized by $\theta \in \Theta$, where Θ is the parameter space. Let $\mathbb{P}[\mathcal{V}, \mathcal{L}; \theta]$ denote the joint distribution induced by θ , and let $\mathbb{P}[\mathcal{V}; \theta]$ denote the corresponding observed distribution obtained by marginalizing out $\mathcal{L}$. The model is said to be identifiable if

$$\forall \theta, \theta' \in \Theta, \quad \mathbb{P}[\mathcal{V}; \theta] = \mathbb{P}[\mathcal{V}; \theta'] \Rightarrow \theta = \theta'.$$

Equivalently, the mapping $\theta \mapsto \mathbb{P}[\mathcal{V}; \theta]$ is injective, so the parameters are uniquely determined by the observed distribution.

While strict identifiability provides a clean theoretical guarantee, it is often overly restrictive for latent causal models. In general, it fails over the full parameter space [2, 3], because degenerate parameter configurations may render distinct models observationally indistinguishable, and inherent symmetries such as latent-state label permutations do not affect the observed distribution. This motivates weaker but practically sufficient notions of identifiability, introduced below.

DEFINITION 4 (GENERIC IDENTIFIABILITY [2]). A latent causal model is said to be generically identifiable if there exists a subset $\Omega \subseteq \Theta$ of Lebesgue measure zero such that

$$\forall \theta, \theta' \in \Theta \setminus \Omega, \quad \mathbb{P}[\mathcal{V}; \theta] = \mathbb{P}[\mathcal{V}; \theta'] \Rightarrow \theta = \theta'.$$

Equivalently, the mapping $\theta \mapsto \mathbb{P}[\mathcal{V}; \theta]$ is injective almost everywhere, except on a measure-zero subset of Θ .

DEFINITION 5 (GENERIC IDENTIFIABILITY UP TO LABEL SWAPPING [2]). Let Π be the set of all permutations of the state labels of $\mathcal{L}$, and for $\pi \in \Pi$, let $\pi(\theta)$ denote the parameter vector obtained by applying π to the state labels of $\mathcal{L}$ in all relevant CPTs. A latent causal model is said to be generically identifiable up to label swapping if there exists a subset $\Omega \subseteq \Theta$ of Lebesgue measure zero such that

$$\forall \theta, \theta' \in \Theta \setminus \Omega, \quad \mathbb{P}[\mathcal{V}; \theta] = \mathbb{P}[\mathcal{V}; \theta'] \Rightarrow \exists \pi \in \Pi \text{ such that } \theta' = \pi(\theta).$$

That is, for almost all parameter values, the observed distribution determines the parameters uniquely, up to a permutation of the latent variable's state labels.

In this paper, we focus on the notion of *Generic Identifiability up to Label Swapping*, which generally does not diminish the practical utility of a model and is sufficient for most applications [2, 3]. For convenience, unless otherwise specified, we use the terms *identifiability* or *generic identifiability* interchangeably to refer to *Generic Identifiability up to Label Swapping*.

2.4 Information Measures

We next introduce several information measures [9] that underpin our analysis. Let X , Y , and Z denote random variables and $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ denote random variable sets. The entropy of X measures the uncertainty associated with its distribution:

$$H(X) = - \sum_x \mathbb{P}[X = x] \log \mathbb{P}[X = x].$$

The conditional entropy of X given Y quantifies the average uncertainty remaining in X once Y is known:

$$H(X | Y) = - \sum_{x,y} \mathbb{P}[X = x, Y = y] \log \mathbb{P}[X = x | Y = y].$$

The mutual information (MI) between X and Y captures the reduction in uncertainty of one variable upon observing the other:

$$I(X; Y) = H(X) - H(X | Y) = H(Y) - H(Y | X).$$

A higher value indicates stronger dependence between X and Y .

The conditional mutual information (CMI) between X and Y given Z extends MI by measuring the remaining dependence between X and Y after accounting for Z :

$$I(X; Y | Z) = H(X | Z) - H(X | Y, Z) = H(Y | Z) - H(Y | X, Z).$$

CMI is nonnegative and equals zero if and only if X and Y are conditionally independent given Z .

For random variable sets $\mathcal{X} = \{X_1, \dots, X_m\}$ and $\mathcal{Y} = \{Y_1, \dots, Y_n\}$, the conditional mutual information between $\mathcal{X}$ and $\mathcal{Y}$ given $\mathcal{Z}$ can be decomposed using the chain rule:

$$I(\mathcal{X}; \mathcal{Y} | \mathcal{Z}) = \sum_{i=1}^m \sum_{j=1}^n I(X_i; Y_j | \mathcal{X}_{<i}, \mathcal{Y}_{<j}, \mathcal{Z}), \quad (2)$$

where $\mathcal{X}_{<i} = \{X_1, \dots, X_{i-1}\}$ and $\mathcal{Y}_{<j} = \{Y_1, \dots, Y_{j-1}\}$.

3 Problem Statement

Given a database $\mathcal{D}$ with d attributes $\mathcal{V} = \{V_1, \dots, V_{d-1}, Y\}$, where Y denotes the label attribute, the remaining attributes $\mathcal{V} \setminus \{Y\}$ are partitioned into four disjoint sets. The set $\mathcal{S}$ contains sensitive attributes such as gender or race, and $\mathcal{I}$ includes inadmissible attributes that directly encode sensitive information. Under justifiable fairness, neither $\mathcal{S}$ nor $\mathcal{I}$ is allowed to have a direct causal effect on the decision. The set $\mathcal{A}$ contains admissible attributes whose influence on the decision is considered legitimate even if affected by $\mathcal{S}$. The set $\mathcal{W}$ includes additional attributes unrelated to sensitive information and can therefore be used as decision factors.

We consider a practical setting where the attribute space of the database $\mathcal{D}$ is imperfect due to attribute ambiguity and attribute absence, as discussed in Section 1.1. Our objective is to design a data pre-processing framework that calibrates the empirical distribution of $\mathcal{D}$ such that any reasonable downstream predictive model trained on the processed data satisfies justifiable fairness while preserving effectiveness.

4 LatentPre

In this section, we introduce our latent-augmented framework, LatentPre. In a nutshell, LatentPre follows the *policy-then-adjust* paradigm and operates in two stages. It first specifies a fairness policy that encodes domain knowledge and fairness requirements, and then adjusts the dataset to satisfy this policy with minimal distributional distortion while preserving data utility.

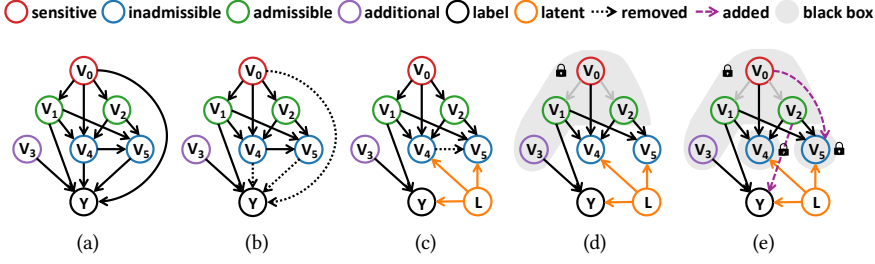


Fig. 2. Framework construction and refinement process.

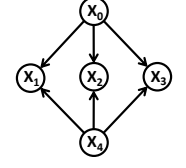


Fig. 3. The generalized model.

4.1 Latent-Augmented Fairness Policy

LatentPre instantiates the policy using a causal DAG. Starting from the raw-data DAG, LatentPre refines the structure through a sequence of steps that adjust causal connections and introduce a controlled interaction between a latent attribute and observed attributes. For clarity, we temporarily assume that the raw-data DAG is available. In Section 4.2, we describe how to construct the policy when such prior knowledge is unavailable.

Given the raw-data DAG in Figure 2a, we first refine it to retain only justifiably fair causal pathways, following Proposition 1. Specifically, we remove all directed edges from sensitive and inadmissible attributes to the label, as shown in Figure 2b. This yields a fair policy in which sensitive attributes influence the label only through admissible attributes. We preserve this structure in subsequent steps, so the fairness guarantee holds throughout.

Next, we augment the policy by introducing a latent attribute to capture utility-relevant signals not fully represented in the observed attributes. The key idea is to model hidden factors that affect both inadmissible attributes $\mathcal{I}$ and the label Y while remaining independent of sensitive attributes. This design treats inadmissible attributes as jointly determined by observed sensitive factors and an unobserved latent factor, allowing L to extract decision-relevant signals. Since L is independent of the sensitive attributes by design, the fairness guarantees established above are preserved.

First-cut solution. A straightforward approach is to model L as a source node that affects all inadmissible attributes $\mathcal{I}$ and the label Y . However, connecting L to the entire set $\mathcal{I}$ substantially increases model complexity and makes data calibration computationally expensive when $|\mathcal{I}|$ is large. We therefore adopt a more fine-grained design by allowing L to affect only the label and a selected subset of inadmissible attributes, namely the direct parents of Y , denoted by $\mathcal{I}_c := \mathcal{I} \cap \Pi_Y$. The intuition is that, under the local Markov property and Markov compatibility, the effect of $\mathcal{I}$ on Y is mainly captured through $\mathcal{I}_c$, as reflected in Equation (1) and supported by the analysis in Equation (3) in the next subsection. Modeling L as connecting only to $\mathcal{I}_c$ thus preserves sufficient decision-relevant information while keeping data calibration scalable.

Refinement for identifiability. A subtle yet critical limitation of the first-cut solution is that it overlooks *identifiability*, that is, the ability to uniquely infer the effect of the latent attribute from observed data. Without identifiability, L may capture redundant or unstable signals. This issue becomes particularly severe when the attributes in $\mathcal{I}_c$ are tightly connected, as their information may then collapse into a single shared source of variation. In such cases, L behaves as if it interacts with only one inadmissible attribute and the label, weakening the statistical basis for estimating L and leading to ambiguous or meaningless contributions.

To tackle this, we introduce a pruning mechanism that reshapes the local causal structure around $\mathcal{I}_c$ to improve identifiability. Since adopting a fairness policy already removes all direct causal effects

from inadmissible attributes to the label, it prevents the information in $\mathcal{I}_c$ and Y from collapsing into a single source. Building on this property, it suffices to further limit dependence within $\mathcal{I}_c$. Accordingly, LatentPre partitions $\mathcal{I}_c$ into two conditionally independent subsets $\mathcal{I}_{c_1}$ and $\mathcal{I}_{c_2}$, and eliminates direct interconnections between them. The latent attribute L then points to both subsets and to the label Y , as illustrated in Figure 2c. Formal identifiability analysis and algorithmic details are provided in Section 5.

Beyond identifiability and implementability, this design also provides an alternative view of robustness. By distributing fairness-aware adjustments across the DAG, it avoids making any single region overly sensitive to imperfections in attribute specification. As shown in Figure 2b, fairness is enforced by severing all edges from $\mathcal{S} \cup \mathcal{I}$ to Y , localizing modifications near the label. In contrast, Figure 2c introduces the latent variable L to mediate the influence of $\mathcal{I}_c$ on Y , preserving utility-relevant signals while adjusting local dependencies among $\{V_4, V_5\}$. This redistribution spreads the impact of fairness constraints more evenly, preventing structural collapse and maintaining overall stability and utility.

4.2 DAG Refinement

A key challenge in applying this policy is that the underlying DAG is rarely available in practice [30, 37]. While causal discovery methods such as Max-Min Hill Climbing (MMHC) [43] can estimate the DAG structure, their computational cost grows exponentially with the number of attributes [43], posing serious scalability concerns. This motivates a more efficient procedure for extracting only the structural information needed for policy enforcement.

We address this challenge using a two-step refinement strategy inspired by CausalPre [55], but tailored to support latent attribute integration. The key idea is to construct a coarse-grained fairness policy that avoids full DAG reconstruction while preserving causally fair information.

In the first step, we identify relationships that remain unchanged throughout processing and abstract them as a black box. Since these structures require no modification, the policy only needs to indicate that they are preserved, without specifying their internal form. As discussed in Section 4.1 and formalized in Proposition 1, fairness-aware pre-processing only modifies the structure around the label Y . To recover hidden utility-relevant signals, our framework further updates the region around $\mathcal{I}_c$. These localized adjustments confine the scope of modification to the neighborhoods of Y and $\mathcal{I}_c$, leaving the rest intact. We therefore treat these preserved relationships among $\mathcal{V} \setminus (\mathcal{I}_c \cup Y)$ collectively as a black box, yielding the simplification from Figure 2c to Figure 2d. In this representation, only the edges directed to Y and $\mathcal{I}_c$ (shown as solid black arrows) are explicitly encoded in the fairness policy, while all others are abstracted away. Although the figure shows only a few attributes, in practice, most structural information is absorbed into the black-box region, substantially reducing computational cost.

In the second step, we refine the structure around Y and $\mathcal{I}_c$ by greedily expanding their parent sets, avoiding expensive exact parent computation. As shown in [55], this does not compromise informational utility. For Y , we admit all attributes in $\mathcal{A} \cup \mathcal{W}$ as potential parents. For each $V_i \in \mathcal{I}_c$, its parents are drawn from $\mathcal{S} \cup \mathcal{I} \cup \mathcal{A}$, ensuring that sensitive information comes from observable sources and that the latent attribute reflects only unbiased signals.

However, since each $V_i (\in \mathcal{I}_c)$ also belongs to $\mathcal{I}$, naive parent selection may introduce self-loops or cycles. To avoid this, we further abstract the relationships within $\mathcal{I}_c$. Let $\mathcal{I}_o := \mathcal{I} \setminus \mathcal{I}_c$ denote the remaining inadmissible attributes. Because dependencies within each partition of $\mathcal{I}_c$ remain locally unchanged during construction, we treat them as two additional black boxes. Each black box then admits $\mathcal{S} \cup \mathcal{I}_o \cup \mathcal{A}$ as its effective parent set, preventing cyclic dependencies while preserving necessary information flow. This yields the coarse-grained representation in Figure 2e.

Algorithm 1: DATAPREPROCESSING

Input: Database $\mathcal{D}$ with attribute set $\mathcal{V} = \mathcal{S} \cup \mathcal{I} \cup \mathcal{A} \cup \mathcal{W} \cup \{Y\}$, the number of latent states τ , maximum number of iterations n , convergence threshold η , maximum size of conditioning set α

Output: Processed data $\mathcal{D}'$

```

1  $\mathcal{I}_c \leftarrow \text{IDENTIFICATION}(\mathcal{D}, \alpha);$  // we set  $\alpha = 2$ 
2  $\mathcal{I}_o \leftarrow \mathcal{I} \setminus \mathcal{I}_c;$ 
3  $\mathcal{I}_{c_1}, \mathcal{I}_{c_2} \leftarrow \text{PARTITION}(\mathcal{D}, \mathcal{I}_c, \mathcal{S} \cup \mathcal{I}_o \cup \mathcal{A}, \tau);$ 
4  $\theta^* \leftarrow \text{PARAMETERESTIMATION}(\mathcal{D}, \mathcal{I}_{c_1}, \mathcal{I}_{c_2}, n, \eta);$ 
5  $k \leftarrow |\mathcal{D}|;$ 
6 for  $\theta_i \in \theta^*$  do
7   Sample  $k$  values for attributes in set “ $i$ ” from distribution  $\mathbb{P}_{\theta_i}$ , and fill the corresponding columns
   in  $\mathcal{D}'$ ;
8 Marginalize out latent attribute  $L$  from  $\mathcal{D}'$ ;
9 return  $\mathcal{D}'$ ;
```

Although this refinement may introduce additional dependencies to Y and $\mathcal{I}_c$, these effects are acceptable. For Y , the added decision factors are admissible by definition. For $\mathcal{I}_c$, its influence on Y is already blocked, thus new dependencies do not affect decision-making or violate fairness. While this may slightly distort the local distribution around $\mathcal{I}_c$, it enables a more faithful reconstruction of the distribution around Y through the latent variable L . Experimental results in Section 6 confirm that this trade-off yields clear performance gains. Moreover, such additional dependencies are rare in practice, as real-world datasets typically exhibit rich attribute interactions that already account for most dependencies.

The approximated distribution under the refined coarse-grained DAG takes the general form:

$$\mathbb{P}[L, \mathcal{V}] \approx \mathbb{P}[\mathcal{V} \setminus (\mathcal{I}_c \cup \{Y\})] \cdot \mathbb{P}[L] \cdot \mathbb{P}[\mathcal{I}_{c_1} \mid \Pi'_{c_1}] \cdot \mathbb{P}[\mathcal{I}_{c_2} \mid \Pi'_{c_2}] \cdot \mathbb{P}[Y \mid \Pi'_Y], \quad (3)$$

where $\Pi'_{c_1} = \{L, \mathcal{S}, \mathcal{I}_o, \mathcal{A}\}$, $\Pi'_{c_2} = \{L, \mathcal{S}, \mathcal{I}_o, \mathcal{A}\}$, and $\Pi'_Y = \{L, \mathcal{A}, \mathcal{W}\}$ represent the greedy fair parent sets of $\mathcal{I}_{c_1}$, $\mathcal{I}_{c_2}$, and Y , respectively.

4.3 The Complete Framework

With the refined fairness policy in place, LatentPre adjusts the raw data to satisfy its structural constraints while preserving the original distribution as much as possible. This amounts to estimating the optimal parameters θ^* for the refined DAG that best approximate the raw data distribution. Details of the implementation are provided in Section 5.4.

Algorithm 1 summarizes the full processing pipeline. Given a raw dataset $\mathcal{D}$, LatentPre first identifies the set $\mathcal{I}_c$ and partitions it to construct the latent-augmented fairness policy based on Equation (3). It then estimates the optimal parameter set θ^* under this policy and treats the induced distribution as the target for data processing. Given θ^* , LatentPre samples each attribute set “ i ” from its factor distribution $\mathbb{P}_{\theta_i}$ and overwrites the corresponding columns in the dataset. After marginalizing out the latent attribute L , it outputs the processed dataset $\mathcal{D}'$.

5 Algorithmic Design and Analysis

This section presents the theoretical analysis and algorithmic design of LatentPre. We first establish the identifiability of the latent-augmented DAG in Section 5.1. We then describe the algorithms for identifying and partitioning the attribute set $\mathcal{I}_c$ in Sections 5.2 and 5.3. Finally, in Section 5.4, we describe the parameter estimation procedure for enforcing the policy and obtaining the optimal parameters θ^* for data adjustment.

5.1 Identifiability Analysis

To enable formal identifiability analysis, we simplify the DAG by grouping related attributes into composite units [14, 21, 51] and treating each composite as a single clumped attribute [2]. The resulting generalized model is shown in Figure 3, where X_0 denotes the latent attribute L , X_1 and X_2 represent the two disjoint subsets $\mathcal{I}_{c_1}$ and $\mathcal{I}_{c_2}$, X_3 denotes the label Y , and X_4 aggregates $\mathcal{S} \cup \mathcal{I}_0 \cup \mathcal{A}$. The following theorem shows that, under mild conditions, this generalized model is identifiable, from which the identifiability of LatentPre follows directly.

THEOREM 1. *Consider a latent causal model with one latent attribute X_0 and $d \geq 4$ observed attributes. The observed attributes are partitioned into four nonempty groups X_1, X_2, X_3, X_4 , such that conditional on (X_0, X_4) the groups X_1, X_2, X_3 are mutually independent, and moreover $X_0 \perp X_4$. The latent attribute X_0 takes $\tau \geq 2$ states with strictly positive mixture weights. For each group X_i with $i \in \{1, 2, 3, 4\}$, define its group cardinality as*

$$\kappa_i := \prod_{j \in X_i} k_j,$$

where k_j is the number of states of the attribute j .

Assume $\kappa_i \geq 2$ for all i . If τ does not exceed the cardinality of at least two of the groups among X_1, X_2, X_3 , that is, if

$$\tau \leq \min\{\max(\kappa_1, \kappa_2), \max(\kappa_2, \kappa_3), \max(\kappa_1, \kappa_3)\}, \quad (4)$$

then the model parameters are generically identifiable up to a permutation of the latent states.

PROOF. Let θ_i denote the local parameter at X_i for $i \in \{0, 1, 2, 3, 4\}$. We aim to show that the local parameters θ_i of each X_i are generically identifiable up to a global permutation of the latent states.

Fix any value $j \in \{1, \dots, \kappa_4\}$ of X_4 . Conditioning on $X_4 = j$, we obtain a latent class model with three observed attribute groups (X_1, X_2, X_3) and one latent variable X_0 . For $i = 0, 1, 2, 3$, denote these conditioned attributes or groups by $X_i^{(j)}$ and let $\theta_i^{(j)}$ denote their associated local parameters.

By assumption on the group cardinalities, at least two of $\kappa_1, \kappa_2, \kappa_3$ are not smaller than τ . Without loss of generality, assume $\kappa_1, \kappa_2 \geq \tau$. Then, for each conditional model $(X_0^{(j)}, X_1^{(j)}, X_2^{(j)}, X_3^{(j)})$, the identifiability condition in Equation (4), derived from tensor decomposition [2, 3], is satisfied. It therefore follows that the parameters $(\theta_0^{(j)}, \theta_1^{(j)}, \theta_2^{(j)}, \theta_3^{(j)})$ are generically identifiable, up to a permutation π_j of the latent states.

Since the mixing distribution of X_0 does not depend on X_4 , we have $\theta_0^{(j)} = \theta_0$ for all j . This implies that the permutations π_j obtained for different j must be consistent; otherwise, they would yield distinct versions of θ_0 , contradicting uniqueness. Therefore, all π_j coincide with a single global permutation π . Consequently, $(\theta_0, \theta_1, \theta_2, \theta_3)$ are generically identifiable up to a global permutation of the latent states. Finally, the parameters θ_4 are directly estimated from the conditional distributions $\mathbb{P}[X_4]$, which are part of the observed distribution. This completes the proof. $\square$

In addition to identifiability, Theorem 1 provides a practical upper bound on the number of latent states. Since the label domain is often small, we conservatively set

$$\tau \leq \min(|\mathcal{I}_{c_1}|, |\mathcal{I}_{c_2}|). \quad (5)$$

5.2 Identification of $\mathcal{I}_c$

To construct the latent-augmented fairness policy, we must identify the attribute set $\mathcal{I}_c$, as mentioned in Section 4.1. A natural approach is to first recover the parent set Π_Y of the label Y and then compute $\mathcal{I}_c = \mathcal{I} \cap \Pi_Y$. However, this requires $O(d \cdot 2^d)$ conditional independence (CI) tests, since

Algorithm 2: IDENTIFICATION**Input:** Database $\mathcal{D}$ with attribute set $\mathcal{V} = \mathcal{S} \cup \mathcal{I} \cup \mathcal{A} \cup \mathcal{W} \cup \{Y\}$, maximum size of conditioning set α **Output:** Inadmissible subset $\mathcal{I}_c$

```

1  Let  $\mathcal{I}_c \leftarrow \mathcal{I}$ ;
2  for  $m \leftarrow 0$  to  $\alpha$  do
3      for each attribute  $X \in \mathcal{I}_c$  do
4          if  $m = 0$  then
5              Test if  $X \perp Y$  using chi-square test;
6              If independent: Remove  $X$  from  $\mathcal{I}_c$ ;
7          else
8               $\mathcal{Z} \leftarrow$  Construct conditioning sets of size  $m$  from  $\mathcal{V} \setminus \{X, Y\}$ ;
9              Test if  $X \perp Y \mid Z$  using G-test, where  $Z \in \mathcal{Z}$ ;
10             If independent for any  $Z$ : Remove  $X$  from  $\mathcal{I}_c$  and break;
11 return  $\mathcal{I}_c$ ;

```

for each attribute $V_i \in \mathcal{V} \setminus Y$, we must evaluate whether $V_i \perp Y \mid \Omega$ for all subsets $\Omega \subseteq \mathcal{V} \setminus \{V_i, Y\}$. This exponential cost is prohibitive in practice.

Given that the inadmissible set is typically much smaller than the full attribute set, we instead test each inadmissible attribute individually to determine whether it is a parent of Y . This reduces the number of CI tests to $O(|\mathcal{I}| \cdot 2^d)$. Although still exponential, empirical studies [16, 39] show that limiting the size of conditioning sets to 2 or 3 is usually sufficient in practice, which reduces the overall cost to $O(|\mathcal{I}| \cdot d)$. In this paper, we set the maximum size of the conditioning set, denoted by α , to 2 by default.

Algorithm 2 outlines the identification procedure, which consists of two steps: (i) test whether a pathway exists from X to Y , and (ii) if so, determine whether it is direct. In the initial step (Lines 5-6), we apply the chi-square test to evaluate whether an inadmissible attribute X is independent of Y . If independence is confirmed, we eliminate X from further consideration. Otherwise, we proceed to the second step (Lines 8-10), using the G-test to evaluate whether the dependence persists after conditioning on various subsets Z . If the dependency holds across all such Z , we conclude that no set blocks the path, indicating a direct causal link. This identification process is computationally efficient due to the linear number of tests and the early-stop mechanism.

5.3 Partition of $\mathcal{I}_c$

After identifying the set $\mathcal{I}_c$, we partition it into two disjoint and non-empty subsets, $\mathcal{I}_{c_1}$ and $\mathcal{I}_{c_2}$, to enable reasonable integration of the latent attribute. The subsets $\mathcal{I}_{c_1}$ and $\mathcal{I}_{c_2}$ are required to be conditionally independent given $\mathcal{Z} = \mathcal{S} \cup \mathcal{I}_o \cup \mathcal{A}$ to satisfy the identifiability condition described in Section 5.1. To preserve as much information as possible, we formulate the partitioning process as a constrained optimization problem that aims to find two sufficiently large subsets with minimal dependence under the given context:

$$\min_{\mathcal{I}_{c_1}, \mathcal{I}_{c_2}} I(\mathcal{I}_{c_1}; \mathcal{I}_{c_2} \mid \mathcal{Z}) \quad \text{s.t.} \quad |\mathcal{I}_{c_1}| \geq \tau, |\mathcal{I}_{c_2}| \geq \tau,$$

where τ represents the number of latent states and the constraint ensures that each subset is large enough for the latent attribute to represent meaningful states, as bounded by Equation (5).

This optimization is computationally intractable due to two challenges: computing multivariate conditional mutual information (MCMI) is costly, and the number of possible bipartitions grows exponentially with the size of $\mathcal{I}_c$. To address this, we apply two approximations. First, we estimate

Algorithm 3: PARTITION

Input: Database $\mathcal{D}$, inadmissible subset $\mathcal{I}_c$, conditioning set $\mathcal{Z}$, the number of latent states τ , relative tolerance ε

Output: Subsets $\mathcal{I}_{c_1}$ and $\mathcal{I}_{c_2}$

```

1 Initialize  $\mathcal{I}_{c_1}, \mathcal{I}_{c_2} \leftarrow$  random bipartition of  $\mathcal{I}_c$ ;
2 Check feasibility: if  $|\mathcal{I}_{c_1}| < \tau$  or  $|\mathcal{I}_{c_2}| < \tau$ , repeatedly move a randomly selected attribute  $v$  from the
   larger subset to the smaller one until both subsets satisfy the cardinality threshold;
3  $n \leftarrow I(\mathcal{I}_{c_1}; \mathcal{I}_{c_2} \mid \mathcal{Z})$  (as shown in Equation (6)),  $n_0 \leftarrow n$ ;
4 while true do
5   Initialize  $\Delta^* \leftarrow 0$ ,  $(A^*, B^*) \leftarrow \perp$ ;
6   // (1) Apply the best 1-move
7   for  $v \in \mathcal{I}_c$  do
8     Form candidate partition  $(A, B)$  by moving  $v$  to the opposite partition;
9     if  $|A| \geq \tau, |B| \geq \tau$  then
10      Compute  $n' \leftarrow I(A; B \mid \mathcal{Z})$ ,  $\Delta \leftarrow n - n'$ ;
11      if  $\Delta > \Delta^*$  then
12        Update  $\Delta^*, (A^*, B^*)$ ;
13   // (2) No improving 1-move: try the best 2-swap
14   if  $\Delta^* = 0$  then
15     for  $(u, v) \in \mathcal{I}_{c_1} \times \mathcal{I}_{c_2}$  do
16       Form candidate partition  $(A, B)$  by moving  $u$  and  $v$  to the opposite partition;
17       if  $|A| \geq \tau, |B| \geq \tau$  then
18         Compute  $n' \leftarrow I(A; B \mid \mathcal{Z})$ ,  $\Delta \leftarrow n - n'$ ;
19         if  $\Delta > \Delta^*$  then
20           Update  $\Delta^*, (A^*, B^*)$ ;
21   if  $\Delta^* < \varepsilon n$  or  $n \leq \varepsilon n_0$  then
22     break
23   Update  $(\mathcal{I}_{c_1}, \mathcal{I}_{c_2}) \leftarrow (A^*, B^*)$ ,  $n \leftarrow n - \Delta^*$ ;
24 return  $\mathcal{I}_{c_1}, \mathcal{I}_{c_2}$ 

```

the MCMC using the sum of pairwise conditional dependencies (i.e., CMI) [8] as shown below:

$$I(\mathcal{I}_{c_1}; \mathcal{I}_{c_2} \mid \mathcal{Z}) \approx \sum_{X \in \mathcal{I}_{c_1}} \sum_{Y \in \mathcal{I}_{c_2}} I(X; Y \mid \mathcal{Z}). \quad (6)$$

This captures the essential cross-set dependencies while remaining computationally efficient. Second, we employ a greedy hill-climbing strategy to search for a near-optimal partition, as described in Algorithm 3. The algorithm begins with a random bipartition of $\mathcal{I}_c$ that satisfies the cardinality constraint (Lines 1–2), then iteratively refines it through two steps. In the first step (1-move), it considers moving a single attribute from one group to the other and selects the move that most reduces the estimated CMI (Lines 6–11). If no such move is valid, it performs the second step (2-swap) by exchanging one attribute from each group (Lines 12–18). This exchange step is particularly helpful when the partition is close to the cardinality threshold and single moves would violate the constraint. The algorithm repeats these updates until no significant CMI reduction is observed, as determined by the relative tolerance parameter ε . The parameter ε controls the trade-off between runtime and partition quality, and we set $\varepsilon = 10^{-5}$ in experiments to ensure

high-quality partitions with reasonable runtime. The final output is a partition $(\mathcal{I}_{c_1}, \mathcal{I}_{c_2})$, which serves as the basis for policy construction.

Complexity analysis. Let $m = |\mathcal{I}_c|$. As described, the while loop updates the partition $(\mathcal{I}_{c_1}, \mathcal{I}_{c_2})$ only when the CMI improvement satisfies $\Delta^* \geq \varepsilon n$ and stops once $n \leq \varepsilon n_0$. Since each accepted update reduces the estimated CMI value by at least a multiplicative factor of $(1 - \varepsilon)$, the loop performs at most $O(\log(\varepsilon)/\log(1 - \varepsilon)) = O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ iterations. In each iteration, the dominant cost comes from the 2-swap step, which considers at most $|\mathcal{I}_{c_1}| |\mathcal{I}_{c_2}| \leq m^2/4$ candidate swaps. For each candidate partition, evaluating Equation (6) involves $O(m^2)$ pairwise CMI evaluations. Therefore, each iteration requires at most $O(m^4)$ pairwise CMI evaluations, and the overall procedure requires at most $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon} \cdot m^4)$ such evaluations.

5.4 Parameter Estimation

Once the fairness policy is established, we adjust the raw data to satisfy its structural constraints while minimizing distortion to the underlying distribution, thereby completing the fairness-aware pre-processing. Equivalently, we seek the parameter setting θ^* of the refined DAG structure that is most faithful to the original data distribution among all feasible configurations. This can be cast as a maximum likelihood estimation (MLE) problem:

$$\theta^* = \arg \max_{\theta} \log \sum_L \mathbb{P}[\mathcal{V}, L \mid \theta]. \quad (7)$$

Because the latent variable L appears inside the logarithm, direct maximization is intractable. To address this, we introduce an auxiliary distribution $\mathbb{P}[L \mid \mathcal{V}, \hat{\theta}]$ with fixed parameter $\hat{\theta}$ and apply Jensen's inequality to derive a tractable lower bound:

$$\begin{aligned} \log \sum_L \mathbb{P}[\mathcal{V}, L \mid \theta] &= \log \mathbb{E}_{L \sim \mathbb{P}[L \mid \mathcal{V}, \hat{\theta}]} \left[\frac{\mathbb{P}[\mathcal{V}, L \mid \theta]}{\mathbb{P}[L \mid \mathcal{V}, \hat{\theta}]} \right] \\ &\geq \mathbb{E}_{L \sim \mathbb{P}[L \mid \mathcal{V}, \hat{\theta}]} \left[\log \mathbb{P}[\mathcal{V}, L \mid \theta] \right] + H \left[\mathbb{P}[L \mid \mathcal{V}, \hat{\theta}] \right], \end{aligned} \quad (8)$$

where $H[\cdot]$ denotes entropy. Since the second term is constant with respect to θ , maximizing the lower bound reduces to

$$\theta^* = \arg \max_{\theta} Q(\theta \mid \hat{\theta}), \quad (9)$$

where

$$Q(\theta \mid \hat{\theta}) := \mathbb{E}_{L \sim \mathbb{P}[L \mid \mathcal{V}, \hat{\theta}]} \left[\log \mathbb{P}[\mathcal{V}, L \mid \theta] \right].$$

The validity and convergence of this reformulation follow the standard analysis in [10, 46].

If L were observed, Equation (9) could be optimized directly. Since L is latent, we treat it as a column of missing data and adopt the Expectation-Maximization (EM) paradigm to jointly estimate both L and θ . The procedure alternates between two steps. In the E-step, we consider the current parameter $\hat{\theta}$ to be fixed and estimate the distribution of L , which provides a probabilistic imputation of the missing values. In the M-step, we treat the data as complete and optimize $Q(\theta \mid \hat{\theta})$ to update θ . This process repeats until convergence, yielding the final estimate $\theta^* = \theta^{(T)}$ after T iterations. Both steps are tailored to exploit the structural properties of our refined fairness policy, which is key to achieving efficient computation.

5.4.1 Analysis of E-step. At k -th iteration, parameters are fixed as $\hat{\theta}^{(k)}$, initialized randomly for $k=1$ and updated as $\hat{\theta}^{(k)} := \theta^{(k-1)}$ if $k>1$. The goal is to infer the posterior of L given $\mathcal{V}$:

$$\mathbb{P}[L \mid \mathcal{V}; \hat{\theta}^{(k)}] = \frac{\mathbb{P}[\mathcal{V}, L; \hat{\theta}^{(k)}]}{\mathbb{P}[\mathcal{V}; \hat{\theta}^{(k)}]} \propto \mathbb{P}[\mathcal{V}, L; \hat{\theta}^{(k)}].$$

By factorizing the refined DAG, as shown in Equation (3), we obtain:

$$\mathbb{P}[L \mid \mathcal{V}; \hat{\theta}^{(k)}] \propto \mathbb{P}[L; \hat{\theta}^{(k)}] \cdot \mathbb{P}[\mathcal{I}_{c_1} \mid \Pi'_{c_1}; \hat{\theta}^{(k)}] \cdot \mathbb{P}[\mathcal{I}_{c_2} \mid \Pi'_{c_2}; \hat{\theta}^{(k)}] \cdot \mathbb{P}[Y \mid \Pi'_Y; \hat{\theta}^{(k)}]. \quad (10)$$

Since only four terms involve the latent attribute L , we compute their product and normalize to obtain the posterior, instead of computing the full joint distribution. This yields a soft assignment of L for each record, effectively completing the data over $\mathcal{V} \cup L$ for the subsequent M-step.

5.4.2 Analysis of M-step. In this step, we update parameters using the posterior $\mathbb{P}[L \mid \mathcal{V}; \hat{\theta}^{(k)}]$ from the E-step. Let $\omega_j(l) := \mathbb{P}[L = l \mid \mathcal{V}^j; \hat{\theta}^{(k)}]$ denote the posterior weight of latent state l for record j . The objective is to maximize:

$$\theta^{(k)} = \arg \max_{\theta} \sum_j \sum_l \left(\omega_j(l) \cdot \log \mathbb{P}[\mathcal{V}^j, L = l; \theta] \right). \quad (11)$$

Using the DAG factorization in Equation (3),

$$\mathbb{P}[\mathcal{V}, L; \theta] \propto \mathbb{P}[L; \theta] \cdot \mathbb{P}[\mathcal{I}_{c_1} \mid \Pi'_{c_1}; \theta] \cdot \mathbb{P}[\mathcal{I}_{c_2} \mid \Pi'_{c_2}; \theta] \cdot \mathbb{P}[Y \mid \Pi'_Y; \theta],$$

the objective decomposes additively across local conditionals. In the discrete setting, each local conditional is multinomial, so parameters admit EM closed-form updates as normalized expected counts. Let $N_E(\cdot)$ denote expectations under the posterior. The updates are:

$$\begin{aligned} \theta_l &= \frac{N_E(l)}{\sum_l N_E(l)}, & \theta_{i_{c_1} | \pi_{c_1}} &= \frac{N_E(i_{c_1}, \pi_{c_1})}{\sum_{i_{c_1}} N_E(i_{c_1}, \pi_{c_1})}, \\ \theta_{y | \pi_y} &= \frac{N_E(y, \pi_y)}{\sum_y N_E(y, \pi_y)}, & \theta_{i_{c_2} | \pi_{c_2}} &= \frac{N_E(i_{c_2}, \pi_{c_2})}{\sum_{i_{c_2}} N_E(i_{c_2}, \pi_{c_2})}, \end{aligned} \quad (12)$$

where

$$N_E(y, \pi_y) = \sum_j \sum_l \left(\mathbb{P}[L = l \mid \mathcal{V}^j; \hat{\theta}^{(k)}] \cdot \mathbb{I}[Y^j = y, \Pi_Y'^j = \pi_y] \right),$$

and analogously for other factors. The final $\theta^{(k)}$ is formed by concatenating all local parameters.

5.4.3 Overall. After each M-step, we calculate the refined log-likelihood as

$$\mathcal{L}^{(k)} \propto \sum_j \log \left(\sum_l \theta_l^{(k)} \cdot \theta_{i_{c_1}^j | \pi_{c_1}^j(l)}^{(k)} \cdot \theta_{i_{c_2}^j | \pi_{c_2}^j(l)}^{(k)} \cdot \theta_{y^j | \pi_y^j(l)}^{(k)} \right) \quad (13)$$

and terminate the procedure when $|\mathcal{L}^{(k)} - \mathcal{L}^{(k-1)}| \leq \eta$ or the iteration count reaches n . Once converged, we obtain the optimal parameters for the local structures around L , $\mathcal{I}_{c_1}$, $\mathcal{I}_{c_2}$, and Y . For the remaining attributes, denoted as $\mathcal{X} = \mathcal{V} \setminus (\mathcal{I}_c \cup \{Y\})$, the parameters are constant and can be computed directly via MLE:

$$\theta_x = \frac{N(x)}{\sum_{x'} N(x')}. \quad (14)$$

Algorithm 4: PARAMETERESTIMATION

Input: Database $\mathcal{D}$ with attribute set $\mathcal{V} = \mathcal{X} \cup \mathcal{I}_0 \cup \mathcal{I}_{c_1} \cup \mathcal{I}_{c_2} \cup \mathcal{A} \cup \mathcal{W} \cup \{Y\}$, maximum number of iterations n , convergence threshold η

Output: Parameters θ^*

- 1 Initialize $\theta_l, \theta_{i_{c_1}|\pi_{c_1}}, \theta_{i_{c_2}|\pi_{c_2}}, \theta_{y|\pi_y}$ randomly;
- 2 **for** $i \leftarrow 0$ **to** n **do**
- 3 $\mathbb{P}_{L|\mathcal{V}} \leftarrow$ Equation (10);
- 4 Update $\theta_l, \theta_{i_{c_1}|\pi_{c_1}}, \theta_{i_{c_2}|\pi_{c_2}}, \theta_{y|\pi_y}$ based on Equation (12);
- 5 Check log-likelihood according to Equation (13): if the change is less than η , stop the iteration;
- 6 $i \leftarrow i + 1$
- 7 Compute θ^* based on Equations (14) and (15);
- 8 **return** θ^* ;

The final parameter set is formed by concatenating the estimates over all local distributions:

$$\theta^* = (\theta_x, \theta_l, \theta_{i_{c_1}|\pi_{c_1}}, \theta_{i_{c_2}|\pi_{c_2}}, \theta_{y|\pi_y}). \quad (15)$$

Note that high-dimensional distributions can be approximated using techniques in [8]; since this is not our focus, we treat all required distributions as given. The full parameter estimation procedure is summarized in Algorithm 4.

6 Experiments

In this section, we evaluate the effectiveness of LatentPre through extensive experiments. We first describe the experimental setup in Section 6.1. We then compare LatentPre with baselines by reporting end-to-end performance in Section 6.2 and visualizing missing-relationship recovery in Section 6.3. Finally, we provide a deeper analysis of LatentPre by testing the dependence between the learned latent attribute and sensitive attributes in Section 6.4, examining how key parameters affect performance in Section 6.5, and profiling runtime in Section 6.6.

6.1 Setup

We implement the proposed LatentPre framework in Python¹. All experiments are conducted on a machine with two Xeon(R) Gold 6326@2.90 GHz CPUs and 256GB of DRAM.

Datasets. We evaluate the performance of LatentPre on three real-world datasets, including Adult [4], COMPAS [19], and Census-KDD [1], and two synthetic datasets generated following [24]. The key statistics of these datasets are summarized in Table 1.

Table 1. Dataset statistics.

Dataset	#Records	#Attributes	Avg. Dom
Adult	32,561	13	13.33
COMPAS	6,130	8	4
Census-KDD	196,130	28	11.67
Synthetic (Section 6.3)	50,000	7	4
Synthetic (Section 6.6)	50,000–500,000	15–60	4

¹Available at <https://github.com/iamzhengying/LatentPre.git>

Baselines. We compare LatentPre with state-of-the-art pre-processing methods, Cap-MS [37], Cap-MF [37], OTClean [30], and CausalPre [55]. For all of them, we use the source code provided by the authors. Since Cap-MS and Cap-MF are designed only for saturated CI constraints over all attributes, we extend them by first processing $\mathcal{D}_{\mathcal{V} \setminus \mathcal{W}}$ and then sampling $\mathcal{W}$ from its conditional distribution to complete the entire $\mathcal{D}'$. For OTClean [30], we report two variants to ensure a fair comparison, following the evaluation protocol in [55]: (i) OTClean-RT, which directly uses the provided codebase and thus modifies the testing data, and (ii) OTClean, which processes only the training data and leaves the testing set unchanged. We also report results on the original dataset, denoted as “Original”, and the dataset with all sensitive and inadmissible attributes dropped, denoted as “Dropped”. The former provides a baseline for understanding the dataset’s utility and inherent discrimination without any intervention, while the latter reflects what a naive brute-force approach can achieve.

Measurement and metrics. We evaluate each framework by measuring the quality of its processed data along two dimensions: utility and fairness. To achieve this, each raw dataset is divided into a training set and a testing set; only the training set is pre-processed, while the testing set remains unchanged to simulate a realistic environment. Two standard classifiers, random forest (RF) and multilayer perceptron (MLP), are then trained on the processed training data and evaluated on the testing data. The resulting predictive performance then serves as a proxy for data quality, as higher-quality training data enables models to produce more accurate and less biased predictions. For utility, we report the AUC score, where a higher value indicates better generalization and thus higher data quality. For fairness, we use the Ratio of Observational Discrimination (ROD) [37], which measures deviation from fairness; formally

$$\text{ROD} = \max_{S_0, S_1 \in \text{Dom}(S)} \frac{1}{|\text{Dom}(\mathcal{A})|} \sum_{a \in \text{Dom}(\mathcal{A})} \overline{\text{ROD}}(s_0, s_1; \hat{Y} \mid a),$$

where

$$\overline{\text{ROD}}(s_0, s_1; \hat{Y} \mid a) = \frac{\mathbb{P}[\hat{Y} = 1 \mid s_0, a] \cdot \mathbb{P}[\hat{Y} = 0 \mid s_1, a]}{\mathbb{P}[\hat{Y} = 0 \mid s_0, a] \cdot \mathbb{P}[\hat{Y} = 1 \mid s_1, a]}.$$

For consistent interpretation, we report the normalized absolute logarithm of the ROD value; a value of 0 indicates no discrimination, and larger values indicate greater discrimination. Finally, to ensure robustness and reduce variability, all evaluations are conducted using five-fold cross-validation.

6.2 End-to-End Performance Evaluation

We evaluate the end-to-end performance of LatentPre under three scenarios: (i) attribute ambiguity, (ii) attribute absence, and (iii) perfect attribute space. Results are shown in Figures 4, 5, and 6, respectively. Performance is measured using box plots of 5-fold AUC and average ROD. Cap-MS is excluded from Census-KDD due to excessive runtime (over 24 hours). OTClean is omitted from Adult and Census-KDD because it runs out of memory on our machine. Default parameters for LatentPre are set as follows: the number of latent states $\tau = 6, 3$, and 6 for Adult, COMPAS, and Census-KDD; the number of EM iterations $n = 800$; and convergence threshold $\eta = 0.001$. A method is deemed invalid if its fairness is worse than the unprocessed “Original” baseline or its utility lower than the “Dropped” case; these regions are shaded in gray. Key findings for each scenario are summarized below.

6.2.1 Attribute Ambiguity. This setting evaluates robustness when certain utility-relevant but ambiguous attributes are conservatively treated as inadmissible. To simulate this scenario, we randomly reassign a subset of utility-relevant attributes from the admissible or additional set to

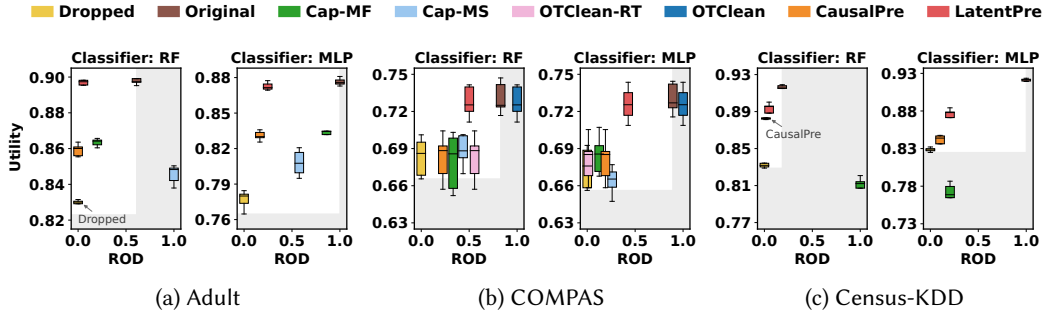


Fig. 4. End-to-end performance under attribute ambiguity across three datasets, measured by AUC (utility) and ROD (fairness). Each box summarizes the 5-fold AUC values and the corresponding average ROD for one approach; higher AUC and lower ROD indicate better performance. Shaded regions indicate invalid results.

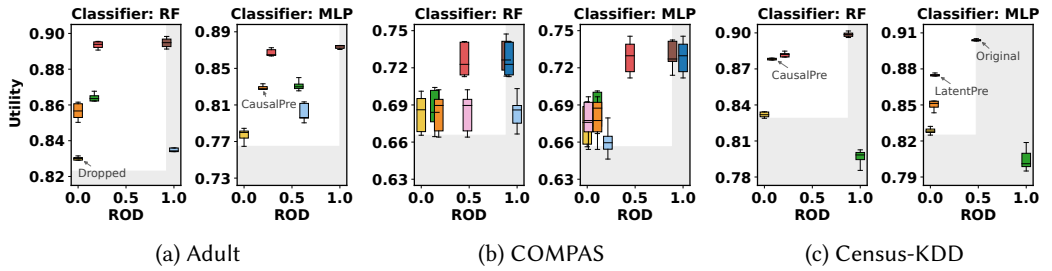


Fig. 5. End-to-end performance under attribute absence across three datasets.

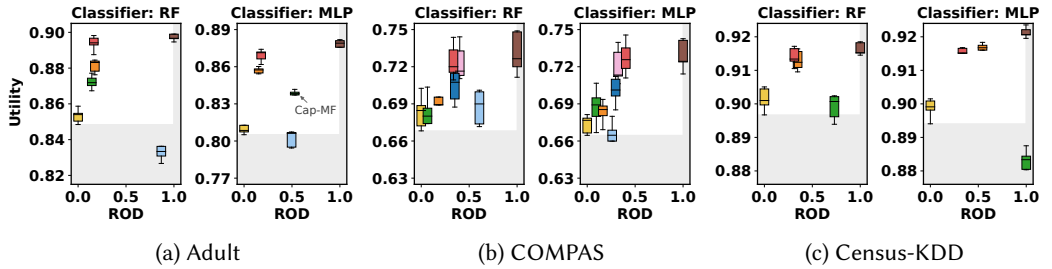


Fig. 6. End-to-end performance under perfect attribute space across three datasets.

the inadmissible set. Specifically, we reassign one attribute for the Adult and COMPAS datasets, and six for the larger Census-KDD dataset. We use a larger number for Census-KDD because it contains many structurally similar or semantically related attributes, which practitioners may be more inclined to conservatively label as inadmissible together. Figure 4 reports the results. Across all datasets, LatentPre significantly outperforms the baselines in utility and consistently remains valid. This holds even on COMPAS, a low-dimensional dataset where a limited attribute space makes the causal structure more fragile and fairness-aware processing more challenging. Even in this setting, LatentPre remains robust, demonstrating the effectiveness of latent augmentation in recovering and reinforcing the underlying structure from sparse observations. In addition, LatentPre reduces average discrimination by 80%, 46%, and 77%, with only 0.2%, 0.5%, and 4.2% utility loss on datasets Adult, COMPAS, and Census-KDD, respectively. The fairness gain on COMPAS is

relatively smaller, and we defer the explanation to Section 6.2.4. Nevertheless, this result is still acceptable, as the utility loss is minimal and the fairness gain comes at negligible cost. On Census-KDD, performance declines slightly because we treat more attributes as imperfect to reflect a more challenging and realistic setting, yet LatentPre still achieves a 2.2% average utility improvement over the best-performing baseline.

6.2.2 Attribute Absence. This setting evaluates performance when some meaningful attributes are entirely missing from the dataset. To simulate this scenario, we randomly remove utility-relevant attributes from the admissible or additional set: one for Adult and COMPAS, and six for Census-KDD. The results in Figure 5 show that LatentPre substantially outperforms all baselines in utility while also reducing discrimination. Specifically, LatentPre lowers average discrimination by 75%, 50%, and 83%, with only 0.4%, 0.3%, and 2.6% utility loss on datasets Adult, COMPAS, and Census-KDD, respectively. The overall trend is consistent with the results observed under attribute ambiguity.

6.2.3 Perfect Attribute Space. This setting evaluates performance when all attributes are correctly collected and specified, as in prior work [37, 55]. The results in Figure 6 show that LatentPre still achieves utility gains in most cases, even in this idealized setting. This indicates that attribute ambiguity or absence may naturally arise in practice, and LatentPre can effectively identify and correct such hidden imperfections through latent modeling. Note that although LatentPre sometimes attains the best ROD, introducing the latent attribute does not itself contribute to fairness. Fairness is ensured by explicitly removing all unfair causal pathways from sensitive or inadmissible attributes to the label. Meanwhile, introducing the latent attribute requires pruning certain dependencies among inadmissible attributes to ensure identifiability. This identifiability-driven pruning can also reduce spurious high-dimensional correlations that powerful classifiers might otherwise inadvertently exploit as unfair influence. This side effect is typically more pronounced in higher-dimensional datasets such as Census-KDD.

6.2.4 Additional Discussion. As noted, LatentPre yields smaller fairness improvements on COMPAS across all three scenarios. This outcome is expected given two inherent properties of the dataset. First, COMPAS exhibits strong correlations between utility-related and sensitive information, making the utility–fairness trade-off particularly pronounced. Although the introduced latent attribute is not causally influenced by the sensitive attribute (and we further evaluate their independence in Section 6.4), powerful classifiers may still exploit spurious high-dimensional correlations in the data. This phenomenon is not unique to LatentPre: OTClean shows a similarly sharp utility–fairness trade-off on COMPAS; moreover, since OTClean is not designed for imperfect scenarios, its behavior in such settings is even less controllable. Second, COMPAS is relatively small, which makes the empirical distribution less stable and further increases the risk that powerful classifiers capture spurious correlations. To validate this explanation, we conduct an auxiliary experiment that enlarges COMPAS via repeated sampling (by increasing k in Algorithm 1, Line 7). With the MLP classifier under this setting, LatentPre achieves a lower (thus better) ROD of 0.30, which is comparable to the gains observed on other datasets. In summary, the limited gains on COMPAS primarily reflect the dataset’s challenging characteristics rather than a weakness of LatentPre. Importantly, LatentPre still delivers meaningful fairness improvements with almost no sacrifice in utility, making the outcome practically satisfactory.

6.3 Missing Relationship Recovery

To evaluate how well LatentPre and the baselines preserve potential causal relationships when some attributes are missing, we generate a synthetic dataset from the DAG in Figure 7a. The dataset contains 50,000 records and 7 attributes. Note that the attribute space is deliberately kept small to

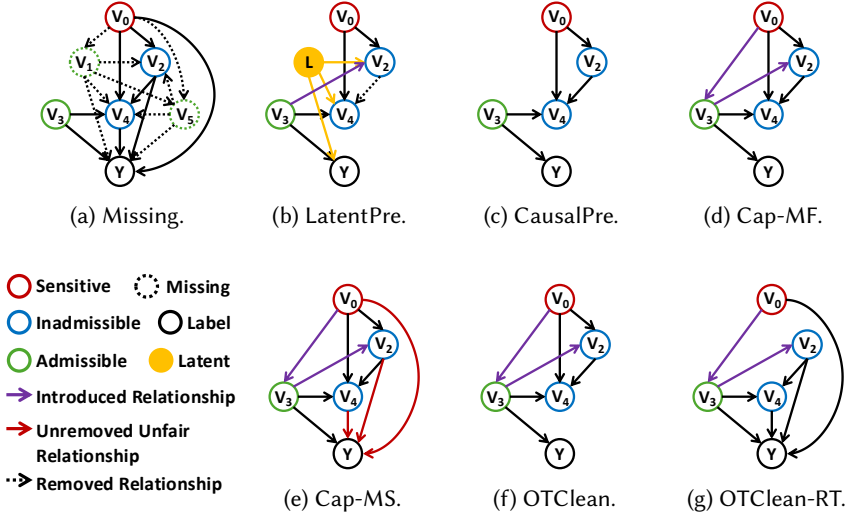


Fig. 7. Relationship recovery with $\mathcal{S}=\{V_0\}$, $\mathcal{I}=\{V_2, V_4\}$, $\mathcal{A}=\{V_1, V_3, V_5\}$, and missing attributes $\{V_1, V_5\}$.

make DAG visualization feasible. On this dataset, we first randomly assign attributes to reasonable roles, such as inadmissible or admissible, and then randomly remove several attributes from the admissible attribute set to simulate scenarios where some meaningful and usable information is lost. We next apply LatentPre and the baselines to the resulting dataset and reconstruct a DAG from each processed output using the Python libraries *pgmpy.estimators* and *networkx*. These recovered DAGs provide a direct view of which causal relationships are preserved after processing.

Figure 7 shows one representative setting, where $\{V_0\}$ is sensitive, $\{V_2, V_4\}$ are inadmissible, $\{V_1, V_3, V_5\}$ are admissible, and Y is the label, with V_1 and V_5 removed to simulate missing attributes. As observed, CausalPre in Figure 7c removes all unfair pathways and preserves observed V_3 as a decision factor. OTClean in Figure 7f and Cap-MF in Figure 7d yield a similar structure but introduce two additional edges, which are highlighted in purple. In contrast, Cap-MS fails to block all unfair causal pathways, as shown in Figure 7e, because it substantially distorts the data distribution during processing. This also explains why its results are always invalid in Section 6.2. Figure 7g reports OTClean-RT. Since it targets a different processing constraint, we evaluate it under a corresponding criterion that focuses on severing dependence between sensitive and inadmissible attributes. Under this criterion, OTClean-RT successfully breaks this dependence, but it still introduces the same two extra edges as OTClean and Cap-MF. Overall, none of the baselines recover the potential utility-relevant information associated with the missing attributes.

LatentPre exhibits a different behavior by explicitly recovering the lost information. As shown in Figure 7b, it introduces a latent attribute L that contributes to prediction alongside V_3 . This latent attribute captures the influence of the missing attributes V_1 and V_5 by reinstating their direct relationships with other attributes such as V_2 and V_4 , and by passing the recovered information to the label. Consequently, LatentPre restores the missing causal pathways, as if V_1 and V_5 were still present. Achieving this recovery requires a deliberate trade-off. To preserve more utility-relevant information near the label, LatentPre modifies certain relationships around the inadmissible attributes: it removes the edge between V_2 and V_4 for identifiability and introduces an additional edge $V_3 \rightarrow V_2$ to support a coarser but tractable policy. Strictly speaking, V_3 should be treated as an additional attribute rather than an admissible one, in which case the edge $V_3 \rightarrow V_2$ would

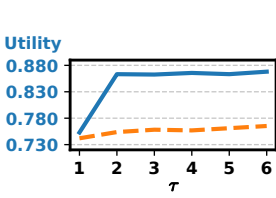
Table 2. Independence analysis of L and S .

Dataset	Scenario	Sensitive Attribute	NMI	chi-square
Adult	Ambiguous	sex	$(3.1 \pm 0.8) \times 10^{-5}$	0.71 ± 0.12
		race	$(1.7 \pm 0.2) \times 10^{-4}$	0.74 ± 0.14
		sex + race	$(3.6 \pm 0.4) \times 10^{-4}$	0.58 ± 0.21
	Absent	sex	$(2.9 \pm 1.5) \times 10^{-5}$	0.74 ± 0.20
		race	$(2.0 \pm 0.1) \times 10^{-4}$	0.58 ± 0.09
		sex + race	$(3.8 \pm 0.3) \times 10^{-4}$	0.51 ± 0.16
	Perfect	sex	$(2.7 \pm 0.6) \times 10^{-5}$	0.76 ± 0.09
		race	$(2.0 \pm 0.2) \times 10^{-4}$	0.56 ± 0.12
		sex + race	$(3.8 \pm 0.4) \times 10^{-4}$	0.53 ± 0.22
COMPAS	Ambiguous	sex	$(5.9 \pm 3.0) \times 10^{-5}$	0.74 ± 0.13
		race	$(2.3 \pm 0.5) \times 10^{-4}$	0.75 ± 0.10
		sex + race	$(4.4 \pm 1.1) \times 10^{-4}$	0.86 ± 0.09
	Absent	sex	$(1.1 \pm 1.3) \times 10^{-4}$	0.66 ± 0.29
		race	$(1.8 \pm 0.3) \times 10^{-4}$	0.85 ± 0.06
		sex + race	$(4.8 \pm 1.1) \times 10^{-4}$	0.82 ± 0.11
	Perfect	sex	$(3.5 \pm 2.4) \times 10^{-5}$	0.83 ± 0.10
		race	$(2.2 \pm 0.5) \times 10^{-4}$	0.76 ± 0.09
		sex + race	$(4.4 \pm 1.6) \times 10^{-4}$	0.83 ± 0.15
Census-KDD	Ambiguous	sex	$(2.0 \pm 1.0) \times 10^{-5}$	0.28 ± 0.30
		race	$(4.1 \pm 1.3) \times 10^{-5}$	0.73 ± 0.22
		sex + race	$(8.6 \pm 2.3) \times 10^{-5}$	0.57 ± 0.30
	Absent	sex	$(2.2 \pm 0.5) \times 10^{-5}$	0.12 ± 0.08
		race	$(4.3 \pm 0.5) \times 10^{-5}$	0.61 ± 0.13
		sex + race	$(8.5 \pm 0.9) \times 10^{-5}$	0.43 ± 0.17
	Perfect	sex	$(1.2 \pm 0.1) \times 10^{-5}$	0.31 ± 0.06
		race	$(2.7 \pm 0.3) \times 10^{-5}$	0.86 ± 0.07
		sex + race	$(6.2 \pm 0.3) \times 10^{-5}$	0.70 ± 0.08

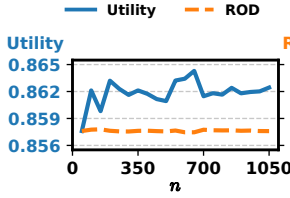
not be introduced. We treat V_3 as admissible here only to allow the baselines to operate more effectively on this simple dataset. This trade-off proves beneficial: as evidenced by our further end-to-end evaluations, LatentPre consistently outperforms the baselines, achieving an average utility improvement of 22% across both classifiers.

6.4 Verifying Latent-Sensitive Independence

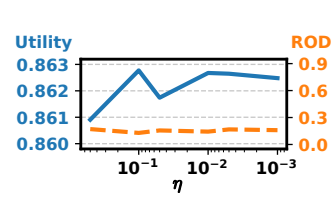
As mentioned in Section 4.1, the latent attribute L is designed to be independent of the sensitive attributes S . To empirically validate this property, we test whether the learned L exhibits any residual dependence on S from both information-theoretic and statistical perspectives, using Normalized Mutual Information (NMI) and the chi-square test. NMI measures the strength of dependence, where a value of 0 indicates independence and values below 10^{-3} are commonly treated as negligible in practice. The chi-square test assesses statistical evidence of dependence: the resulting p -values above 0.01 indicate no statistically significant dependence. Table 2 presents the 5-fold results on three real-world datasets under two imperfect scenarios as well as the perfect scenario. Since each dataset includes two sensitive attributes, sex and race, we test dependence



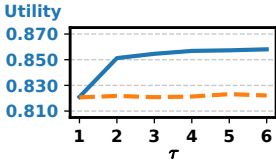
(a) Attribute ambiguity.



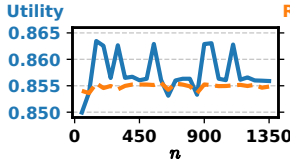
(a) Attribute ambiguity.



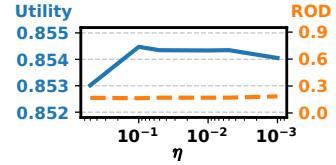
(a) Attribute ambiguity.



(b) Attribute absence.



(b) Attribute absence.



(b) Attribute absence.

Fig. 8. Parameter analysis on the number of latent states τ .Fig. 9. Parameter analysis on the number of iterations n .Fig. 10. Parameter analysis on the convergence threshold η .

with respect to each attribute separately as well as their intersectional groups. Across all settings, NMI values are near zero and chi-square p -values consistently exceed the threshold. Both results indicate no detectable dependence between L and S , which is consistent with our design.

6.5 Parameter Analysis

We study the sensitivity of LatentPre to its key hyperparameters. For brevity, we report results on the representative Adult dataset with the MLP classifier; similar trends hold across other datasets and classifiers. As described in Algorithm 1, LatentPre has three parameters: the number of latent states τ , the maximum number of iterations n , and the convergence threshold η . Unless otherwise stated, when varying one hyperparameter, we fix the other two to $\tau = 6$, $n = 800$, and $\eta = 0.001$.

Varying τ . Figure 8 shows the impact of varying the number of latent states τ . When $\tau = 1$, all records share a single latent value, which is equivalent to having no latent attribute. As τ increases and captures more variation, utility improves consistently, with gains of 14.66% and 3.68% under the ambiguous and absent attribute settings, respectively. The gains are more pronounced under attribute ambiguity, since the ambiguous attributes remain in the dataset, making their influence easier to recover in the early stages. As τ continues to grow, utility improves marginally while ROD remains stable. Due to identifiability constraints, τ must lie within a bounded range; for Adult, we set the maximum to 6.

Varying n and η . Figure 9 illustrates performance changes with the number of iterations n . Across both scenarios, utility increases in the early iterations and then stabilizes, while fairness remains steady throughout. A similar pattern is observed on the convergence threshold η , as shown in Figure 10, within the expected noise levels.

6.6 Runtime Evaluation

6.6.1 Staged. To evaluate the efficiency of the identification and partition algorithms proposed in Sections 5.2 and 5.3, we study how runtime scales with key dataset properties using synthetic data. For identification, we vary one factor at a time among the inadmissible-set size $|I|$, the dataset size $|\mathcal{D}|$, and the number of attributes $|\mathcal{V}|$, as shown in Figures 11a–11c, while fixing the other two at $|I|=10$, $|\mathcal{D}|=50,000$, and $|\mathcal{V}|=30$. As shown, increasing $|I|$ or $|\mathcal{V}|$ leads only to a mild runtime

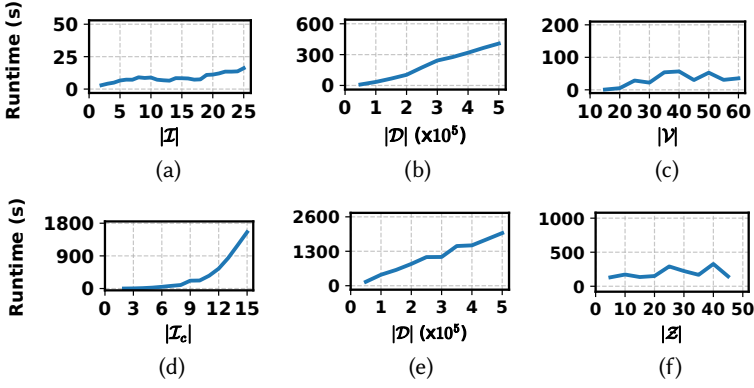


Fig. 11. Staged runtime analysis for $\mathcal{I}_c$ identification algorithm ((a)–(c)) and $\mathcal{I}_c$ partition algorithm ((d)–(f)).

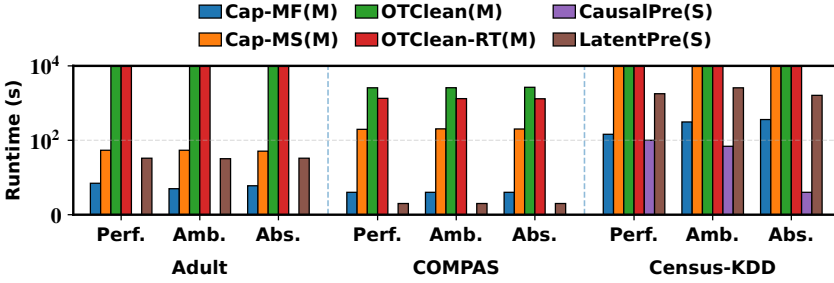


Fig. 12. Runtime comparison. (M: execution with 64 threads; S: execution with a single thread.)

increase with small fluctuations. This indicates that CI tests executed in Lines 8–10 of Algorithm 2 are far fewer than the worst-case count. In practice, the procedure typically terminates early rather than enumerating all conditioning sets. When $|\mathcal{D}|$ increases, the runtime grows linearly, as each CI test becomes more expensive with more records. For partition, we similarly vary one factor at a time among $|\mathcal{I}_c|$, $|\mathcal{D}|$, and the conditioning-set size $|\mathcal{Z}| = |\mathcal{S} \cup \mathcal{I}_o \cup \mathcal{A}|$, as shown in Figures 11d–11f, while fixing the remaining two at $|\mathcal{I}_c| = 10$, $|\mathcal{D}| = 50,000$, and $|\mathcal{Z}| = 20$. The partition runtime scales quadratically with $|\mathcal{I}_c|$, grows linearly with $|\mathcal{D}|$, and shows only mild sensitivity to $|\mathcal{Z}|$. Overall, the observed trends are well below the theoretical worst-case bound discussed in Section 5.3 and support practical scalability.

6.6.2 Overall. Figure 12 reports the runtime of LatentPre and the baselines. LatentPre and CausalPre use a single thread, while the others use 64 threads. Bars reaching the top indicate failure to complete. CausalPre is sometimes not visible because its runtime falls below the plotted scale, under 2 seconds. Overall, among the six approaches, LatentPre typically ranks third, after CausalPre and Cap-MF. In the most challenging setting, Census-KDD under the ambiguous scenario, LatentPre completes in 2,573 seconds, which remains acceptable for an offline pre-processing task.

7 Related Work

Our work focuses on justifiable fairness and is, to our knowledge, the first to address fair data pre-processing under an imperfect attribute space. All existing works assume that the available

attributes are complete and correctly specified (as admissible, inadmissible, etc.), and are therefore not designed to tackle this new challenge.

Salimi et al. [37] are the first to formalize the fairness-aware database repair problem and define the notion of justifiable fairness. Their framework, Capuchin, aims to achieve fairness while minimizing the number of record modifications. Building upon Capuchin, OTClean [30] further controls distributional distortion via optimal transport, seeking to minimize distribution shift in addition to enforcing fairness. More recently, CausalPre [55] adopts a two-step strategy: it efficiently learns causally fair relationships from scratch and applies them to refine the dataset, ensuring fairness while preserving the overall statistical character.

Several other pre-processing techniques [12, 30, 33, 34, 36] also adopt causal fairness, but target different problems. PreFair [33] also builds on justifiable fairness, but does not consider causality; instead, it focuses on incorporating fairness constraints into private data synthesis, which differs from our repair-oriented framework. FairPFN [34] studies fairness-aware pre-training with a fair data generation module. It aims to eliminate the total causal effect of a single binary sensitive attribute, and is therefore not directly comparable to our setting. FairExp [36] and SeqSel [12], on the other hand, focus on fair feature selection rather than data adjustment. Beyond causal fairness, several pre-processing methods [7, 11, 13, 15, 25, 45, 47, 48, 50] target associational fairness, especially the notion of demographic parity. Since they enforce only associational constraints, they generally cannot guarantee causal fairness [37].

Finally, some works [17, 22, 35, 42] also attempt to explore unobserved signals from the data. EXCO [42], for example, introduces latent attributes to represent exogenous background factors that influence the sensitive attribute and uses them to develop a fair predictor. CLAIRE [22] does not explicitly model latent attributes but instead learns fair representations with a variational autoencoder. These methods operate during model training, whereas our work addresses fairness through the pre-processing stage.

8 Conclusion

In this paper, we re-examined the problem of fair data pre-processing under an imperfect attribute space. Our framework, LatentPre, augments the fairness policy with latent attributes that guarantee identifiability, recover valid missing signals, and block inadmissible influence. Guided by this policy, the raw data is adjusted to satisfy justifiable fairness while preserving utility. Extensive experiments show that LatentPre is robust across a variety of real-world imperfect settings and achieves strong fairness-utility trade-offs.

References

- [1] 2000. Census-Income (KDD). UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5N30T>. Accessed: 2025-06.
- [2] Elizabeth S Allman and Catherine Matias. 2009. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics* 37, 6A (2009), 3099–3132.
- [3] Elizabeth S Allman, John A Rhodes, Elena Stanghellini, and Marco Valtorta. 2015. Parameter identifiability of discrete Bayesian networks with hidden variables. *Journal of Causal Inference* 3, 2 (2015), 189–205.
- [4] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5XW20>. Accessed: 2025-06.
- [5] Peter J Bickel, Eugene A Hammel, and J William O’Connell. 1975. Sex Bias in Graduate Admissions: Data from Berkeley: Measuring bias is harder than is usually assumed, and the evidence is sometimes contrary to expectation. *Science* 187, 4175 (1975), 398–404.
- [6] Tim Brennan, William Dieterich, and Beate Ehret. 2009. Evaluating the predictive validity of the COMPAS risk and needs assessment system. *Criminal Justice and Behavior* 36, 1 (2009), 21–40.
- [7] Flávio P. Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. 2017. Optimized Pre-Processing for Discrimination Prevention. In *NeurIPS*. 3992–4001.

- [8] C. K. Chow and C. N. Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory* 14, 3 (1968), 462–467.
- [9] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of information theory* (2. ed.). Wiley.
- [10] AP Dempster, NM Laird, and DB Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* (1977), 1–38.
- [11] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *KDD*. 259–268.
- [12] Sainyam Galhotra, Karthikeyan Shanmugam, Prasanna Sattigeri, and Kush R. Varshney. 2022. Causal Feature Selection for Algorithmic Fairness. In *SIGMOD*. 276–285.
- [13] Paula Gordaliza, Eustasio del Barrio, Fabrice Gamboa, and Jean-Michel Loubes. 2019. Obtaining Fairness using Optimal Transport Theory. In *ICML*, Vol. 97. 2357–2365.
- [14] Mohammad Hashemi, Shengbo Gong, Juntong Ni, Wenqi Fan, B Aditya Prakash, and Wei Jin. 2024. A comprehensive survey on graph reduction: sparsification, coarsening, and condensation. In *IJCAI*. 8058–8066.
- [15] Faisal Kamiran and Toon Calders. 2011. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* 33, 1 (2011), 1–33.
- [16] Murat Kocaoglu. 2023. Characterization and learning of causal graphs with small conditioning sets. *NeurIPS*, 74140–74179.
- [17] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *NeurIPS*. 4066–4076.
- [18] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. 2019. iFair: Learning Individually Fair Data Representations for Algorithmic Decision Making. In *ICDE*. 1334–1345.
- [19] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. COMPAS. ProPublica. Available at <https://github.com/propublica/compas-analysis/>. Accessed: 2025-06.
- [20] Yin Lin, Samika Gupta, and H. V. Jagadish. 2024. Mitigating Subgroup Unfairness in Machine Learning Classifiers: A Data-Driven Approach. In *ICDE*. 2151–2163.
- [21] Andreas Loukas. 2019. Graph Reduction with Spectral and Cut Guarantees. *J. Mach. Learn. Res.* 20 (2019), 1–42.
- [22] Jing Ma, Ruocheng Guo, Aidong Zhang, and Jundong Li. 2023. Learning for counterfactual fairness from observational data. In *SIGKDD*. 1620–1630.
- [23] John F Mahoney and James M Mohen. 2007. Method and system for loan origination and underwriting. US Patent 7,287,008.
- [24] Markos Markakis, Ziyu Zhang, Rana Shahout, Trinity Gao, Chunwei Liu, Ibrahim Sabek, and Michael J. Cafarella. 2024. Press ECCS to Doubt (Your Causal Graph). In *GUIDE-AI@SIGMOD*. 6–15.
- [25] Razieh Nabi and Ilya Shpitser. 2018. Fair Inference on Outcomes. In *AAAI*. 1931–1940.
- [26] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366, 6464 (2019), 447–453.
- [27] Tony Pals and Marla Koenigsknecht. 2024. Study: Algorithms Used by Universities to Predict Student Success May Be Racially Biased. *American Educational Research Association (AERA)* (2024).
- [28] J Pearl. 2009. *Causality*. Cambridge University Press.
- [29] Dana Pessach and Erez Shmueli. 2022. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–44.
- [30] Alireza Pirhadi, Mohammad Hossein Moslemi, Alexander Cloninger, Mostafa Milani, and Babak Salimi. 2024. OTClean: Data Cleaning for Conditional Independence Violations using Optimal Transport. In *SIGMOD*, Vol. 2. 160.
- [31] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2022. Fairness in rankings and recommendations: an overview. *VLDB J.* 31, 3 (2022), 431–458.
- [32] Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. 2022. Interpretable Data-Based Explanations for Fairness Debugging. In *SIGMOD*. 247–261.
- [33] David Pujol, Amir Gilad, and Ashwin Machanavajjhala. 2023. PreFair: Privately Generating Justifiably Fair Synthetic Data. In *VLDB*, Vol. 16. 1573–1586.
- [34] Jake Robertson, Noah Hollmann, Samuel Müller, Noor H. Awad, and Frank Hutter. 2025. FairPFN: A Tabular Foundation Model for Causal Fairness. In *ICML*.
- [35] Chris Russell, Matt J Kusner, Joshua Loftus, and Ricardo Silva. 2017. When worlds collide: integrating different counterfactual assumptions in fairness. *NeurIPS*.
- [36] Ricardo Salazar, Felix Neutatz, and Ziawasch Abedjan. 2021. Automated Feature Engineering for Algorithmic Fairness. In *VLDB*, Vol. 14. 1694–1702.
- [37] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional Fairness: Causal Database Repair for Algorithmic Fairness. In *SIGMOD*. 793–810.
- [38] Andrew D Selbst. 2017. Disparate impact in big data policing. *Ga. L. Rev.* 52 (2017), 109.

- [39] Arjun Sondhi and Ali Shojaie. 2019. The reduced PC-algorithm: Improved causal structure learning in large random networks. *J. Mach. Learn. Res.* 20, 164 (2019), 1–31.
- [40] Tanmay Surve and Romila Pradhan. 2025. Explaining Fairness Violations using Machine Unlearning. In *EDBT*. 623–635.
- [41] Ki Hyun Tae, Hantian Zhang, Jaeyoung Park, Kexin Rong, and Steven Euijong Whang. 2024. Falcon: Fair Active Learning Using Multi-Armed Bandits. In *VLDB*, Vol. 17. 952–965.
- [42] Bowei Tian, Ziyao Wang, Shwai He, Wanghao Ye, Guoheng Sun, Yucong Dai, Yongkai Wu, and Ang Li. 2025. Towards counterfactual fairness through auxiliary variables. In *ICLR*.
- [43] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* 65, 1 (2006), 31–78.
- [44] Sotiris Tsioutsoulis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. 2021. Fairness-aware pagerank. In *WWW*. 3815–3826.
- [45] Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. 2021. DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks. In *NeurIPS*. 22221–22233.
- [46] CF Jeff Wu. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics* (1983), 95–103.
- [47] Zikai Xiong, Niccolò Dalmaso, Alan Mishler, Vamsi K. Potluru, Tucker Balch, and Manuela Veloso. 2024. FairWASP: Fast and Optimal Fair Wasserstein Pre-processing. In *AAAI*. 16120–16128.
- [48] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2018. FairGAN: Fairness-aware Generative Adversarial Networks. In *IEEE BigData*. 570–575.
- [49] Ke Yang and Alexandra Meliou. 2024. Non-Invasive Fairness in Learning Through the Lens of Data Drift. In *ICDE*. 2164–2178.
- [50] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *ICML*. 325–333.
- [51] Anna Zeng, Michael Cafarella, Batya Kenig, Markos Markakis, Brit Youngmann, and Babak Salimi. 2025. Causal DAG Summarization. In *VLDB*, Vol. 18. 1933–1947.
- [52] Hantian Zhang, Ki Hyun Tae, Jaeyoung Park, Xu Chu, and Steven Euijong Whang. 2023. iFlipper: Label Flipping for Individual Fairness. In *SIGMOD*, Vol. 1. 1–26.
- [53] Qi Zhang, Rong-Hua Li, Zifan Zheng, Hongchao Qin, Ye Yuan, and Guoren Wang. 2025. Efficient Maximum Fair Clique Search over Large Networks. In *ICDE*. 1043–1055.
- [54] Lecheng Zheng, Dawei Zhou, Hanghang Tong, Jiejun Xu, Yada Zhu, and Jingrui He. 2024. Fairgen: Towards Fair Graph Generation. In *ICDE*. 2285–2297.
- [55] Ying Zheng, Yangfan Jiang, and Kian-Lee Tan. 2025. CausalPre: Scalable and Effective Data Pre-processing for Causal Fairness. *arXiv preprint arXiv:2509.15199* (2025).
- [56] Aoqi Zuo, Yiqing Li, Susan Wei, and Mingming Gong. 2024. Interventional Fairness on Partially Known Causal Graphs: A Constrained Optimization Approach. In *ICLR*.

Received October 2025; revised January 2026; accepted February 2026