CARMA: A practical framework to generate recommendations for causal algorithmic recourse at scale

Ayan Majumdar
ayanm@mpi-sws.org
MPI-SWS, Saarland University
Saarbrücken, Germany

Isabel Valera
ivalera@cs.uni-saarland.de
Saarland University, MPI-SWS
Saarbrücken, Germany

ABSTRACT

Algorithms are increasingly used to automate large-scale decision-making processes, e.g., online platforms that make instant decisions in lending, hiring, and education. When such automated systems yield unfavorable decisions, it is imperative to allow for recourse by accompanying the instantaneous negative decisions with recommendations that can help affected individuals to overturn them. However, the practical challenges of providing algorithmic recourse in large-scale settings are not negligible: giving recourse recommendations that are actionable requires not only causal knowledge of the relationships between applicant features but also solving a complex combinatorial optimization problem for each rejected applicant. In this work, we introduce CARMA, a novel framework to generate causal recourse recommendations at scale. For practical settings with limited causal information, CARMA leverages pre-trained state-of-the-art causal generative models to find recourse recommendations. More importantly, CARMA addresses the scalability of finding these recommendations by casting the complex recourse optimization problem as a prediction task. By training a novel neural-network-based framework, CARMA efficiently solves the prediction task without requiring supervision for optimal recourse actions. Our extensive evaluations show that post-training, running inference on CARMA reliably amortizes causal recourse, generating optimal and instantaneous recommendations. CARMA exhibits flexibility, as its optimization is versatile with respect to the algorithmic decision-making and pre-trained causal generative models, provided their differentiability is ensured. Furthermore, we showcase CARMA in a case study, illustrating its ability to tailor causal recourse recommendations by readily incorporating population-level feature preferences based on factors such as difficulty or time needed.

CCS CONCEPTS

• **Social and professional topics**; • **Computing methodologies** → **Machine learning**;

KEYWORDS

Recourse, Causality, Neural Networks, Counterfactual Explanations

FAccT '24, June 03–06, 2024, Rio de Janeiro, Brazil

ACM ISBN 979-8-4007-0450-5/24/06
https://doi.org/10.1145/3630106.3659003

ACM Reference Format:
Ayan Majumdar and Isabel Valera. 2024. CARMA: A practical framework to generate recommendations for causal algorithmic recourse at scale. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT '24), June 03–06, 2024, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3630106.3659003

1 INTRODUCTION

In societal settings ranging from lending [36] and hiring [40] to education [53], today's consequential decision-making tasks are frequently being automated using powerful algorithms like machine learning models. For instance, banks and financial institutions can provide *instant* lending decisions by powering automated online lending programs with machine learning models. Unfortunately, these models also make decision-making systems *complex* and *opaque* to the individuals subjected to algorithmic decisions. For example, individuals getting rejected from such opaque automated online loan application programs perceive a lack of explainability and transparency, eventually losing trust in these systems [50]. Hence, to improve the trustworthiness of automated decision-making systems, these systems must provide explanations and mechanisms to overturn unfavorable outcomes. Such a mechanism may be ethically desirable [47] or, in some cases, mandated by regulations [17, 52].

The literature on algorithmic recourse [16, 18, 19, 45] formalized the problem of overcoming unfavorable algorithmic decisions by providing individuals with relevant explanations and suggestions. To ensure recourse suggestions are optimal and actionable in practice, *causal algorithmic recourse* [18, 19, 23] emphasized the necessity of considering the causal relationships between the features. For instance, in the online loan application scenarios, the features might be causally related [19] as in Fig. 1a. Accounting for these causal relations enables the usage of the *counterfactual framework* formalized by Pearl [31]. This framework can use the *causal knowledge* of the feature relationships to estimate, for any recourse *intervention* of an individual, the hypothetical, *counterfactual* downstream effect it would entail in the real world. Hence, with knowledge of the actual causal relationships, causal recourse suggestions are *optimal interventions* that can be *acted upon* in reality. For online automated systems like the lending application, these suggestions should be generated quickly [17] to supplement the instantaneously provided negative decisions. However, for scenarios with potentially *many features* and *large populations* seeking recourse, causal recourse deployment remains challenging [19].

To estimate the downstream impact of interventions, the causal recourse optimization problem needs access to causal knowledge, i.e., the exact causal relations between features. While we can use expert insights to find the *causal graph* for each scenario showing

the *cause-effect links* between features [19], knowing the exact equational forms of the relations is generally impossible [32]. Although prior work [19] explored approximating the causal relations for recourse, they were *not accurate* enough to compute *counterfactual* estimates for *individual* recourse suggestions. Moreover, the prior approaches did not focus on the core optimization problem to ensure the provision of causal recourse *at scale*.

The issue with the causal recourse optimization problem is its inherent combinatorial complexity [19] since we need to find i) the feature combination to target and ii) the targeted features' value combination that ensures post-recourse favorable algorithmic decision. Unfortunately, the complexity is exacerbated by the *unamortized nature* of existing causal recourse optimization approaches [18, 19, 23]: the *combinatorial* problem needs to be solved *separately* for each *individual*. While this individualized approach may yield optimal recourse for each person, its inherent combinatorial complexity results in significant computation time in large-scale settings, with the time *increasing exponentially* with more features. Hence, in scenarios like the online lending programs, the significant computation time makes it challenging to provide instantaneous causal recourse recommendations [19], e.g., through conversational settings [11, 26, 42]. While some works on recourse [24, 49] have introduced amortized approaches, they cannot consider the extensive causal feature relationships required for providing *optimal, actionable* recourse interventions [18, 19].

For algorithmic decision-making classifiers deployed in large-scale settings, this work shows how to practically provide causal recourse by introducing CARMA, a novel **C**ausal **A**lgorithmic **R**ecourse framework utilizing neural network **M**odel-based **A**mortization. CARMA uses a key insight of causal recourse: all individuals share the underlying causal structure generating the features and the downstream decision-making classifier [18, 19]. This insight allows CARMA to transform the complex combinatorial recourse optimization problem into a simpler data-driven predictive task. CARMA solves the predictive task using an *unsupervised* approach, i.e., without relying on access to *ground-truth optimal recourse actions*. Furthermore, this problem is solved using a novel *neural network* (NN)-based framework. Once trained, CARMA's amortization readily uses *inference* on the NN framework to provide *optimal* and *instant* causal recommendations at scale.

Furthermore, to enable causal recourse in *practical* settings with *limited* causal knowledge, CARMA leverages *pre-trained* state-of-the-art causal generative models [15, 20, 38, 39, 55]. Using only observed data and the causal graph (cause-effect links between features), these models can *accurately estimate* the unknown causal relations and efficiently compute *individualized* counterfactuals. Hence, incorporating these *pre-trained* models helps CARMA to amortize recourse by *efficiently* approximating *individualized* recourse solutions without complete knowledge of the causal relations.

By successfully *amortizing* causal recourse, CARMA enables large-scale algorithmic systems providing instant decisions to supplement their negative predictions with instantaneous, optimal, and actionable recourse recommendations. CARMA's optimization remains flexible concerning the downstream algorithmic decision-making model and the pre-trained causal generative model, provided we ensure their *differentiability*. Moreover, CARMA can tailor the amortized recourse recommendations by incorporating *population-level feature preferences* regarding recourse based on factors like difficulty or the time required. To the best of our knowledge, CARMA is the first approach to *amortize* the causal recourse optimization problem. Our contributions are as follows.

- We propose CARMA, a novel and efficient recourse method that uses neural networks to amortize the causal recourse problem by mapping the complex optimization to a data-driven predictive problem.
- We show that CARMA is flexible in leveraging recent advances in deep causal generative models to amortize causal recourse even without complete causal knowledge about the features.
- We perform extensive empirical evaluations and show CARMA can solve for *optimal* causal recourse, *closely matching* the oracle's performance (using perfect causal knowledge) while requiring minimal computation time.
- We show in a case study how a population's preferences over different features can be incorporated in CARMA to provide more tailored and sparse causal recourse suggestions.

2 BACKGROUND

2.1 Causality

Solving causal algorithmic recourse relies on estimating the impact of the recommendations (causal interventions) on features and downstream algorithmic decisions. This is achieved by using the *structural causal model* framework [31].

A structural causal model (SCM) $\mathcal{M} = (\mathbf{X}, \mathbf{U}, \tilde{\mathbf{F}})$ is defined by observed (endogenous) features $\mathbf{X}$, unobserved exogenous variables $\mathbf{U}$ and a set of equations $\tilde{\mathbf{F}}$ that describe the causal relationship between features, their causal parents (pa), and the exogenous variables as $\tilde{f}_i = (\mathbf{x}_{\text{pa}_i}, \mathbf{u}_i)$. We can also represent the cause-effect relations through a causal graph $\mathcal{G}$ (e.g., Fig. 1a), where the nodes represent $\mathbf{X}$, and the edges represent the causal links between $\mathbf{X}$. Following prior work on causality [19, 25, 51], we consider that features $\mathbf{X}$ are related following a directed acyclic graph (DAG).

Importantly, SCMs allow reasoning about hypothetical causal modifications (interventions) for a given individual. Specifically, they allow answering counterfactual questions: "What would have been the features of $\mathbf{x}^{\text{F}}$, had $\mathbf{X}_i$ been set to a value $\imath_i$, all exogenous factors being equal?" Interventions can be represented with the do-operator $\text{do}(\mathbf{X}_i = \imath_i)$ [31] and lead to a modified SCM $\mathcal{M}'$. Given an action as a set of interventions $\boldsymbol{a} = \{\{\text{do}(\mathbf{X}_i = \imath_i)\}_{i \in \mathcal{I}}\}$, the *structural counterfactual* $\mathbf{x}^{\text{CF}} := \mathbf{x}^{\text{SCF}}(\boldsymbol{a}, \mathbf{x}^{\text{F}})$ represents the features that would have been observed in $\mathcal{M}'$, keeping exogenous variables $\mathbf{u}^{\text{F}}$ unchanged. For an individual $\mathbf{x}^{\text{F}}$, structural counterfactuals are computed with Pearl's three steps [31]: i) *abduction*: estimating exogenous variables $\mathbf{u}^{\text{F}}$ from $\mathcal{M}$, ii) *action*: performing intervention using $\text{do}(\mathbf{X}_i = \imath_i)$, iii) *prediction*: estimating the counterfactual features $\mathbf{x}^{\text{CF}}$ using $\mathbf{u}^{\text{F}}$ and the do-operation. It is important to note that counterfactual computation using these three steps requires complete access to the SCM $\mathcal{M}$, which is infeasible in many practical settings [32].

2.2 Approximating counterfactuals with causal generative models

While complete access to the SCM $\mathcal{M}$ is usually impractical, expert knowledge can provide us with the causal graph $\mathcal{G}$ [18, 19]. Using observed (factual) data $\mathbf{X}^F$ and $\mathcal{G}$, recent advancements in *causal generative models* have accurately *approximated* unknown $\mathcal{M}$ (information of unknown $\mathbf{U}, \mathbf{F}$ in § 2.1) by leveraging deep-learning methods like graph NNs [39, 55], diffusion models [38], and normalizing flows [15, 20]. Training these deep-learning-based models successfully approximate $\mathcal{M}$ by approximating *causal abduction* (estimating the exogenous $\mathbf{U}$ by mapping $\mathbf{X}^F$ to latent $\mathbf{Z}^F$) and *causal prediction* (mapping $\mathbf{Z}^F$ to features $\mathbf{X}^F$). Additionally, these trained models estimate *causal actions* by modeling interventions on the approximated $\mathcal{M}$ to compute counterfactual $\mathbf{X}^{CF}$. Hence, once trained, these novel causal generative models can successfully approximate unknown SCMs and efficiently generate causal counterfactuals using *model inference*, simplifying the abduction-action-prediction process [31].

We focus on the state-of-the-art in deep causal estimators, the causal normalizing flows [15]. Causal flows have been shown to be highly accurate in their causal approximations while working with minor assumptions: invertible and differentiable structural functions, acyclic graphs, and causal sufficiency (exogenous variables are mutually independent). When these assumptions hold, Javaloy et al. [15] showed that any SCM $\mathcal{M}$ can be modeled using autoregressive Triangular Monotonic Increasing (TMI) maps. Consequently, modeling these maps using autoregressive normalizing flows (ANF) **)7** allows causal flows to satisfy *causal consistency* [15] regarding $\mathcal{M}$.

By satisfying causal consistency, causal flows accurately approximate $\mathcal{M}$ by isolating the exogenous $\mathbf{U}$. Hence, the latent $\mathbf{Z}$ of the flows model **)7** correspond precisely to the exogenous $\mathbf{U}$. Causal consistency also allows a *single invertible* flows model to leverage $\mathbf{Z}$, correspondingly $\mathbf{U}$, to faithfully approximate both the causal *abduction* and *prediction* steps (§ 2.1) [31]. To approximate the causal *action* step, causal flows implement a different variation of the do-operator $\mathrm{do}(\mathbf{X}_{\mathcal{I}} = \backslash^{\mathcal{I}})$. Instead of modifying features, causal flows directly modify exogenous factors $\mathbf{U}$ (corr. $\mathbf{Z}^F$) corresponding to the respective intervened features $\mathcal{I}$. Correspondingly, the *intervened exogenous* distribution [15] is:

$$\mathbf{Z}^{\mathcal{I}} = \prod_{i \in \mathcal{I}} \chi\left(\mathbf{X}_i = \backslash_i^{\mathcal{I}}\right) \cdot \prod_{j \notin \mathcal{I}} p_j\left(I_j^F\right) \tag{1}$$

Here, χ is the *Dirac delta* that sets the exogenous value for each intervened feature $i \in \mathcal{I}$ such that the intervened value is $\backslash_i^{\mathcal{I}}$. Effectively, under mild assumptions on $\mathcal{M}$, causal flows can use $\mathbf{X}^F$ and $\mathcal{G}$ to train a single NN model to approximate the three-step causal counterfactual computation [31]. Hence, using model inference allows for time-efficient counterfactual generation $\mathbf{X}^{CF}$ from $\mathbf{Z}^{\mathcal{I}}$.

2.3 Finding actionable interventions for algorithmic recourse

Algorithmic recourse aims to provide individuals who received undesired, negative predictions from an algorithmic decision-making classifier with a mechanism to overturn the predictions through meaningful recommendations. While existing works on *nearest counterfactual explanation* [24, 27, 41, 49] are related, they assume features are *independently manipulable*. Hence, these methods *do not* have the realistic consideration that changing one feature impacts and automatically changes other features (e.g., increasing income automatically improves savings, all else remaining the same). As a result, such recourse methods usually cannot provide optimal, actionable recourse suggestions that individuals can follow in reality [18].

To find recourse recommendations that are *actionable*, *optimal*, and *successful*, recent works [18, 19, 51] have shown that it is imperative to consider the *causal relationships* between the features in the recourse solution. Such solutions allow individuals to perform *meaningful interventions* on their features in a *causal algorithmic recourse* mechanism. For any individual with features $\mathbf{x}^F$ and set of feasible actions $\mathcal{A}$, *causal algorithmic recourse* uses the SCM $\mathcal{M}$ to find *optimal causal actions* $\boldsymbol{a}^*$. Recourse actions are fundamentally some unknown function of the individual features $\mathbf{x}^F$ and SCM $\mathcal{M}$. Based on the causal do-operator $\boldsymbol{a}^* = \mathrm{do}(\mathbf{X}_{\mathcal{I}} = \backslash^{\mathcal{I}})$, the actions change the values of certain features $\mathcal{I}$ to $\backslash^{\mathcal{I}}$. Based on these values, causal recourse uses the three-step *abduction-action-prediction* method on $\mathcal{M}$ (§ 2.1) to estimate the resulting *counterfactual features* $\mathbf{x}^{CF}$. Assuming a binary downstream predictive algorithm, to ensure *optimality*, $\boldsymbol{a}^*$ must lead to counterfactual $\mathbf{x}^{CF}$ that ensures a positive algorithmic prediction $h(\mathbf{x}^{CF}) = 1$ at *minimal cost*.

$$\boldsymbol{a}^* \in \underset{\boldsymbol{a} \in \mathcal{A}(\mathbf{x}^F)}{\arg\min}\ \mathrm{cost}(\boldsymbol{a}; \mathbf{x}^F) \quad \text{subj"to} \quad h(\mathbf{x}^{CF}(\boldsymbol{a}, \mathbf{x}^F)) = 1 \tag{2}$$

While this framework can provide optimal solutions in theory, it has seen limited practical use owing to the complexity of the underlying optimization problem [19].

2.4 Practical challenges of causal recourse

The first challenge in solving the causal recourse problem (Eq. 2) is the unavailability of the exact SCM $\mathcal{M}$ in most practical settings [32]. To overcome this challenge and still compute counterfactual estimates of recourse actions, we solve the recourse problem using the recent advancements in deep causal generative models. As shown in § 2.2, using observed data and the causal graph $\mathcal{G}$, these models accurately approximate SCMs and compute counterfactuals efficiently [1]. Hence, if we incorporate these powerful models into the recourse pipeline, we can deploy recourse in larger-scale practical settings.

However, even with accurate causal counterfactual estimation, the main challenge remains the inherent *combinatorial complexity* of the recourse problem. Specifically, finding optimal $\boldsymbol{a}^*$ in Eq. 2 requires solving a nested combinatorial problem to find i) the feature combination ($\mathcal{I}$) for intervention and ii) the values ($\backslash^{\mathcal{I}}$) to assign to these features. Moreover, the recourse problem must be solved for each individual separately. This *unamortized* nature of the complex recourse problem leads to significant computation times (see results in § 5.2). Consequently, this complexity limits the possibility of providing instantaneous recourse recommendations [11, 17, 26, 42, 54] in large-scale settings (e.g., online lending programs) with many features and individuals seeking recourse.

[1] Although GPs and CVAEs were introduced in [19] to address causal recourse in the absence of complete knowledge of $\mathcal{M}$, they *only* solved for interventional suggestions, not counterfactual. Moreover, these approaches were inefficient as they solved individual combinatorial optimization problems.

(a) Example of causal graph G (b) Proposed method CARMA

Figure 1: (a) Causal graph for Loan (immutable : gender , age , actionable : loan amount ! , income , savings (, mutable/non-actionable : duration). (b) CARMA pipeline using deep causal generative models to amortize causal abduction (estimate exogenous Z^F) and prediction (counterfactual X^{CF}). The novel mask and action networks (shaded) amortize causal actions where the former predicts optimal feature targets I , and the latter predicts optimal value changes in latent $\frac{I}{Z}$ such that $^1X^{CFo} = 1$.

Practical desiredata. We aim to identify cost-e ective causal recourse suggestions for a downstream algorithmic classi cation model. This involves utilizing either gradient access (gradient of predictions with respect to input features) or white-box access (complete access to model parameters) [17]. We focus on binary classi ers in large-scale, real-world scenarios like online automated lending programs. In such contexts, we aim to deploy a causal recourse optimization method that: i) is optimal (minimizes cost), ii) ensures high validity (successful recourse interventions for the majority of the population), and iii) requires minimal computation time.

3 PROPOSED METHOD

We overcome the complexities of the causal recourse optimization problem by introducing CARMA, a novel method to solve Causal Algorithmic Recourse with Model-based Amortization (Fig. 1b). CARMA casts the combinatorial optimization problem as a predictive task. This approach leverages a fundamental insight of causal recourse (Ÿ 2.3): since the underlying causal model generating each individual's features, as well as the downstream decision-making classi er, are common for all individuals, recourse actions can be predicted from an individual's features. Given gradient or white-box access to the downstream algorithmic binary classi er/decision-making model $^1 o$ and to a pre-trained causal generative model, CARMA simpli es the combinatorial recourse optimization problem (Ÿ 2.4) into a predictive task that estimates the function that maps feature values and causal relations to optimal recourse interventions. Furthermore, by learning this task on ample representative data, CARMA can amortize and deliver optimal recourse suggestions at scale, demanding minimal compute time as it circumvents the need for individual optimizations.

3.1 CARMA: Overview

As shown in Fig. 1b, CARMA can accurately compute counterfactuals even with limited causal information by leveraging deep causal generative models (Ÿ 2.2). Speci cally, we design CARMA as a exible framework that can incorporate any recent (or future) deep causal generative model that is di erentiable and provides exogenous estimates through some latent factors. These state-of-the-art causal generative models also help CARMA to amortize the causal abduction and prediction steps [31] by performing e cient counterfactual computations during inference time.

Moreover, CARMA amortizes causal actions [31] of causal recourse using a novel NN framework comprising two models that jointly predict optimal recourse interventions. First, CARMA's mask network (Ÿ 3.2) predicts which features to target for recourse by minimizing the loss function L^{mask} by leveraging data features X^F and causal knowledge that has been approximated by the causal generative model) [15, 39] via latent $Z^F =)\ ^1X^{Fo}$. Secondly, the action network (Ÿ 3.3) predicts the interventional action values in the exogenous latent space as $\frac{I}{Z}$ by minimizing its loss function L^{action} using estimated causal knowledge of Z^F and the mask network's output I .

For a downstream binary classi er, pre-trained causal generative model) , and dataset of the features of individuals rejected by , i.e., $?^1X^F \mid {}^1X^F = 0^{oo}$, we train CARMA's models with parameters $\bullet q$ jointly[2] by optimizing the loss:

$$L^{recourse}_{\bullet q \bullet \_}\ ^1x^{CF} \bullet x^{Fo} = \min_{\bullet q} \Big[L^{mask}_{|}\ ^1x^{Fo} \ , \underbrace{L^{action}_{q}\ ^1x^{CF} \bullet x^{Fo}}_{\text{intervention loss}} \Big]$$

$$, \_ \underbrace{\text{HingeLoss}\ ^1x^{CFo}; 1 \bullet V}_{\text{outcome loss}} \quad (3)$$

This loss function follows the overall objective of causal and non-causal recourse [19, 24, 51]. The recourse outcome loss ensures a positive downstream algorithmic decision that we model following [24] with the hinge loss [37]. In this loss, the V hyperparameter controls the classi er margin distance, with larger values encouraging the recourse counterfactual x^{CF} to result in larger positive prediction probabilities. The _ parameter balances the outcome loss with the recourse intervention loss that controls the intervention's optimality regarding cost. The intervention loss is optimized by jointly training the primary components of CARMA: the mask and action networks. Using unsupervised learning, we readily optimize these NN models requiring no access to ground-truth data of optimal causal recourse actions.[3] We next detail these two NN models of CARMA and their corresponding loss functions.

[2] Regarding optimization, the mask and action networks are jointly trained and act as one model (like VAE). Separation is for aiding semantic interpretability.

[3] While prior methods also did not require true actions, prior causal approaches needed the true M and solved individual optimization problems.

3.2 Mask network: Determining target features for recourse intervention

CARMA simplifies the first part of the combinatorial recourse problem (Ÿ 2.4) by using the mask network to amortize finding the intervention targets I. For amortizing finding I, the mask network utilizes the amortized causal abduction capabilities of state-of-the-art causal generative models. Specifically, the latents Z^F of the causal generative models approximate the causal information and the exogenous factors (Ÿ 2.2). The mask network leverages this capability, training on Z^F of individuals rejected by the downstream classifier to automatically predict I.

We instantiate the mask network as a deep, feed-forward NN @ with parameters Γ. The model @ is trained with the estimated causal latents Z^F (of individuals who received the unfavorable algorithmic decision) to predict the mask probability vector $\alpha = \{\alpha_{1:D}, \ldots, \alpha_{K:D}\}$ where K is the number of actionable features. The mask value $\alpha_{i:\{1\}}$ (correspondingly $\alpha_{i:\{0\}}$) denotes the probability of selecting (correspondingly not selecting) the feature i for recourse intervention. The mask probability is computed from the raw logit outputs c of @ following the Gumbel trick [14], i.e.,

$$\alpha_{i:k} = \frac{\exp\{(\log(c_k) + \delta_k)/\tau\}}{\sum_{j=0}^{1} \exp\{(\log(c_j) + \delta_j)/\tau\}} \quad \text{for } k = 0,1 \tag{4}$$

Here, $k \in \{0,1\}$ denotes not selecting or selecting a particular feature. Parameters δ are sampled from the Gumbel distribution and τ is a temperature hyperparameter that interpolates the estimated distribution of the target feature selection between a categorical and uniform one-hot distribution [14].

Estimating the feature selection probability α_i from Z^F is equivalent to using Z^F to fit a Bernoulli distribution with $\theta = \alpha_{i:1}$. This probabilistic estimation is typically performed using variational Bayesian methods [22] that require considering a prior distribution. We assume a Bernoulli(θ^o) prior over each actionable feature X_i. Estimating the probabilistic distribution minimizes the divergence between the prior and the mask network's estimated posterior.

$$L_{\Gamma}^{\text{mask}}(x^F) = \min_{\Gamma} \sum_{i=1}^{K} \left(@(\alpha_{i:1} \mid Z^F = f_{\phi}(x^F)) \,\|\, \text{Bernoulli}(\theta^o) \right) \tag{5}$$

In the general case, we fix the prior of each feature to $\theta^o = 0.5$. Hence, the prior assumes that selecting each actionable feature i has a 0.5 probability. As we show later in our case study (Ÿ 5.5), these prior probabilities may be changed to reflect the population's preferences regarding changing different features for recourse.

However, in the case of causal recourse, we can either select a feature for intervention or not select it; hence, we require a binary vector I. This difficulty is overcome during CARMA's training with the Gumbel trick as we can sample the binary I from the probabilities α to determine which features are selected for intervention. For each feature at index i, $I_i \sim @(\alpha_{i:1} \mid Z^F)$. However, traditional gradient-based training used for neural networks does not work with sampling operations. CARMA circumvents this issue by utilizing the straight-through Gumbel trick [14]. At a high level, in gradient-based learning, this method allows performing a forward pass (inference) by sampling a binary I while using the probabilistic α for backpropagating the gradients during optimization. This methodology allows the optimization to learn a biased estimate while training the mask network model. At test time, the sampling may be replaced by a simple thresholding on the probability α.

3.3 Action network: Predicting interventional action values

Along with determining the target features I, solving for recourse interventions requires predicting the interventional action values Δ_I for the intervening features I (Ÿ 2.4). While our mask network automates the estimation of I, we predict optimal interventional values through a separate action network. Modeled as a feed-forward NN Ω_η, the action network leverages the amortized causal abduction capability of recent causal generative models, utilizing the causal information encoded in Z^F (of individuals rejected by classifier f). With Z^F, the action network uses the mask network's predicted I for training to predict the optimal recourse action. The optimal action is predicted in the causal latent space as Δ_{Z_I}. This output method allows leveraging the amortized causal prediction capability of the pre-trained causal generative models, efficiently converting Δ_{Z_I} to post-recourse counterfactual features x^{CF}.

Hence, for an individual x^F, our action network Ω_η takes I (features targeted for intervention by mask network) and the latent factors Z^F as input to predict the intervened latent values Δ_{Z_I}, i.e., $\Delta_{Z_I} = \Omega_\eta(I; Z^F)$. To ensure that the predicted intervention actions Δ_{Z_I} lead to a favorable decision, we need to compute the resultant post-recourse counterfactual features x^{CF}. Estimating x^{CF} from our predicted Δ_{Z_I} is straightforward owing to the amortized causal prediction achieved by the pre-trained causal generative models. For example, the causal flows [15] input the binary target vector I and the intervened latent factors $z + \Delta_{Z_I}$ to estimate counterfactual features x^{CF} from the inverse of the ANF-based generative model f^{-1} [15] (for other models like VACA x^{CF} is generated using a separate decoder model, see Appendix C) as $x^{CF} = f^{-1}(z + \Delta_{Z_I}; I)$. Correspondingly, we minimize the loss regarding the cost of our predicted interventions:

$$L_{\Omega}^{\text{action}}(x^{CF} \mid x^F) = \min_{\Omega} \text{cost}(x^{CF} \mid x^F)$$
$$\text{where } x^{CF} = f^{-1}(z + \Omega_\eta(I; Z^F); I) \tag{6}$$

For simplicity, we measure the cost using ℓ_2 norms. Specifically, for any recourse-seeking individual, we measure $\text{cost} = \frac{1}{K} \| I \odot (x^F - x^{CF}) \|_2$, where K indicates the number of actionable features, I the features targeted for recourse, x^{CF} the feature values post-recourse. We can also incorporate feature preferences regarding recourse using weights for each feature in the cost function (see Appendix D and the case study in Ÿ 5.5). Our framework is also readily expandable to work with differentiable versions of more complex cost functions, e.g., distributional effort [10], and percentiles [45].

4 RELATED WORK

Counterfactual explanations and (non-causal) algorithmic recourse. For algorithmic decision-making systems, the rich literature on counterfactual explanations has attempted to provide methods to find what features describing the individual would need to change to achieve the desired output [16]. A

framework for finding such nearest counterfactual explanations was defined in [52]. Later, several methods to solve for counterfactual explanations were proposed that span across techniques like SMT solvers [16], generative models (e.g., VAEs and GANs) [24, 28, 29], and evolutionary algorithms [6, 41]. Some methods also proposed combining the task of finding nearest counterfactuals with other properties such as sparsity [45], diversity [27], and robustness [44]. In contrast to the concept of counterfactual explanations, Ustun et al. [45] introduced the ideal of algorithmic recourse in the context of actionability of features (only certain features can be acted upon by individuals). Although this work differentiated recourse from simple explanations, it did not consider the data features to have causal relations, instead assuming them to be independently manipulable. Considering the richness and vastness of the field of recourse and explanations, our discussion here is limited. Hence, we refer to the foundational surveys [17, 48] that explored many other aspects of these frameworks. In this context, our work specifically expands the literature on causal algorithmic recourse [18, 19, 23, 51].

Amortized (non-causal) recourse. Recent studies have explored the automation of algorithmic recourse and counterfactual explanations, intending to provide solutions in minimal time. The DiCE-VAE [24] explored amortizing non-causal recourse/counterfactual explanations using the statistical generative model, VAE. This approach acts as an amortization baseline in our evaluations. Another approach, FastCFE [49], used a reinforcement learning pipeline with proximal policy optimization (PPO) to find counterfactual explanations. FastCFE generated sequential explanations by changing one feature at a time, albeit with the limitation of discretizing the feature space to use PPO efficiently. While these methods [24, 49] incorporated unary/binary constraints for modeling some causal characters (e.g., monotonicity), unlike CARMA, they did not leverage the expansive causal mechanisms using an SCM, hindering their ability to provide actionable causal interventions. Distinct from other methods, Guo et al. [17] considered solving for recourse as ad hoc training a counterfactual explanation solver and the classifier concurrently. While this method provided amortization, it did not use causal relations, and only works if we can retrain the classifiers. In contrast, our work introduces a novel post hoc amortized recourse method that considers a fixed algorithmic classifier and incorporates causal information using SCMs to generate instant and actionable recourse recommendations.

5 EXPERIMENTAL EVALUATION

Our experimental evaluations aim to answer the following questions regarding CARMA[4] (more analyses in Appendix F):

- (R1): How optimal is CARMA's amortization of the causal algorithmic recourse optimization problem?
- (R2): How does CARMA's amortized recourse impact the feature distributions compared to unamortized recourse?
- (R3): How flexible is CARMA in using different causal generative models, and how does it impact performance?
- (R4): How can CARMA incorporate population-level feature preferences to provide tailored recourse suggestions?

[4]Our code is publicly available at www.github.com/ayanmaj92/carma-recourse/.

5.1 Setup

Datasets and classifiers. For R1, we design synthetic datasets (triangle, collider, chain) with linear (LIN) and non-linear (NLIN) causal relations [39] where three out of four features are actionable. For R1-R4, we also consider the more complex loan dataset [19] (Fig. 1a), modeling the German Credit data [12] with four out of seven features actionable (education level, loan amount, income, and savings). Regarding downstream decision-making classifiers, we train logistic regression classifiers for LIN data and NNs for NLIN and Loan. Refer to Appendix A and E for more details.

Methods. For our analysis, we train CARMA that uses the state-of-the-art causal flows (CF) [15] model as the deep causal generative model to estimate causal relations and efficiently approximate counterfactuals. However, to test R3, we also use the Graph NN (GNN) based VACA generative model [34] in CARMA. To compare CARMA in answering our recourse analysis questions, we compare it to the causal unamortized oracle recourse method (M) [19] that has perfect access to the causal relations, i.e., the SCM $\mathcal{M}$ and computes unamortized recourse. Furthermore, to analyze the amortization performance (R1), we compare CARMA's running time to the amortized but non-causal DiCE-VAE [24]. More details about these baseline recourse methods and their training setup can be found in Appendix B and E.

5.2 Can CARMA optimally amortize causal recourse?

We aim to understand if CARMA can optimally approximate the causal recourse optimization process to provide amortized solutions. To this end, we compare the recourse performance of CARMA with the causal unamortized oracle (M) and the amortized non-causal DiCE-VAE in Table 1. In the table, for unseen test samples of different datasets, we report how the different recourse methods perform in terms of the *cost* (ℓ_2 of intervened feature changes), *optimization coverage* (percentage of test set for which the optimization can find recourse solutions), *mean number of actions* (average features acted on per individual), *mean prediction probability* (average post-recourse classifier probability $\mathbb{1}-h(\mathbf{x}^{CF})$), *compute time* (average time to provide recourse solution per datum), and *causal validity* (for causal recourse methods what percentage of recourse solutions are successful when analyzed under the true, hidden SCM).

We observe that across the different setups, CARMA provides optimal causal recourse suggestions, providing *cost* as low as and nearly reaching the perfect (100%) coverage of the causal oracle (M). Hence, even without access to the exact SCM, CARMA, when deployed, can find optimal recourse suggestions for almost all individuals in the unseen test dataset. For instance, for the larger Loan dataset, CARMA provides amortized recourse suggestions that incur cost of 7.13 ± 1.27, close to the 5.19 cost of M. When deployed, in addition to not requiring full causal knowledge on the SCM, CARMA finds the recourse solutions in significantly lower time compared to M, requiring around 1 millisecond (msec) per individual compared to several seconds for M. For instance, even for the simplest linear (LIN) triangle data, M can take around 8 seconds to solve the recourse problem and provide optimal suggestions, whereas CARMA provides equally optimal suggestions in 0.77 msec. This improvement is more apparent for the larger Loan

Table 1: Comparing CARMA 's amortized recourse performance on unseen test data (number of data shown in # test-data) to causal unamortized oracle M and amortized non-causal DiCE-VAE across multiple datasets (10 seeds). Metrics include intervention cost (2, lower be er), optimization coverage/success rate (%, higher better), mean intervened feature count, mean post-recourse prediction probability, compute time per datum (milliseconds, lower be er), and validity under true SCM (%, higher be er). 7 indicates non-causal.

Dataset (features)	# test-data	Method	recourse cost (2 100)	optim. coverage (%)	mean actions (per datum)	mean predict probability	time per datum (msec)	causal validity (%)
triangle-LIN		M	11.62	100.0	1.11	0.59	8294.48	100.0
actionable: 3	2248	DiCE-VAE	33"57 2"58	100"0 0"0	3"0 0"0	0"93 0"02	0"32 0"17	7
total: 4		CARMA	11"36 4"13	99"92 0"25	1"45 0"33	0"72 0"08	0"77 0"0	99"94 2"5
collider-LIN		M	9.99	100.0	1.48	0.54	9992.56	100.0
actionable: 3	2284	DiCE-VAE	23"98 2"03	100"0 0"0	3"0 0"0	92"23 1"61	0"32 0"22	7
total: 4		CARMA	10"36 4"06	99"96 0"07	2"37 0"47	0"61 0"03	0"77 0"1	99"98 1"48
chain-LIN		M	10.39	100.0	1.42	0.55	9456.32	100.0
actionable: 3	2095	DiCE-VAE	25"98 2"00	100"0 0"0	3"0 0"0	0"92 0"01	0"32 0"22	7
total: 4		CARMA	10"62 5"12	99"91 0"27	2"26 0"61	0"59 0"02	0"74 0"0	99"91 3"0
triangle-NLIN		M	9.82	100.0	1.07	0.72	10741.22	100.0
actionable: 3	2106	DiCE-VAE	24"91 1"78	99"92 0"23	3"0 0"0	0"95 0"01	0"42 0"22	7
total: 4		CARMA	8"95 3"13	100"0 0"0	1"96 0"64	0"77 0"13	1"01 0"0	99"89 3"3
collider-NLIN		M	3.91	100.0	1.88	0.583	14379.07	100.0
actionable: 3	2179	DiCE-VAE	8"74 0"73	99"18 0"64	3"0 0"0	0"90 0"02	0"43 0"22	7
total: 4		CARMA	4"19 1"10	99"74 0"64	2"62 0"42	0"69 0"05	0"98 0"0	99"74 5"06
chain-NLIN		M	9.74	100.0	1.24	0.65	11991.38	100.0
actionable: 3	2320	DiCE-VAE	21"46 1"11	99"56 0"5	3"0 0"0	0"92 0"01	0"42 0"22	7
total: 4		CARMA	11"81 2"05	99"99 0"04	2"32 0"62	0"76 0"06	1"05 0"35	99"98 1"31
Loan		M	6.25	100.0	1.88	0.63	53580.98	100.0
actionable: 4	2364	DiCE-VAE	13"32 1"05	99"98 0"02	4"0 0"0	0"96 0"00	0"48 0"2	7
total: 7		CARMA	6"69 1"36	99"94 0"1	2"89 0"29	0"85 0"05	1"17 0"1	99"95 2"16

data (1.17 msec compared to 1 minute), indicating the clear advantage of amortization regarding compute time. The computation times required by CARMA are in the same order of msec compared to existing amortized methods like DiCE-VAE (takes around 0"48 msec compared to CARMA's 1.17 msec). But, since DiCE-VAE cannot incorporate the underlying causal information of the SCM, its suggestions are sub-optimal, resulting in signi cantly higher cost. Similarly, the amortized DiCE-VAE solver is inherently non-sparse, intervening on all actionable features. In contrast, CARMA, by being an amortized recourse method that uses approximated causal information, intervenes on less features than DiCE-VAE. Hence, unlike non-causal methods like DiCE-VAE, CARMA can provide sparse recourse solutions even in the absence of sparsity constraints by leveraging the causal information. However, CARMA is not as sparse as the oracle M , acting on more features on average across all setups. Additionally, note that CARMA can provide 100% causal validity. This means that the approximate recourse suggestions of CARMA are also successful if the individuals performed them in reality under the true, unobserved SCM M . Finally, Table 1 also shows that amortized recourse methods are generally more conservative. So, to ensure successful recourse for all individuals, the suggestions provided by amortized methods like CARMA result in higher predicted probabilities (0"85 0"05 for Loan) from the downstream algorithmic classi er post-recourse when compared to the unamortized oracle M (0.63 for Loan). Nonetheless, more conservative suggestions (CARMA) may possess some desirable characteristics in comparison to the unamortized minimum-cost suggestions M( ), particularly in terms of their robustness to minute feature perturbations [3]. A comprehensive examination of this aspect is deferred to future work.

The results in Table 1 show that CARMA can be trained using a sample of observational data and knowledge of just the causal graph G to be deployed on large populations to provide instantaneous, amortized recourse suggestions that are optimal in terms of cost and causally valid under the true, hidden causal mechanisms de ned by the SCM. Next, we compare the impact of CARMA's amortized causal interventions on the features to the oracle M 's unamortized ones.

5.3 How does causal recourse impact the feature distributions?

In this section, we explore how performing the recourse interventions shifts the di erent features for the Loan dataset, comparing the impact of amortization in CARMA to the unamortized M . We

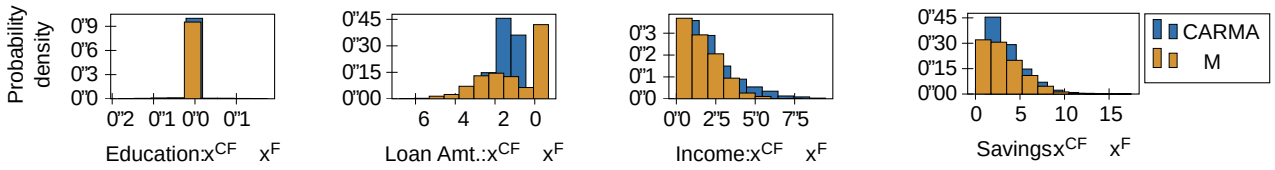

Figure 2: Histograms comparing post-recourse feature shifts $^1x^{CF} - x^{Fo}$ between CARMA and unamortized causal M .

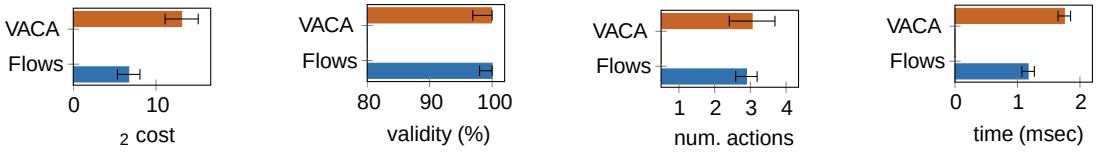

Figure 3: Recourse metrics on Loan across 10 seeds for CARMA using Causal Flows and VACA as the generative model.

visualize our analysis in Fig. 2, where we plot the histograms of the feature value shifts occurring due to recourse, with the values $(x^{CF} - x^F)$ along the x-axis and the histogram probability of the distribution along the y-axis. We plot the histograms for our amortized method CARMA and the ideal oracle M with blue and orange bins, respectively. From Fig. 2, we observe how the di erent features are shifted due to recourse interventions, with more signi cant shifts of $^1x^{CF} - x^{Fo}$ for CARMA on average. Looking at each feature individually, for Education, both CARMA and M induce near zero value shifts. Hence, although CARMA interventions targeted Education (Ÿ F.1), these interventions were of minimal values. For Income and Savings, CARMA and M interventions resulted in increasing these features, with the unamortized CARMA requiring larger shifts. For Loan amount, CARMA and M interventions result in decreasing the value. However, for this feature, there is a di erence in how CARMA and M intervene. Our amortized method CARMA requires low change for most individuals, with $^1x^{CF} - x^{Fo}$ 1"5 having higher probabilities, while imposing signi cant changes $(^1x^{CF} - x^{Fo}$ 3) for very few individuals (probability around 0.05). On the other hand, while M interventions result in no change for most individuals ($^1x^{CF} - x^{Fo}$ 0 has 0.43 probability), it requires some individuals to have large changes ($^1x^{CF} - x^{Fo}$ 3"5 have probabilities as high as 0.1 to 0.15).

Hence, post-recourse distributional shifts a ected by amortization. For the Loan data, we observed that the unamortized M generally induces lower distributional shifts than CARMA. But, in some cases, in contrast to unamortized methods, amortization may cause slight changes in a feature for many but large ones for a few. However, the implication of this di erential behavior needs to be analyzed for particular deployment scenarios. Now that we have studied the nature of amortized recourse interventions, we analyze the exibility of CARMA in using di erent causal generative models.

5.4 How does the choice of the causal generative model a ect CARMA?

We designed CARMA to be a exible amortized recourse framework capable of utilizing any novel causal generative model that o ers gradient access, latent exogenous estimation, and counterfactual computation. To study the exibility and analyze how the choice of the generative model might impact recourse, we compare CARMA training with causal ows (state-of-the-art) with another recent deep-learning-based method, VACA [39]. We compare the two versions of CARMA in Fig. 3, plotting the metrics cost, causal validity, average number of actions (features targeted), and compute time along the x-axes of the plots. We compare the performances of Causal Flows [15] and VACA [39] using horizontal bars with error bars representing the variance across random seeds. From Fig. 3, we see that while CARMA can successfully incorporate VACA, using it instead of causal ows results in less optimal amortization of causal recourse across all metrics. When using VACA in the pipeline, CARMA provides more costly recourse suggestions, intervenes on more features, and leads to slower amortization, requiring more compute time (almost twice as much as the Flows model requires). The variance across seeds also increases when we use VACA, as indicated by the error bars in the plots for the cost, validity, and number of actions. The superior performance when using ows (lower cost and variance) can be attributed to their ability to recover the exogenous factors [15], leading to more accurate counterfactual estimation. Likewise, the lower performance when using VACA can be explained by its inexact exogenous estimation and higher inaccuracies in counterfactual approximations [15, 39]. Moreover, VACA's increased compute time stems from its more computationally complex counterfactual approximation process (requiring combinatorial complex modi cation of the post-intervention adjacency matrix of G) [39]. See Appendix C for details. The only exception can be seen in the recourse suggestions' causal validity metric, which remains high for both VACA and Flows (99%), indicating a good approximation of the SCM.

These results show that CARMA is exible to incorporate different deep causal generative models to approximate the unknown causal relations and the counterfactual generation. Hence, future advancements providing more powerful models can be readily incorporated in CARMA, provided they allow di erentiability or gradient access. Nonetheless, choosing the proper causal generative model is vital to accurately approximate causal counterfactuals and achieve optimal amortized causal recourse. Next, we showcase a case study highlighting how to incorporate population-level feature preferences in amortized recourse to tailor the recourse suggestions.

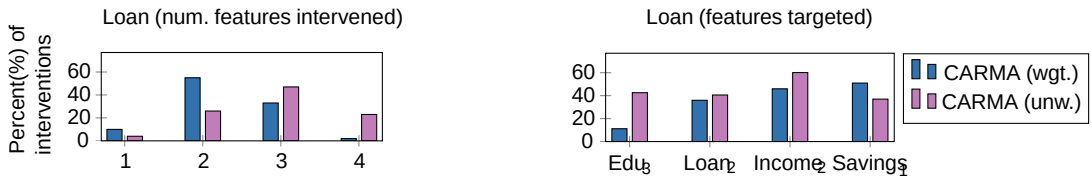

Figure 4: Comparing CARMA with preference weights (wgt.) to the unweighted vanilla (unw.) showing the percentage of test-time interventions: (left) altering particular feature counts, (right) targeting each actionable feature (weights in subscript) in the Loan data.

5.5 Case study: Incorporating population-level feature preferences

In contrast to previous analyses, we now consider that the population's preferences for changing different features for recourse may differ due to various factors. One such factor is time-related difficulty [2], where modifying some features (e.g., those higher up in the graph G) is more challenging since they require more time. For example, saving a specific amount of money may take less time than advancing from a Master's to a Ph.D. education level. Using the Loan data as a case study, we show how to incorporate such population-level preferences into CARMA's amortized causal recourse setup.

Setup. We consider the same actionable features: Education, Loan amount, Income, and Savings. However, now, we consider that the population provides different feature preferences, e.g., features deeper in the causal graph are more difficult to change, requiring more time [2]. The population provides this preference as feature weights (higher weight signifying more difficulty): f(Education, 3), (Loan amount, 2), (Income, 2), (Savings, 1)g. We incorporate this additional preference knowledge in CARMA's learning process. First, the weights denote the difficulty of feature changes. So, in CARMA's action network, we multiply the $_2$ cost of each feature's change with the corresponding weight in Eq. 6. Secondly, the weights denote the likelihood of targeting different features (the higher the weight, the lower the likelihood). So, in CARMA's mask network, we modify the prior Bernoulli $^1?^o$, reducing ? from default 0.5 when weights are $= 1$. Using these modifications, we tune CARMA to find the best model. We note that this best model provided optimal recourse with low cost (near 0") 09 matching the corresponding M) and high causal validity (near 100%) when measured on the held-out test data. More details on the setup can be found in Appendix D. For comparing the impact of the preferences on amortized recourse, we compare CARMA with preferences (wgt.) to the previous vanilla method using no weights (unw.).

Preferences lead to targeting different and fewer features. To understand the impact of the preferences on amortized recourse, in Fig. 4, we plot the percentage of test-time interventions targeting different feature counts (left) and specific features (right). We compare CARMA (wgt.) in blue to the vanilla CARMA (unw.) in violet. By incorporating preferences, CARMA (wgt.) acts mostly on two features (57%) and rarely on four. In contrast, CARMA (unw.) intervened primarily on three and four features. Hence, preferences allow CARMA to offer sparser suggestions, reducing the average number of targeted features from 2.89 (unw.) to 2.26 (wgt.). Additionally, following the preferences, CARMA (wgt.) increasingly targets the lowest weighted Savings (50% vs. 37% of test-time interventions in unw.) and infrequently targets the highest weighted Education (11% in wgt. vs. 42% in unw.). The preference weights also slightly reduce the percentage of interventions targeting Loan amount and Income. Thus, CARMA leverages population preferences via the weights to tailor the recourse suggestions, targeting more preferred features and, potentially, leading to sparser recourse interventions.

Preferences lead to different shifts in the features. To understand the impact of the preferences on amortized recourse, in Fig. 5, we plot histograms with the post-recourse feature value shifts (x^{CF} x^F) for the actionable features in the test data along the x-axis and the probability of the distribution of the values along the y-axis. We compare the recourse suggestions of CARMA with preferences (wgt. in blue) to the vanilla CARMA (unw. in violet). For more difficult features like Loan amount and Income, CARMA (wgt.) incorporates the feature preferences to reduce the shifts x^{CF} x^{Fo} compared to unw. For these features, note the comparatively higher probabilities of the wgt. shifts around zero. For the most difficult feature Education, we see minimal shifts for both approaches. However, for the least weighted Savings, CARMA (wgt.) causes more significant shifts than unw. Since the population marks this feature as easier, these changes should be easier to perform in practice. In conclusion, CARMA can integrate population feature preferences in amortized recourse, tailoring the recourse suggestions to intervene more and change the easier features (leaf nodes in G) like Savings

6 DISCUSSION AND REMARKS

To enable CARMA's practical deployment, the roles of certain assumptions and ethical factors need further study.

Assumptions. Analyzing the assumptions of the causal generative model in CARMA is crucial, especially in scenarios involving confounders or incomplete and inaccurate causal information in G. While understanding the impact of such errors is vital, we may be able to tackle them in CARMA by leveraging future advances in causal generative models. Similarly, the rigorous testing of CARMA and related causal recourse approaches necessitate the future curation of larger societal causal datasets. Moreover, though current recourse solutions assume access to exact feature values, robustness to uncertainties is pivotal [3]. While we assume stability in the data distribution and downstream classifier, exploring methods for robust recourse solutions amidst temporal shifts is essential [35], [e.g., considering feedback dynamics in the decision-making processes. Additionally, the assumption of gradient access to the classifier may not hold in certain real-world scenarios, necessitating exploring alternative approaches, such as leveraging genetic algorithms [41]. Extending the applicability of causal amortized

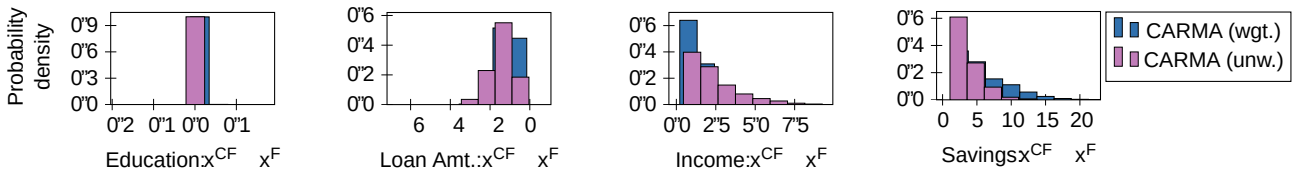

Figure 5: Histograms comparing post-recourse feature shifts between CARMA using preferences (wgt.) and vanilla (unw.).

recourse also requires looking beyond simple ℓ_p-norms for recourse cost, incorporating diverse effort functions [10] and the intricate impacts of time [2, 4]. Integrating more inclusive and realistic costs can better reflect individual capabilities in the recourse process [43] and should be explored in the context of amortization.

Other ethical considerations. The interplay of related ethical aspects with algorithmic recourse needs to be studied in the amortized context. First, providing interfaces for recourse has been shown to have privacy issues [30]. While amortized CARMA can help scale up recourse fairness auditing [51], designing efficient recourse mechanisms that are fair, e.g., regarding the costs across social groups [8, 51] and avoid societal segregation [5, 10] are interesting directions that need further research. Additionally, while recourse allows individuals to challenge decisions, it also raises the risk of exploitation and strategic gaming of the system. Hence, the interplay of amortized causal recourse with strategic behavior [9, 26] (gaming the classifier's prediction) and individual improvement [23] (in the ground truth) needs thorough exploration. Finally, the long-term impact of recourse on the utility and fairness of different stakeholders in automated decision-making [13, 21, 34] must also be studied to ensure optimal and responsible deployment.

7 CONCLUSION

Our work tackled the challenges of implementing causal algorithmic recourse in large-scale settings, aiming to deliver instant and effective recommendations to overcome negative algorithmic decisions. Although previous work highlighted the benefits of causal recourse, its practicality has been constrained by the need for complete causal information and the computational complexity of existing approaches. To overcome these issues, we introduced a method called CARMA, which offers flexibility and efficiency in providing optimal recourse recommendations.

CARMA uses state-of-the-art causal generative models and neural networks to provide a novel data-driven predictive approach that overcomes the limitations of existing methods. It offers optimal causal recourse interventions with significantly reduced compute times. Moreover, our case study demonstrated how CARMA can customize recourse suggestions based on population-level feature preferences, considering factors like difficulty or time. Based on the promise of our initial results, we aim for our work to promote future research toward enhancing the accessibility of recourse mechanisms for individuals affected by negative automated decisions, especially in complex, large-scale application scenarios.

ACKNOWLEDGMENTS

We thank Adrián Javaloy and Pablo Sánchez-Martín for providing invaluable guidance regarding the incorporation of the causal generative models. We also thank the anonymous reviewers for providing valuable feedback on our manuscript and Rose Hoberman for providing guidance during the writing process. This project is funded by the Deutsche Forschungsgemeinschaft (DFG) grant number 389792660 as part of the Transregional Collaborative Research Centre TRR 248: Center for Perspicuous Computing (CPEC).

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

[2] Isacco Beretta and Martina Cinquini. 2023. The Importance of Time in Causal Algorithmic Recourse. In Explainable Artificial Intelligence - First World Conference, xAI 2023, Lisbon, Portugal, July 26-28, 2023, Proceedings, Part I (Communications in Computer and Information Science, Vol. 1901). Springer, 283 298. https://doi.org/10.1007/978-3-031-44064-9_16

[3] Ricardo Dominguez-Olmedo, Amir H Karimi, and Bernhard Schölkopf. 2022. On the adversarial robustness of causal algorithmic recourse. In International Conference on Machine Learning. PMLR, 5324 5342.

[4] João Fonseca, Andrew Bell, Carlo Abrate, Francesco Bonchi, and Julia Stoyanovich. 2023. Setting the Right Expectations: Algorithmic Recourse Over Time. In Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization. 1 11.

[5] Ruijiang Gao and Himabindu Lakkaraju. 2023. On the impact of algorithmic recourse on social segregation. In International Conference on Machine Learning. PMLR, 10727 10743.

[6] Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. 2019. Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems 34, 6 (2019), 14 23.

[7] Hangzhi Guo, Thanh H Nguyen, and Amulya Yadav. 2023. CounterNet: End-to-End Training of Prediction Aware Counterfactual Explanations. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 577 589.

[8] Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. 2019. Equalizing recourse across groups. arXiv preprint arXiv:1909.03166 (2019).

[9] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. 2016. Strategic classification. In Proceedings of the 2016 ACM conference on innovations in theoretical computer science. 111 122.

[10] Hoda Heidari, Vedant Nanda, and Krishna Gummadi. 2019. On the Long-term Impact of Algorithmic Decision Policies: Effort Unfairness and Feature Segregation through Social Learning. In International Conference on Machine Learning. PMLR, 2692 2701.

[11] Denis J Hilton. 1990. Conversational processes and causal explanation. Psychological Bulletin 107, 1 (1990), 65.

[12] Hans Hofmann. 1994. Statlog (German Credit Data). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77.

[13] Yaowei Hu and Lu Zhang. 2022. Achieving long-term fairness in sequential decision making. In Proceedings of the AAAI Conference on Artificial Intelligence Vol. 36. 9549 9557.

[14] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee

[15] Adrián Javaloy, Pablo Sánchez-Martín, and Isabel Valera. 2024. Causal normalizing flows: from theory to practice. Advances in Neural Information Processing Systems 36 (2024).

[16] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions. In International Conference on Artificial Intelligence and Statistics. PMLR, 895 905.

[17] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. 2022. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. Comput. Surveys 55, 5 (2022), 1 29.

[18] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic recourse: from counterfactual explanations to interventions. In Proceedings of the 2021 ACM conference on fairness, accountability, and transparency 353 362.

[19] Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. Advances in Neural Information Processing Systems 33 (2020), 265 277.

[20] Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. 2021. Causal autoregressive ows. In Proceedings of International Conference on Arti cial Intelligence and Statistics (AISTATS) (Vol. 24. PMLR.

[21] Niki Kilbertus, Manuel Gomez Rodriguez, Bernhard Schölkopf, Krikamol Muandet, and Isabel Valera. 2020. Fair Decisions Despite Imperfect Predictions. In Proceedings of the Twenty Third International Conference on Arti cial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 108). Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 277 287.

[22] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In 2nd International Conference on Learning Representations, ICLR 2014, Ban , AB, Canada, April 14-16, 2014, Conference Track Proceedings. Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

[23] Gunnar König, Timo Freiesleben, and Moritz Grosse-Wentrup. 2023. Improvement-focused causal recourse (ICR). In Proceedings of the AAAI Conference on Arti cial Intelligence Vol. 37. 11847 11855.

[24] Divyat Mahajan, Chenhao Tan, and Amit Sharma. 2019. Preserving causal constraints in counterfactual explanations for machine learning classi ers. arXiv preprint arXiv:1912.03277 (2019).

[25] Vishwali Mhasawade and Rumi Chunara. 2021. Causal multi-level fairness. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society 784 794.

[26] Tim Miller. 2019. Explanation in arti cial intelligence: Insights from the social sciences. Arti cial intelligence 267 (2019), 1 38.

[27] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classi ers through diverse counterfactual explanations. In Proceedings of the 2020 conference on fairness, accountability, and transparency 607 617.

[28] Daniel Nemirovsky, Nicolas Thiebaut, Ye Xu, and Abhishek Gupta. 2022. CounteRGAN: Generating counterfactuals for real-time recourse and interpretability using residual GANs. In Uncertainty in Arti cial Intelligence PMLR, 1488 1497.

[29] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. 2020. Learning model-agnostic counterfactual explanations for tabular data. In Proceedings of the web conference 2020 3126 3132.

[30] Martin Pawelczyk, Himabindu Lakkaraju, and Seth Neel. 2023. On the privacy risks of algorithmic recourse. In International Conference on Arti cial Intelligence and Statistics PMLR, 9680 9696.

[31] Judea Pearl. 2009. Causal inference in statistics: An overview. Statistics surveys 3 (2009).

[32] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. Elements of causal inference: foundations and learning algorithms. The MIT Press.

[33] John Platt and Alan Barr. 1987. Constrained di erential optimization. Neural Information Processing Systems

[34] Miriam Rateike, Ayan Majumdar, Olga Mineeva, Krishna P. Gummadi, and Isabel Valera. 2022. Don't Throw it Away! The Utility of Unlabeled Data in Fair Decision Making. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (Seoul, Republic of Korea) (FAccT '22). Association for Computing Machinery, New York, NY, USA, 1421 1433. https://doi.org/10.1145/3531146.3533199

[35] Kaivalya Rawal, Ece Kamar, and Himabindu Lakkaraju. 2020. Algorithmic recourse in the wild: Understanding the impact of data and model shifts. arXiv preprint arXiv:2012.11788 (2020).

[36] Cristóbal Romero and Sebastián Ventura. 2011. Preface to the special issue on data mining for personalised educational systems. User Modeling and User Adapted Interaction 21, 1 (2011), 1.

[37] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. 2004. Are loss functions all the same? Neural computation 16, 5 (2004), 1063 1076.

[38] Pedro Sanchez and Sotirios A Tsaftaris. 2022. Di usion Causal Models for Counterfactual Estimation. In Causal Learning and Reasoning 2022

[39] Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. 2022. VACA: Designing Variational Graph Autoencoders for Causal Queries. In Proceedings of the AAAI Conference on Arti cial Intelligence Vol. 36. 8159 8168.

[40] Kumba Sennaar. 2019. Machine Learning for Recruiting and Hiring 6 Current Applications. En línea]. Disponible en: https://emerj. com/ai-sector-overviews/machine-learning-forrecruiting-and-hiring/.[Último acceso: 30 Mayo 2020] (2019).

[41] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2020. CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (New York, NY, USA) (AIES '20). Association for Computing Machinery, New York, NY, USA, 166 172. https://doi.org/10.1145/3375627.3375812

[42] Kacper Sokol and Peter A Flach. 2018. Conversational Explanations of Machine Learning Predictions Through Class-contrastive Counterfactual Statements.. In IJCAI 5785 5786.

[43] Emily Sullivan and Philippe Verreault-Julien. 2022. From Explanation to Recommendation: Ethical Standards for Algorithmic Recourse. In Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society 712 722.

[44] Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. 2021. Towards robust and reliable algorithmic recourse. Advances in Neural Information Processing Systems 34 (2021), 16926 16937.

[45] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classi cation. In Proceedings of the conference on fairness, accountability, and transparency 10 19.

[46] Arnaud Van Looveren and Janis Klaise. 2021. Interpretable counterfactual explanations guided by prototypes. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases Springer, 650 665.

[47] Suresh Venkatasubramanian and Mark Alfano. 2020. The philosophical basis of algorithmic recourse. In Proceedings of the 2020 conference on fairness, accountability, and transparency 284 293.

[48] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E Hines, John P Dickerson, and Chirag Shah. 2020. Counterfactual explanations and algorithmic recourses for machine learning: A review. arXiv preprint arXiv:2010.10596 (2020).

[49] Sahil Verma, Keegan Hines, and John P Dickerson. 2022. Amortized generation of sequential algorithmic recourses for black-box models. In Proceedings of the AAAI Conference on Arti cial Intelligence Vol. 36. 8512 8519.

[50] Warren J von Eschenbach. 2021. Transparency and the black box problem: Why we do not trust AI. Philosophy & Technology 34, 4 (2021), 1607 1622.

[51] Julius von Kügelgen, Amir-Hossein Karimi, Umang Bhatt, Isabel Valera, Adrian Weller, and Bernhard Schölkopf. 2022. On the fairness of causal algorithmic recourse. In Proceedings of the AAAI Conference on Arti cial Intelligence Vol. 36. 9584 9594.

[52] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harv. JL & Tech. 31 (2017), 841.

[53] Austin Waters and Risto Miikkulainen. 2014. Grade: Machine learning support for graduate admissions. Ai Magazine 35, 1 (2014), 64 64.

[54] Adrian Weller. 2017. Challenges for Transparency. CoRR abs/1708.01870 (2017). arXiv:1708.01870 http://arxiv.org/abs/1708.01870

[55] Matej Ze£evi¢, Devendra Singh Dhami, Petar Veli£kovi¢, and Kristian Kersting. 2021. Relating graph neural networks to structural causal models. arXiv preprint arXiv:2109.04173 (2021).

A DATASETS

We list the di erent datasets used in the evaluations, their causal graphs in Fig. 6, and the detailed structural equations or functions. We also detail the dataset generation processes used for training the di erent models.

Data sample for training downstream algorithmic classifiers. We generate 20,000 samples for each dataset and split them into train-validation-test as 0.5, 0.25, 0.25.

Data sample for training causal generative models. We generate 25,000 samples for each dataset and split them into train-validation-test as 0.8-0.1-0.1. These datasets are then used to train the causal generative models (causal ows or VACA) used by CARMA.

Data sample for running recourse. We separately generate 20,000 samples for each dataset and split them into train-validation-test as 0.5-0.25-0.25. Then, we utilize the downstream classi er algorithm to lter the datasets to include only the negatively predicted individuals. This ltered dataset is used to train the recourse models CARMA and DiCE-VAE. The oracle SCM directly solves for recourse on the test dataset.

A.1 Synthetic datasets

We created simple synthetic datasets inspired by [15, 39]. These datasets have 4 variables, where one variable is a binary non-actionable feature. The other features x_1, x_2, x_3 are continuous variables. The causal graphs are in Fig. 6a, 6b, 6c, showing di erent causal relations between the features. The nature of the equations may be linear (LIN) or non-linear (NLIN). The symbol σ represents the sigmoid function in the following equations. Moreover, $\mathcal{N}(0, 1)$ denotes the Gaussian distribution where 0 is the mean and 1 is the variance of the distribution, $\sqrt{1}$ denoting the standard deviation.

A.1.1 Synthetic triangle-LIN.

$$
\begin{aligned}
x_0 &:= f(u_0; \theta_0) \quad u_0 \sim \text{Bernoulli}(0.5) \\
x_1 &:= f(u_1; \theta_1) \quad u_1 \sim \text{MoG}(0.5, 2.5, 0.5, -2.5, 1.0) \\
x_2 &:= f(x_1, u_2; \theta_2) \quad u_2 \sim \mathcal{N}(0, 1.0) \\
x_3 &:= f(x_1, x_2, u_3; \theta_3) \quad u_3 \sim \mathcal{N}(0, 1.0) \\
y &:= \mathbb{I}(f(-x_1, 0.5x_2, -x_3, 0.5x_0) > 0.5)
\end{aligned}
$$

A.1.2 Synthetic triangle-NLIN.

$$
\begin{aligned}
x_0 &:= f(u_0; \theta_0) \quad u_0 \sim \text{Bernoulli}(0.5) \\
x_1 &:= f(u_1; \theta_1) \quad u_1 \sim \text{MoG}(0.5, 2.5, 0.5, -2.5, 1.0) \\
x_2 &:= f(1, 3\sin(-x_1), u_2; \theta_2) \quad u_2 \sim \mathcal{N}(0, 0.1) \\
x_3 &:= f(x_1, x_2, u_3; \theta_3) \quad u_3 \sim \mathcal{N}(0, 1.0) \\
y &:= \mathbb{I}(f(1, 2^{-x_1}, 0.75\exp(-x_2^3), 0.1x_3^2, 0.5x_0) > 0.5)
\end{aligned}
$$

A.1.3 Synthetic collider-LIN.

$$
\begin{aligned}
x_0 &:= f(u_0; \theta_0) \quad u_0 \sim \text{Bernoulli}(0.5) \\
x_1 &:= f(u_1; \theta_1) \quad u_1 \sim \text{MoG}(0.5, 2.5, 0.5, -2.5, 1.0) \\
x_2 &:= f(u_2; \theta_2) \quad u_2 \sim \mathcal{N}(0, 1.0) \\
x_3 &:= f(0.5x_1, 0.25x_2, u_3; \theta_3) \quad u_3 \sim \mathcal{N}(0, 1.0) \\
y &:= \mathbb{I}(f(x_1, 0.5x_2, -x_3, 0.5x_0) > 0.5)
\end{aligned}
$$

A.1.4 Synthetic collider-NLIN.

$$
\begin{aligned}
x_0 &:= f(u_0; \theta_0) \quad u_0 \sim \text{Bernoulli}(0.5) \\
x_1 &:= f(u_1; \theta_1) \quad u_1 \sim \text{MoG}(0.5, 2.5, 0.5, -2.5, 1.0) \\
x_2 &:= f(0.5x_1, u_2; \theta_2) \quad u_2 \sim \mathcal{N}(0, 0.1) \\
x_3 &:= f(0.25x_1^2, 0.25x_2^2, u_3; \theta_3) \quad u_3 \sim \mathcal{N}(0, 1.0) \\
y &:= \mathbb{I}(f(1, x_1^3, -x_2, 3x_3, 0.5x_0) > 0.5)
\end{aligned}
$$

A.1.5 Synthetic chain-LIN.

$$
\begin{aligned}
x_0 &:= f(u_0; \theta_0) \quad u_0 \sim \text{Bernoulli}(0.5) \\
x_1 &:= f(u_1; \theta_1) \quad u_1 \sim \text{MoG}(0.5, 2.5, 0.5, -2.5, 1.0) \\
x_2 &:= f(0.5x_1, u_2; \theta_2) \quad u_2 \sim \mathcal{N}(0, 1.0) \\
x_3 &:= f(0.25x_2, u_3; \theta_3) \quad u_3 \sim \mathcal{N}(0, 1.0) \\
y &:= \mathbb{I}(f(-x_1, 0.5x_2, -x_3, 0.5x_0) > 0.5)
\end{aligned}
$$

A.1.6 Synthetic chain-NLIN.

$$
\begin{aligned}
x_0 &:= f(u_0; \theta_0) \quad u_0 \sim \text{Bernoulli}(0.5) \\
x_1 &:= f(u_1; \theta_1) \quad u_1 \sim \text{MoG}(0.5, 2.5, 0.5, -2.5, 1.0) \\
x_2 &:= f(x_1, u_2; \theta_2) \quad u_2 \sim \mathcal{N}(0, 0.1) \\
x_3 &:= f(\tfrac{2}{x_2}, u_3; \theta_3) \quad u_3 \sim \mathcal{N}(0, 1.0) \\
y &:= \mathbb{I}(f(\tfrac{3}{x_1}, -x_2, 0.75\exp(-x_3^3), 0.5x_0) > 0.5)
\end{aligned}
$$

A.2 Semi-synthetic dataset: Loan

Following [19], we consider the 7-variable Loan dataset modeling the Credit dataset [12], shown in Fig. 6d. Similar to [19], the equations are non-linear with non-additive exogenous variables. Using the usage of [19], Gender and Age are non-actionable, loan amount, education-level, income, and savings are actionable. At the same time, loan duration is non-actionable but mutable (owing to the downstream impact of interventions on the parents of). The equations are as follows. The symbol σ represents the sigmoid function in the following equations. Moreover, $\mathcal{N}(0, 1)$ denotes the Gaussian distribution where 0 is the mean and 1 is the variance of the distribution, $\sqrt{1}$ denoting the standard deviation. Finally, Gamma$(0.1, 1.0)$ denotes the Gamma distribution where 0 is the concentration/shape and 1 is the scale, 1.1 being the rate (used in PyTorch's distribution de nitions).

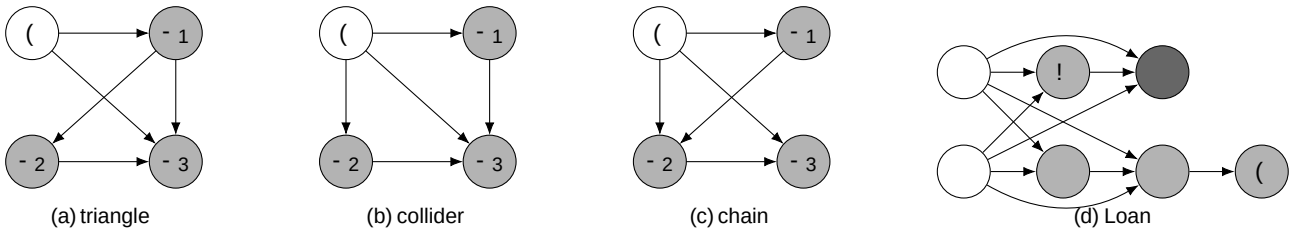

Figure 6: Causal graphs for synthetic and the semi-synthetic Loan datasets. Non-actionable features are , actionable features are , and non-actionable but mutable features are .

5 : = * ; * Bernoulli¹0”5º
5 : = 35¸ * ; * Gamma¹10•3”5º
5 : = 0”5¸ f ¹ 1¸ 0”5 ¸ f ¹0”1 º¸ * º; * N¹ 0•0”25º
5 : ! = 1¸ 0”01¹ 5º¹5 º¸ ¹ 1 º¸ *! ; *! N¹ 0•4º
5 : = 1¸ 0”1 ¸ 2¹1 º¸ ! ¸ * ; * N¹ 0•9º
5 : = 4¸ 0”1¹ ¸ 35º¸ 2 ¸ ¸ * ; * N¹ 0•4º
5 : (= 4¸ 1”5If ¡ 0g¸ *(; *(N¹ 0•25º
5 : . = If f ¹0”3¹ ! ¸ ¸ (¸ U (ºº 0”5g

where U = 1 if »If ¡ 0g ^ If(¡ 0g¼ = 1; U = 1 otherwise.

B RECOURSE SOLVER BASELINES

B.1 Causal Oracle

The causal oracle recourse solver M is based on the methodology used in [19, 51]. This methodology assumes that we have access to the causal knowledge, i.e., the exact nature of the causal relations between the data features. Access to this knowledge allows the oracle M to exactly compute causal counterfactuals with the abduction-action-prediction steps [31]. However, M needs to solve the complex combinatorial optimization problem behind causal recourse for each individual separately. To do so, M discretizes the feature space and applies grid-based search over this space to find the interventions that lead to minimal cost and ensure successful recourse. For the synthetic datasets (triangle, collider, and chain), we discretize using 25 equally-spaced bins. For the more complex Loan data, we discretize using 15 equally-spaced bins. The bins are in the range »x$_8^F$ 2¹x$_8^F$ min¹x^{F}ºº•x$_8^F$¸ 2¹max¹x^{F}º x$_8^F$º¼ for each individual 8 with features x^F. The search algorithm then reviews the bins to analyze all possible intervention combinations for optimal cost and coverage.

B.2 Non-causal amortization using DiCE-VAE

Assuming binary prediction labels, DiCE-VAE generates x^{CF} given x^F. It uses the VAE formulation to encode x^F to a latent z through &¹ º, then decode to x^{CF} such that ¹x^{CF}º = 1. It aims to change all the features X without causal information. Note that the counterfactuals x^{CF} are not causal counterfactuals . Specifically, they are not computed as interventions on a causal SCM M .

The corresponding evidence lower-bound (ELBO), following the VAE construction, is:

$$\ln \Pr\left(x^{CF} \mid x^F\right) \geq E_{\&\left(z \mid x^F\right)}\left[\ln \%\left(x^{CF} \mid z, x^F\right) + \ln \&(z \mid x^F) - \ln \%(z)\right] \quad (7)$$

Then, [24] solves for counterfactual explanations using the VAEs as:

$$E_{\&\left(z \mid x^F\right)}\left[\mathrm{Dist}\left(x^F, x^{CF}\right) + \_ \cdot \mathrm{HingeLoss}\left(x^{CF}, 1, V\right)\right] - \ln \%(z) \quad (8)$$

The distance function Dist is ℓ_2 norm (we use ℓ_2 for a fair comparison to CARMA), and V denotes the hinge margin. A higher V value leads to better coverage over the entire population but generates more conservative explanations with higher costs.

Enhancing DiCE-VAE with feature actionability. By default, the existing DiCE-VAE formulation does not support non-actionability of features. So, it simply assumes all input features as actionable. To allow for non-actionability of certain features, we input an actionability mask M . Then, the ELBO used to generate x^{CF}.

$$\ln \Pr\left(x^{CF} \mid x^F, M\right) \geq E_{\&\left(z \mid x^F\right)}\left[\ln \%\left(x^{CF} \mid z, x^F, M\right) - \ln \&(z \mid x^F) + \ln \%(z)\right] \quad (9)$$

The mask M is used by the VAE decoder to generate x^{CF} abiding by the non-actionability requirements, setting the change for a feature to 0 if the feature is non-actionable.

C COUNTERFACTUAL APPROXIMATION USING VACA

While the causal normalizing flows [15] trained a single autoregressive normalizing flows model f_k to approximate both causal abduction (mapping X^F to Z^F) and causal prediction (mapping Z^F to X^F), the Graph-NN (GNN) based VACA [39] utilized a different modeling. VACA approximated causal abduction using two models: an encoder $f_{k_4}^{enc}$ that uses GNNs to map x^F to latent Z^F and a decoder $f_{k_3}^{dec}$ that uses GNNs to map latent Z^F to features X^F.

Specifically, VACA assumes a distribution of latent factors Z where we have one Z_i for each feature X_i. Additionally, while causal flows can model the causal graph ordering using TMI maps, VACA needs ϕ^{enc} and ϕ^{dec} also to be provided with the adjacency matrix A for the causal graph G. As shown in [39], A is a $d \times d$ binary matrix (where d denotes the total number of features in the data) with $A_{ij} = 1$ when feature X_j is a parent of X_i or $i = j$. Otherwise, the value is 0.

While VACA can utilize GNNs in the encoder-decoder setup to model the causal data distribution, it cannot satisfy causal consistency like the causal flows. Specifically, as discussed in the original paper [39], the latent factors Z of VACA do not need to correspond to the exogenous U, i.e., $P(U) < p(Z)$. Similarly, the VACA decoder does not aim to approximate the causal structural functions. VACA can only ensure each latent Z_i to capture the information in feature X_i not described by its causal parents.

Nonetheless, Sánchez-Martin et al. [39] showed how VACA, under certain conditions, can be leveraged to compute causal counterfactual features. As discussed in [15], VACA's interventions are similar to the traditional method of Pearl [31]. Hence, to compute counterfactuals, we perform interventions by severing the edges in every GNN layer whose endpoints fall in the path that generates a particular intervened feature. This technique ensures that the causal ancestors of the intervened feature do not influence the downstream computations. Particularly, given factual latent factors Z^F for observed features X^F, intervened latent factors Z^I after we intervene on certain features $\mathcal{I}$, the VACA decoder can use the A^I, i.e., the adjacency matrix of the post-intervention causal graph to compute counterfactuals as:

$$x^{CF} \sim \phi^{dec}_{k_3}(X^{CF} \mid Z^F, Z^I, A^I)$$

The causal interventional *do* operator is approximated by generating the interventional A^I for the interventions on the features $\mathcal{I}$ from the original A, where

$$A^I = \{A_{ij}\}_{i \notin \mathcal{I}, j} \cup \{A_{ij} = 0\}_{i \in \mathcal{I}, j}.$$

Hence, computing counterfactuals using VACA needs *combinatorial modification* of the adjacency matrix that increases the complexity and the computation times as seen in § 5.4.

Unfortunately, as shown by Javaloy et al. [15], modifying the adjacency matrix for interventions is *not enough* to ensure accurate counterfactuals. Specifically, severing edges in the VACA decoder may not be faithful in counterfactual estimation since it might require *additional recalibration*. This reason might explain the higher costs reported by CARMA when using VACA compared to causal flows. We refer the reader to Appendix C.2 in [15] and § 4 in [39] for more detailed descriptions.

D WEIGHTED COST CAUSAL RECOURSE FOR POPULATION FEATURE PREFERENCES

We can incorporate the notion of difficulty to change particular features in the amortized causal recourse framework. In our case, we show the simplistic method of incorporating it using different weights in the cost function for the different features. We do this incorporation in two steps. First, we incorporate the weights in the ℓ_2 cost function, where we multiply the cost of feature change with the weight, then add across the features.

$$\text{wcost}_i = \frac{1}{K}\sum_{j=1}^{K} w_j \,(x_j^F - x_j^{CF})^2,$$

We set the weights inspired by the notion of *time* from [2]. Specifically, we set the weight higher for features higher up in the causal ordering in the causal graph. As per [2], deeper nodes in the causal graph might be more challenging to intervene on since these interventions require more time to achieve. For example, the weights in the Loan dataset are for Savings 1, Loan-Amount and Income 2, and Education 3. Note that the weights are precisely equal to the depth of each feature's node in the causal graph.

Secondly, we use the weights to set the prior probabilities in the mask network loss function. This helps more explicitly control the likelihood of selecting features for interventions. In Eq. 5, the prior probability π of Bernoulli(π) is set according to the weights. We keep the same prior $\pi = 0.5$ for the feature with the least cost weight of 1. For any other feature, the prior is set according to the weight w_i as $\pi = 0.5/w_i$. Hence, for the Loan data, Savings has $\pi = 0.5$, Income and Loan amount have $\pi = 0.25$, and Education has $\pi = 1/6$.

E TRAINING DETAILS

E.1 Downstream algorithmic classifiers

For the fixed, downstream algorithmic binary classifiers, we fix the architectures as follows:

- For synthetic datasets with LIN causal relations, we train logistic regression classifiers.
- For synthetic datasets with NLIN causal relations, we train neural network classifiers with a hidden dimension of 16 and ReLU activation.
- For the larger Loan dataset, we train a neural network with hidden layers 32×32 and ReLU activation.

All the models are trained for 500 epochs with batch size 256, using the Adam optimizer with a learning rate of 10^{-3}. We use PyTorch to train all the classifiers for easy gradient access to the trained models.

E.2 Causal generative models

We use the newest two state-of-the-art causal estimators: i) causal normalizing flows [15] shown to perform the best in terms of distributional approximation and counterfactual inferences, and ii) GNN-based VACA [39].

Causal normalizing flows. For the causal normalizing flows [15], we use the original work as inspiration and tune the inner dimensionality of the flow NN models. All other parameters are fixed as follows: optimizer was Adam with learning rate 10^{-3} and plateau scheduler, activation ELU, layer type was Masked Autoregressive Flows (MAF). For other parameters, we refer to the original work [15] and GitHub code[5].

[5] www.github.com/psanch21/causal-flows/

Table 2: Hyperparameters for Causal Flows. The parameters were selected after runs on ve di erent seeds. The log-prob of the observational data distribution estimate was used to select the best hyperparameters.

Dataset	dim-inner	log-prob
triangle-LIN	32	-5.71
collider-LIN	32	-5.7
chain-LIN	64	-5.7
triangle-NLIN	32 32 32	-4.56
collider-NLIN	16 16 16	-4.56
chain-NLIN	32 32 32	-5.71
Loan	64	-13.6

VACA. For the VACA causal generative model based on GNNs, the number of layers-pre of the GNN, the model dropout, and the type of GNN layer (gin , pna, pna-disjoint) were tuned [15]. Based on [15, 39], the following attributes were xed: latent dimension for I of each feature was 4, activation was ReLU, optimizer was Adam with learning rate 5 10 ^{3}, inner dimension of GNN was 16. Other parameters can be referred from [15] and the corresponding GitHub code[5]. Following [15], we used the log-prob (log probability) of the approximation of the observed data distribution to select the best hyperparameters.

Table 3: Hyperparameters for VACA. The parameters were selected after runs on ve di erent seeds. The log-prob of the observational data distribution estimate was used to select the best hyperparameters.

Dataset	GNN layers-pre	dropout	layer	log-prob
Loan	1	0.1	gin	-18.65

From the Tables 2 and 3, it is clear that causal ows can t the observational data distribution better since the log-prob values are always higher than VACA. Remarkably, causal ows achieve better performance despite requiring signi cantly less hyperparameter tuning.

E.3 Recourse solvers

We use the Optuna library [1] for Bayesian hyperparameter tuning. All hidden layer activations are xed to ReLU, and the optimizer is always Adam. We utilize the Modi ed Di erential Method of Multipliers (MDMM) [33] to work with the multi-objective optimization between the recourse cost and recourse coverage. We use the default damping coe cient value of 10 to balance the objective of recourse cost and positive prediction (hinge loss). In our experience, this parameter's value was stable and did not signi cantly in uence the outcome. This also reduces a hyperparameter and allows the learning process to optimize for recourse cost and positive prediction automatically.

E.3.1 Oracle. For the causal oracle M , as mentioned before, we use a simple grid-based search to nd the best intervention. This search is possible since we assume complete access to the SCM Furthermore, this technique has been used in prior work [18, 51]. For the synthetic datasets with fewer features, we use 25 equi-spaced bins. For the larger Loan data, we use 15 equi-spaced bins.

E.3.2 DiCE-VAE. For the DiCE-VAE [24] model, we tune the learning rate, number of layers and neurons in the neural networks, epochs, batch size, and the hinge-margin for the coverage loss as in Eq. 3. As mentioned before, we modify the original model to incorporate a mask to enable DiCE-VAE to satisfy actionability requirements on the di erent features. The VAE latent space always set to the same dimensionality as the number of observed features in the corresponding dataset.

E.3.3 CARMA. We tune the mask selector's learning rate, layers, neurons, action predictor networks, epochs, batch size, hinge margin, and the g parameter for the mask network's loss.

Using causal ows. When using causal ows as the causal generative model, the action-predictor model concatenates all the latent I (exactly corresponding to the causal [2]), then concatenates them with the binary mask selector's estimated intervention mask. The output dimensionality of the action predictor is simply the number of features, and each dimension corresponds to the for feature X_i.

Using VACA. When using VACA, the latent representations of the causal generative model are more complex. Speci cally, for each feature, the I dimension is 4, which increases the total dimensionality over all the features. As a result, in our experiments, we saw that we needed a more complex action-predictor architecture to estimate the causal interventions when using VACA e ectively. The action-predictor model has separate embedding and de-embedding modules for pre-trained VACA representations. The embedding module is a single-layer neural network with ReLU activation. This module concatenates the 4-dimensional for each feature X_i and the corresponding mask-selector's estimated binary mask dimension I_i into a hidden representation. Correspondingly, when the action-predictor outputs I_i, the de-embedding module maps the action-predictor's hidden feature to the dimensionality of X_i through a 2-layer neural network with activation ReLU.

E.4 Computing Hardware

All models were trained using the CPU version of PyTorch 1.13.1 without multiprocessing. The programs, including training and hyperparameter tuning, were run on a cluster machine with two Intel Xeon Gold 6134M processors with 3.2GHz clock speed, up to 768GB of available RAM, and running Linux OS. However, the training and inference times are very similar even when we run our models on a personal computing system, e.g., an Apple M1 MacBook Pro with 16GB RAM.

F ADDITIONAL ANALYSIS

We show analyses for the following aspects.

(R5): How does CARMA's amortized recourse target di erent features compared to unamortized recourse?

(R6): How does CARMA's deployment performance vary with the amount of data available for training?

Table 4: Hyperparameters for DiCE-VAE model that has been extended to include actionability capabilities. We use the cost and the coverage to select the best validation runs in Optuna tuning. The hyperparameters are selected based on runs using ve di erent seeds.

Dataset	learning rate	architecture	epochs	batch-size	hinge-margin
triangle-LIN	5 10^{5}	64	250	128	0.003
collider-LIN	5 10^{5}	64	250	128	0.003
chain-LIN	5 10^{5}	64	250	128	0.003
triangle-NLIN	5 10^{5}	64	250	128	0.042
collider-NLIN	5 10^{5}	64	250	128	0.003
chain-NLIN	5 10^{5}	64	250	128	0.012
Loan	5 10^{5}	64	200	128	0.002

Table 5: Hyperparameters for CARMA using pre-trained causal normalizing ows as the causal generative models for causal estimation. We use the cost and the coverage to select the best validation runs in Optuna tuning. The hyperparameters are selected based on runs using ve di erent seeds.

Dataset	learning rate	mask-selector	action-predictor	epochs	batch-size	hinge-margin	Gumbel $_{g}$
triangle-LIN	5 10^{3}	32	8 8 8 8	500	32	0.007	0.55
collider-LIN	5 10^{3}	32	32 32 32 32	350	256	0.013	0.31
chain-LIN	5 10^{3}	16	8 8	150	128	0.004	0.87
triangle-NLIN	5 10^{3}	32 32 32	8 8 8 8	450	256	0.003	0.2
collider-NLIN	10^{2}	16 16 16	8 8 8	250	64	0.002	0.83
chain-NLIN	5 10^{3}	32 32 32	32 32 32 32	250	256	0.007	0.08
Loan	5 10^{3}	32 32	32 32 32	450	256	0.011	0.45
Loan(weighted case study)	10^{3}	16 16 16	32 32 32 32	450	256	0.007	0.33

Table 6: Hyperparameters for CARMA using pre-trained VACA as the causal generative models for causal estimation. We use the cost and the coverage to select the best validation runs in Optuna tuning. The hyperparameters are selected based on runs using ve di erent seeds.

Dataset	learning rate	mask-selector	action-predictor	epochs	batch-size	hinge-margin	Gumbel $_{g}$
Loan	5 10^{4}	64 64 64	64 64 64 64	450	256	0.003	0.4

(R7): How does CARMA's amortized recourse target di erent features in the smaller synthetic datasets?

(R8): How does CARMA's amortized recourse with preference weights compare to the unamortized M ?

F.1 How does causal recourse target di erent features?

Considering the Loan dataset, this section further explores the nature of the causal recourse interventions, contrasting the amortized interventions of CARMA to the optimal unamortized M . We plot our empirical observations regarding this analysis in Fig. 7. In Fig. 7 (left), we plot the percentage of recourse interventions on the unseen test dataset (y-axis) that intervened on a particular number of actionable features (x-axis). In Fig. 7 (right), we plot the percentage of test-set recourse interventions (y-axis) that targeted each actionable feature (x-axis). In both gures, we show the values of CARMA and M using bars with blue and orange colors, respectively. Note for Fig. 7 that since an intervention can target multiple features, the bar heights for a particular method will not sum to 100%.

Fig. 7 (left) shows that CARMA intervenes mainly on three features while M acts on one or two features. This result con rms the increased sparsity of M compared to CARMA we observed in Ÿ 5.2. Regarding the recourse interventions, the di erence is also observed in Fig. 7 (right) for the targeted features. M , being an unamortized method with perfect causal knowledge, seldom intervenes on Education (around 5% of test-time interventions), instead mainly acting on Loan amount (around 60%) or Income (around 80%). In contrast, for each of the actionable features in the Loan data, the percentage of test-time CARMA interventions targeting them remains around 40 to 55%. This increased percentage across all features for CARMA shows why it cannot achieve the perfect optimality of M in terms of sparsity and cost. This indicates that the

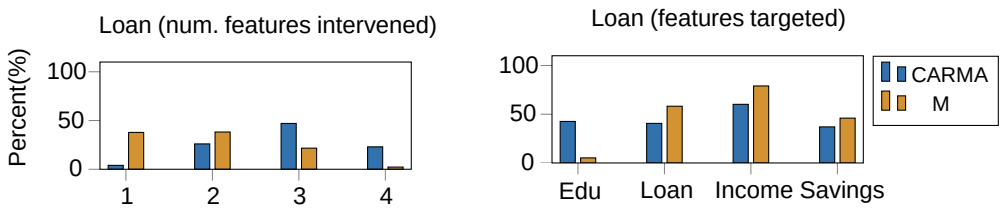

Figure 7: Bar graphs showing (left) the percentage of times recourse interventions changed a particular number of features and (right) the percentage of test-time interventions targeting each feature of the Loan data by recourse methods.

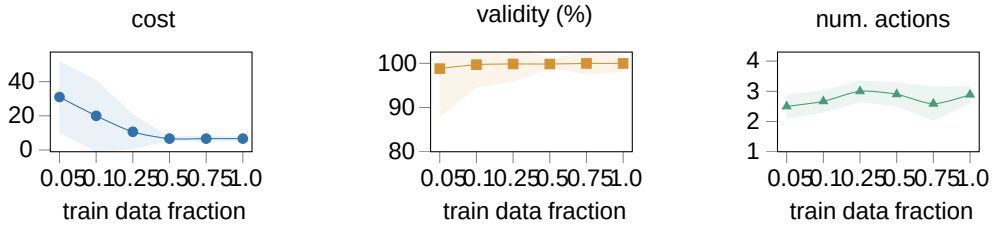

Figure 8: Variation in CARMA 's $_2$ recourse cost, causal validity (%), and mean number of intervened features on Loan across 10 seeds with training data size (di erent fractions of original data).

amortization process of CARMA for causal recourse optimization might tradeo some of these desiderata to ensure high recourse validity.

F.2 How does reducing the amount of training data a ect CARMA?

Determining the amount of data necessary for training CARMA that can o er optimal solutions on deployment across large populations is crucial. Hence, we study how CARMA's performance regarding di erent recourse metrics can vary if we have di erent amounts of accessible training data. For this study, we x the model hyperparameters that we obtained when we used the full Loandata and assume access to the same pre-trained causal ows model that we used in our analyses in the main paper. Then, we vary the fraction of the training data size in 0"05 • 0"1 • 0"25 • 0"5 • 0"75 and train CARMA to report the test-time performance metrics. We show the visualizations in Fig. 8, where we vary the training data fraction along the x-axis for all subplots. Along the y-axis, we report the recourse $_2$ cost (left), the causal validity (in %, center), and the average number of actions or features targeted (right). We also show the variance region with shaded color after running for 10 di erent seeds. Analyzing recourse cost in Fig. 8 (left), reducing the data fraction from the full 1.0 up to 0.5 does not impact the performance. However, if we keep reducing it more, the cost increases. As expected, the cost peaks at extremely low amounts of training data of 0.05 or 0.1 fractions (around 225 or 450 data points). Along with increased cost, we also see the variance in the metric increase across random seeds when we use very low data sizes. We see a similar trend in variance for recourse causal validity in Fig. 8 (center), with the variance increasing for fractions 0"25. The mean value of the validity metric remains high, roughly around 99.99%. However, note that with varying the training data fractions, the average number of actions (features targeted for recourse) shows no clear trend, remaining roughly similar, around 2.6 to 2.9.

In summary, the e cacy of deploying CARMA hinges on the volume of available training data. Inadequate data may lead to suboptimal performance characterized by high costs and increased variance. Optimal performance is achieved with su cient data, such as fractions 0"5 or 2245 data points for Loan It is essential, however, to assess data su ciency for optimal amortized recourse deployment.

F.3 Visualizing the nature of feature interventions for synthetic data

For synthetic data, the features targeted depend on the data. The non-causal DiCE-VAE intervenes on all actionable features. The oracle M provides the most sparse interventions, but the targeted features depend on the dataset's causal graph and the features' importance to the label . CARMA interventions are not as sparse as M , acting on more features on average. However, the features CARMA target most in each dataset align with the most intervened feature by the causal oracle M . For example, in triangle ! # , both CARMA and M maximally intervene on - $_1$. Similarly, while CARMA interventions are less sparse for chain #! # , the most intervened feature for both CARMA and M is - $_2$.

F.4 Comparing recourse with preference weights between CARMA and M

For the population feature preference evaluation performed in Ÿ 5.5, we also compared the recourse metrics after incorporating the feature weight preferences between the amortized recommendations of CARMA and the unamortized recommendations of the causal optimal oracle solver M . We show the results in Table 7. From the metrics in Table 7, we see that CARMA's amortization of the

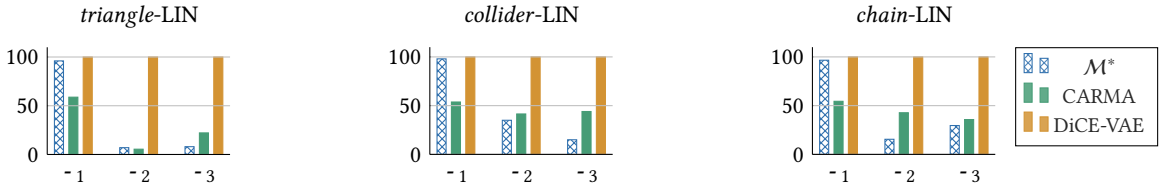

Figure 9: Distribution of feature targets of interventions for synthetic datasets with linear structural equations

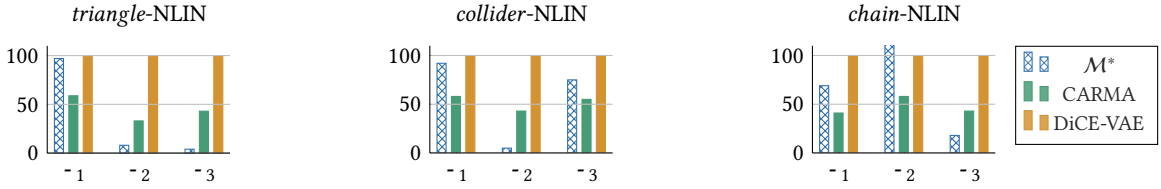

Figure 10: Distribution of feature targets of interventions for synthetic datasets with non-linear structural relations

Table 7: Comparing CARMA's amortized recourse performance on unseen test data (number of data shown in # test-data) to causal unamortized oracle $\mathcal{M}^*$ for *Loan* data with population-level feature preferences (10 seeds). Metrics include intervention cost ($_2$, *lower be er*), optimization coverage/success rate (%, higher better), mean intervened feature count, mean post-recourse prediction probability, compute time per datum (milliseconds, *lower be er*), and validity under true SCM (%, *higher be er*).

Dataset (features)	# test-data	Method	recourse $_2$-cost (×100)	optim. coverage (%)	mean num. actions (per datum)	mean prediction probability	time per datum (msec)	causal validity (%)
Loan (weighted) actionable: 3 total: 4	2364	$\mathcal{M}^*$	9.70	100.0	1.86	0.62	53987.80	100.0
		CARMA	$9''26_{\pm 1''12}$	$99''97_{\pm 0''05}$	$2''26_{\pm 0''4}$	$0''76_{\pm 0''03}$	$1''19_{\pm 0''1}$	$99''88_{\pm 3''5}$

causal recourse process is near-optimal compared to $\mathcal{M}^*$. This is highlighted by the similar low costs and lower compute times for CARMA. CARMA also achieves near-perfect causal validity. Similar to the results in the main paper, we see CARMA's amortization leads to recourse recommendations that are slightly more conservative with higher post-recourse prediction probabilities from the classifier. Similarly, the amortized recourse recommendations are less sparse than the optimal unamortized $\mathcal{M}^*$.