DICTUM: DIFFUSION COUNTERFACTUALS FOR TABULAR MIXED-TYPE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual explanations (CFs) provide actionable recourse by identifying minimal input changes that alter a model's prediction. Existing methods either optimize a single CF for a fixed criteria or generate multiple CFs from narrow distributions, limiting their ability to adapt to diverse user preferences at inference time. We propose DICTUM, a conditional diffusion framework for tabular data that models a broad distribution over the counterfactual space by training on opposite-class neighbors selected via a Determinantal Point Process (DPP). This diverse training signal, combined with a Transformer-based mixed-type diffusion backbone featuring per-column learnable noise schedules, enables DICTUM to generate counterfactual sets with substantially higher distributional coverage than prior generative approaches. At inference time, a branching generation strategy, inspired by SVDD-style guidance, enforces user-specified actionability constraints on the learned distribution, without retraining. Experiments on standard tabular benchmarks show that DICTUM achieves superior performance in proximity, plausibility, and diversity while being competitive in sparsity with the state-of-the-art. A constraint-strictness sweep and a component ablation further confirm that quality and plausibility are preserved as constraints tighten.

1 INTRODUCTION

Counterfactual explanations (CFs) have become a central tool in explainable artificial intelligence for providing actionable recourse to individuals affected by automated decisions (Wachter et al., 2017; Karimi et al., 2022). Given an instance assigned an unfavorable outcome, a CF specifies a minimal change to the input that would yield a desired prediction. For example, a rejected loan applicant might learn that increasing income or reducing debt would reverse the decision.

A single counterfactual explanation, however, rarely represents all useful forms of recourse (Mothilal et al., 2020; Laugel et al., 2018). Users differ in both their preferences and their circumstances: some may prioritize proximity to their current situation, others may prefer to change as few attributes as possible, and many face immutable or directionally constrained features. A diverse set of CFs exposes several recourse paths and allows a user to select one that is compatible with these requirements (Mothilal et al., 2020; Dandl et al., 2020).

However, existing approaches treat diversity primarily as a property of the generated set rather than as a structural property of the underlying generative distribution. Optimization-based methods spread solutions locally around a mode, while generative approaches trained on k-nearest opposite-class neighbors inherit the narrow support of their training targets. As a result, they fail to capture the full range of feasible recourse strategies and offer limited flexibility at inference time.

We argue that diversity should be addressed at the distributional level by learning a *broad conditional distribution* over the counterfactual space. Such a model can be queried post hoc under varying user requirements – most importantly, actionability constraints, without retraining.

We therefore propose DICTUM, a conditional diffusion for diverse CFs generation on mixed-type tabular data. Because ground-truth factual-counterfactual pairs are unavailable, it is trained on a diverse set of heuristically selected counterfactual examples, see Figure 1. For each factual input, greedy maximum a posteriori inference under a k-Determinantal Point Process (k-DPP) selects targets that balance proximity to the input with mutual dissimilarity (Kulesza & Taskar, 2012; Ca-

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

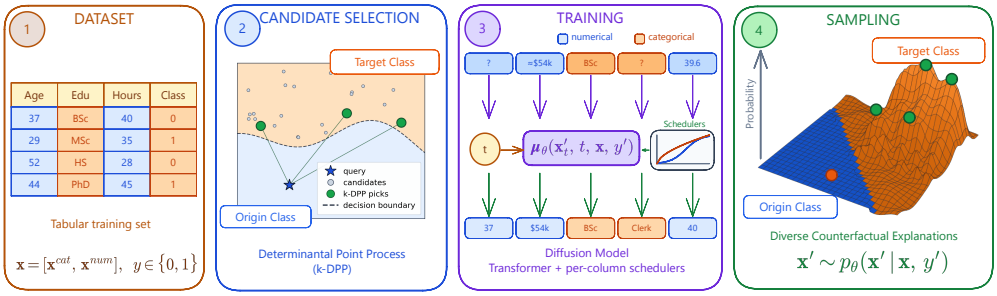


Figure 1: Overview of the DICTUM pipeline. (1) *Dataset*: a mixed-type tabular training set. (2) *Candidate selection*: for each query, a greedy k-DPP selects target-class anchors that are simultaneously close to the query and mutually diverse. (3) *Training*: a Transformer-based mixed-type diffusion backbone is trained on the query, anchor, target-class triplets. (4) *Sampling*: at inference time, the model produces a diverse set of counterfactual explanations from the broad learned distribution.

landriello et al., 2020). Compared with standard k-nearest-neighbor pairing, this procedure exposes the model to structurally different alternatives rather than a single local neighborhood.

To model the resulting conditional distribution, DICTUM uses a Transformer-based mixed-type diffusion backbone derived from TabDiff (Shi et al., 2025). Gaussian and masking-based diffusion processes model numerical and categorical features, respectively, while per-column learnable noise schedules adapt the corruption process to heterogeneous attributes. At inference time, a trajectory-branching strategy inspired by SVDD-style guidance (Li et al., 2024) generates and scores alternative continuations according to user-specified actionability constraints. This separates distribution learning from user-specific preferences and allows the same trained model to generate CFs for different constraint configurations.

Our contributions are as follows:

- We introduce DICTUM, a conditional diffusion framework trained on DPP-selected targets, yielding a counterfactual distribution with substantially broader coverage than prior methods.
- We develop a branching generation strategy, drawing inspiration from SVDD-style guidance (Li et al., 2024), that enforces user-specified actionability constraints on the learned distribution at inference time, without retraining.
- We conduct comprehensive experiments across five benchmark datasets where DICTUM achieves the best continuous proximity on all five datasets and the best diversity and plausibility on four of five, against DiCE (Mothilal et al., 2020), C-CHVAE (Pawelczyk et al., 2020), and DiCoFlex (Furman et al., 2025).

2 Related Work

Counterfactual explanations have emerged as one of the predominant paradigms in Explainable Artificial Intelligence (XAI). Existing approaches can be broadly categorized according to the properties they prioritize and the optimization procedures they employ. A substantial body of work formulates counterfactual generation as an optimization problem. Among these, gradient-based methods dominate, beginning with Wachter et al. (2017) and the contrastive formulation of Dhurandhar et al. (2018), and have been extended by approaches that add probabilistic plausibility constraints to the objective (Wielopolski et al., 2024; Mahajan et al., 2019; Van Looveren & Klaise, 2021) or enforce causal-graph-aware feasibility (Karimi et al., 2021). While gradient-based methods excel at producing valid counterfactuals and scale effectively to high-dimensional inputs, their applicability is inherently restricted to differentiable predictors such as neural networks. For non-differentiable models, mixed-integer programming (Ustun et al., 2019) offers an alternative formulation. We refer the reader to Guidotti (2024), Stepin et al. (2021) and Verma et al. (2024) for comprehensive surveys of the broader literature.

Diverse Counterfactual Explanations. A significant line of research focuses on generating diverse sets of counterfactual explanations, motivated by the observation that a single counterfactual rarely captures the full space of actionable alternatives available to a user. DiCE (Mothilal et al., 2020) is one of the most widely adopted methods in this direction, augmenting the standard counterfactual objective with a diversity term derived from determinantal point processes (DPPs) to encourage the jointly optimized instances to populate distinct regions of the input space. Other approaches handle diversity through alternative formulations: MACE (Yang et al., 2022) combines reinforcement-learning-based feature optimization with diversity-aware post-hoc selection, GeCo (Schleich et al., 2021) applies a custom genetic algorithm with feasibility filtering for tabular counterfactuals, and MOC (Dandl et al., 2020) casts counterfactual generation as a multi-objective optimization via NSGA-II. More recently, DiCoFlex (Furman et al., 2025) leverages a conditional normalizing flow trained to map factual instances to their counterfactual counterparts, enabling efficient sampling of an arbitrary number of diverse counterfactuals in a single forward pass without per-instance optimization. Our method also leverages DPPs, but in a fundamentally different role: whereas DiCE applies DPP kernels in the optimization objective to spread already-generated explanations apart, DICTUM applies DPPs during training-data curation to diversify the targets from which the generative model learns.

Plausible Counterfactual Explanations. A well-documented limitation of proximity-and-validity-only objectives is that they frequently produce counterfactuals lying in low-density, off-manifold regions that are neither realistic nor actionable. To address this, Artelt & Hammer (2020) introduced explicit plausibility constraints encouraging counterfactuals to reside in high-density regions of the input distribution, while FACE (Poyiadzi et al., 2020) constructs counterfactuals along feasible paths through the data manifold. A complementary line of work uses generative models that directly model the data distribution and therefore aim to produce on-manifold counterfactuals by construction. Within this line, methods such as C-CHVAE (Pawelczyk et al., 2020), REVISE (Joshi et al., 2019), CRUDS (Downs et al., 2020), and TabCF (Panagiotou et al., 2024) leverage variational autoencoders, while CeFlow (Duong et al., 2023) adopts normalizing flows, which offer the additional advantage of exact likelihood evaluation. More recently, diffusion-based tabular generators (Shi et al., 2025; Zhang et al., 2024; Kotelnikov et al., 2023; Lee et al., 2023) have surpassed VAE-based baselines on tabular synthesis benchmarks. Their potential for tabular counterfactual generation, however, remains largely unexplored.

In this work, we leverage diffusion models to generate counterfactual explanations for tabular data, combining diffusion-based density modeling with a DPP-driven training-data curation strategy. To our knowledge, DICTUM is the first method to address plausibility and diversity jointly at the data level for tabular counterfactuals, rather than as competing terms in an optimization objective.

3 DICTUM FOR DIVERSE COUNTERFACTUAL EXPLANATIONS GENERATION

3.1 PROBLEM STATEMENT AND MODEL OVERVIEW

A fundamental challenge in Counterfactual Explanation (CFE) is balancing the local proximity of explanations to the query instance with the coverage of diverse recourse paths necessary to satisfy heterogeneous user preferences. Existing approaches are often hindered by a trade-off: they either optimize for a single, locally optimal path or rely on generative frameworks that suffer from mode collapse, effectively restricting the distribution of counterfactuals to the nearest region of the target class.

We introduce DICTUM, a novel framework designed to learn a broad, conditional distribution $p(\mathbf{x}'|\mathbf{x}, y')$ over the counterfactual manifold. DICTUM leverages a conditional diffusion backbone to generate counterfactuals conditioned on the input $\mathbf{x}$ and target class y'. To address the absence of ground-truth counterfactual pairs, we construct training sets using a k-Determinantal Point Process (k-DPP). This approach selects diverse target counterfactuals for each training instance, significantly mitigating the mode-collapse limitations inherent in standard k-nearest neighbor (k-NN) retrieval. Furthermore, our Transformer-based mixed-type diffusion backbone enables unified modeling of numerical and categorical feature spaces.

This learned, broad generative distribution enables efficient post-hoc adaptation; users can extract counterfactuals satisfying specific requirements at inference time without requiring model retraining.

This is realized through diffusion guidance, which steers the generation process toward user-defined objectives (e.g. actionability constraints). Figure 1 summarises the resulting pipeline, from training-data curation to counterfactual sampling.

3.2 Diversity-Aware Training Data Acquisition via k-DPP

To learn a distribution that spans the feasible counterfactual space, we must expose the model to training targets that are not just "the closest ones," but representatives of different, valid recourse strategies. To this end, we employ greedy Maximum A Posteriori (MAP) inference on a k-Determinantal Point Process (k-DPP).

In the case of tabular data, k-DPP is realized using the Gower distance (Gower, 1971). This metric effectively aggregates discrepancies between two mixed-type vectors, $\mathbf{x}$ and $\mathbf{x}'$, by computing a weighted average of feature-wise distances:

$$d(\mathbf{x}, \mathbf{x}') = \frac{1}{D_{\text{num}}} \sum_{j \in \mathcal{J}_{\text{num}}} |x_j - x'_j| + w_{\text{gower}} \cdot \frac{1}{D_{\text{cat}}} \sum_{j \in \mathcal{J}_{\text{cat}}} \chi_{[x_j \neq x'_j]} \tag{1}$$

In this formulation, $\mathcal{J}_{\text{num}}$ represents the set of indices for numerical features with cardinality D_{num}, while $\mathcal{J}_{\text{cat}}$ denotes the set of categorical feature indices with cardinality D_{cat}. The hyperparameter w_{gower} is utilized to modulate the relative influence (or trade-off) between the numerical and categorical components of the distance calculation.

A key component of DICTUM is an efficient realization of k-Determinantal Point Process (k-DPP). Let us recall that k-DPP uses a kernel matrix $\mathbf{L}$ that explicitly balances proximity to the target and diversity among candidates. The kernel is defined as $L_{ij} = q_i q_j S_{ij}$, or in matrix form:

$$\mathbf{L} = \mathbf{S} \odot (\mathbf{q}\mathbf{q}^\top) \tag{2}$$

The quality vector $\mathbf{q}$ prioritizes candidates based on their proximity to the original reference point $\mathbf{x}$ while the similarity matrix $\mathbf{S}$ governs the redundancy between counterfactual candidates $\mathbf{x}'_i$ and $\mathbf{x}'_j$:

$$q_i = \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}'_i)}{\sigma_q}\right), \qquad S_{ij} = \exp(-d(\mathbf{x}'_i, \mathbf{x}'_j)/\sigma_s),$$

The parameters σ_q and σ_s are positive hyperparameter scales that determine the sensitivity of the kernel. By maximizing the determinant of submatrices within $\mathbf{L}$, the model simultaneously selects candidates with high relevance (large q_i) while penalizing redundant pairs through the off-diagonal similarity terms S_{ij}.

Exact k-DPP sampling typically involves an initial eigen decomposition of the kernel matrix $\mathbf{L}$, followed by a recursive selection process using elementary symmetric polynomials Kulesza & Taskar (2012). This approach incurs a computational complexity of $\mathcal{O}(N^3)$, where N is the number of candidates. Given that this process must be repeated for every training instance in each epoch, the overhead becomes prohibitive for large-scale datasets.

To alleviate this, we employ a greedy Maximum A Posteriori (MAP) approximation. This iterative approach constructs a diverse set Y of size k by maximizing the log-determinant gain at each step. Starting with an empty set $Y = \emptyset$, we sequentially add the candidate $i^\star$ that provides the largest marginal contribution:

$$i^\star = \arg\max_{i \notin Y} \left(\log\det(\mathbf{L}_{Y \cup \{i\}}) - \log\det(\mathbf{L}_Y)\right) \tag{3}$$

By the properties of determinants, this selection is equivalent to maximizing the conditional variance of the i-th candidate given the points already selected. In practice, this optimization is implemented efficiently using Cholesky factor updates. Specifically, the selection reduces to identifying the candidate with the largest diagonal entry in the Schur complement of $\mathbf{L}$ relative to Y.

After k iterations, this procedure yields a fixed-size subset that maintains a rigorous balance: candidates are selected if they are both individually relevant to the query (high q_i) and geometrically distant from one another in the feature space (low S_{ij}).

To construct the training dataset for estimating the conditional diffusion model $p(\mathbf{x}'|\mathbf{x}, y')$, we implement a systematic selection procedure for each instance $\mathbf{x}$ of class y in the original data. First, we define a candidate pool consisting of all training examples from the target class y'. We then construct the kernel matrix $\mathbf{L}$ as defined in equation 2, utilizing the Gower distance metric equation 1 and the formulation provided in Eq. equation 3. By applying the greedy MAP inference to this kernel, we extract a subset of k counterfactual candidates that are both proximal to $\mathbf{x}$ and mutually diverse. This process is repeated for every example in the dataset, resulting in a comprehensive training set where each observation is paired with k diverse representatives from the opposing class.

3.3 Transformer-Based Mixed-Type Diffusion Backbone

Modeling a conditional probability distribution $p(\mathbf{x}'|\mathbf{x}, y')$ is challenging for mixed-type tabular data. Building upon the recent work of Shi et al. (2025), we employ a conditional diffusion model that utilizes separate, *per-column* noise schedules $\sigma_j^{\text{num}}(t)$ for each numerical feature $j \in \{1, \dots, d_{\text{num}}\}$ and $\sigma_j^{\text{cat}}(t)$ for each categorical feature $j \in \{1, \dots, d_{\text{cat}}\}$, defined over $t \in [0, 1]$. This allows the model to treat categorical and numerical features differently, which is a critical aspect in tabular settings.

We adopt an EDM-style power-mean schedule $\sigma_j^{\text{num}}(t) = \left(\sigma_{\min}^{1/\rho_j} + t\,(\sigma_{\max}^{1/\rho_j} - \sigma_{\min}^{1/\rho_j})\right)^{\rho_j}$ for numerical features and a log-linear absorbing schedule $\sigma_j^{\text{cat}}(t) = -\log\big(1 - ((1 - \epsilon_{\min} - \epsilon_{\max})\, t^{k_j} + \epsilon_{\min})\big)$ for categorical ones, where $\sigma_{\min}$ and $\sigma_{\max}$ are the numerical noise levels at $t = 0$ and $t = 1$, while $\epsilon_{\min}$ and $\epsilon_{\max}$ set the initial masking probability and the final unmasked probability, respectively. The per-column shape parameters $\{\rho_j\}, \{k_j\}$ are trained jointly with θ, with a Gumbel-softmax reparameterisation on the categorical side.

The forward diffusion process progressively corrupts the data $\mathbf{x}'_0$. For numerical features, noise is injected as $x_{j,t}^{\text{num}} = x_{j,0}^{\text{num}} + \sigma_j^{\text{num}}(t)\,\epsilon_j$ with $\epsilon_j \sim \mathcal{N}(0, 1)$, while for categorical features, we employ a masking mechanism where categories are overwritten by a designated mask token $\mathbf{m}$ with probability proportional to $\alpha_{j,t} = \exp(-\sigma_j^{\text{cat}}(t))$ (we suppress j on α_t in the equations below). The reverse process learns the conditional denoising distribution $p_\theta(\mathbf{x}'_s|\mathbf{x}'_t, \mathbf{x}, y')$, which reconstructs the data given the original input $\mathbf{x}$ and target class y'. We represent this as the product of two distinct components:

$$p_\theta(\mathbf{x}'_s|\mathbf{x}'_t, \mathbf{x}, y') = p_\theta^{\text{num}}(\mathbf{x}_s^{\text{num}'}|\mathbf{x}'_t, \mathbf{x}, y') \cdot p_\theta^{\text{cat}}(\mathbf{x}_s^{\text{cat}'}|\mathbf{x}'_t, \mathbf{x}, y'). \tag{4}$$

The numerical component utilizes a denoising network $\boldsymbol{\mu}_\theta^{\text{num}}(\mathbf{x}_t, t, \mathbf{x}, y')$ to predict the noise, whereas the categorical component, conditioned on the mask state $\mathbf{m}$, predicts the categorical distribution over unmasked features:

$$p_\theta^{\text{cat}}(\mathbf{x}_s^{\text{cat}'}|\mathbf{x}'_t, \mathbf{x}, y') = \text{cat}\left(\frac{(1 - \alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\boldsymbol{\mu}_\theta^{\text{cat}'}(\mathbf{x}'_t, t, \mathbf{x}, y')}{1 - \alpha_t}\right) \tag{5}$$

We instantiate $\boldsymbol{\mu}_\theta^{\text{num}}$ and $\boldsymbol{\mu}_\theta^{\text{cat}'}$ using a shared Transformer-based backbone, which effectively captures dependencies across heterogeneous feature types.

The model is optimized via a joint loss function $\mathcal{L} = \lambda_{\text{num}}\mathcal{L}_{\text{num}} + \lambda_{\text{cat}}\mathcal{L}_{\text{cat}}$, balancing the numerical reconstruction error and categorical cross-entropy:

$$\mathcal{L}_{\text{num}} = \mathbb{E}_{t, \mathbf{x}'_t, \mathbf{x}, y'} \|\boldsymbol{\mu}_\theta^{\text{num}}(\mathbf{x}'_t, t, \mathbf{x}, y') - \boldsymbol{\epsilon}\|_2^2, \tag{6}$$

$$\mathcal{L}_{\text{cat}} = \mathbb{E}_{t, \mathbf{x}'_t, \mathbf{x}, y'} \frac{\alpha'_t}{1 - \alpha_t} \chi_{\{\mathbf{x}'_t = \mathbf{m}\}} \log\langle \boldsymbol{\mu}_\theta^{\text{cat}}(\mathbf{x}'_t, t, \mathbf{x}, y'), \mathbf{x}_0^{\text{cat}'}\rangle. \tag{7}$$

By training on triplets $(\mathbf{x}, \mathbf{x}'_0, y')$ acquired through our k-DPP procedure, the backbone learns to synthesize counterfactuals that inherently reflect the diversity of the counterfactual manifold.

3.4 Efficient Counterfactual Inference via Trajectory Branching

To generate a diverse set of k CFs, we consider two options. In the first approach, we generate a pool of CFs via independent diffusion trajectories, which is followed by k-DPP sampling to retrieve a representative set of diverse high-quality CFs.

However, this procedure may fail if we impose additional user requirements, e.g. actionability constraints. To address this, we consider a second variant and introduce a *branching generation* strategy, drawing inspiration from SVDD-style guidance Li et al. (2024). Given an input $\mathbf{x}$ and target class y', we initiate the generation process from a latent state $\hat{\mathbf{x}}_t$ obtained by noising $\mathbf{x}$ up to an intermediate timestep $t < T$. Rather than following a single sample path, at every reverse step we branch the current state into K candidate transitions $\mathbf{x}_{t-1}^{(1)}, \ldots, \mathbf{x}_{t-1}^{(K)}$. Each candidate is scored with the user's preference evaluated on the model's prediction of the clean sample $\hat{\mathbf{x}}_0$, e.g. the target-class confidence, proximity to $\mathbf{x}$, and a penalty for violating actionability constraints, and one candidate is retained with probability proportional to the softmax of its score (greedily at the final step). We continue the reverse process from the retained state down to $t = 0$.

To generate multiple counterfactuals, we run P such branching trajectories independently and in parallel, yielding a pool of candidates $\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(P)}$. We then apply k-DPP selection to the counterfactuals to obtain a set of k diverse, high-quality counterfactuals. Algorithm 1 provides pseudocode in Appendix B.

4 Experiments

We evaluate DICTUM in three complementary settings. We first benchmark counterfactual quality on standard tabular datasets. We then stress the model under inference-time *actionability* constraints through a constraint-strictness sweep. Finally, we ablate the two DPP-based design choices (training-set curation and inference-time selection) and characterise the role of the k-DPP design parameters σ and k.

Experimental Protocol. We evaluate on five widely used tabular datasets for counterfactual benchmarking: *Adult Income*, *Bank Marketing*, *Credit Default*, *Give Me Some Credit (GMC)*, and *Lending Club*. All datasets correspond to binary classification tasks involving a mixture of numerical and categorical attributes (between 9 and 23 features), and each is associated with a real-world recourse scenario in which counterfactual explanations are practically meaningful. We follow the preprocessing protocol of Furman et al. (2025): numerical features are standardized, categorical features are ordinal-encoded, and each dataset is split into training and test partitions. Further dataset details are provided in Appendix A. For each dataset, we train a deep neural network classifier (see Appendix F).

We compare DICTUM against three representative families of counterfactual explainers: (i) *DiCE* (Mothilal et al., 2020), a per-instance optimization method that incorporates a determinantal point process diversity term in the loss; (ii) *C-CHVAE* (Pawelczyk et al., 2020), a conditional variational autoencoder that searches the learned latent space for proximal counterfactuals; and (iii) *DiCoFlex* (Furman et al., 2025), a conditional normalizing flow trained on k-NN opposite-class targets that produces diverse counterfactuals in a single forward pass. These baselines span the dominant paradigms used in recent literature: gradient-based optimization, latent-space VAE search, and one-shot conditional generative modeling, allowing a fair comparison of DICTUM against the state of the art.

Following Furman et al. (2025), we evaluate $Q{=}2000$ test queries. Each method generates a pool of $M{=}100$ counterfactual candidates per query, from which $K{=}10$ valid CFs are returned. The baselines randomly select valid candidates, whereas DICTUM uses post-hoc k-DPP selection in the main benchmark. Reported means and standard deviations are computed over the test set. We use: *Validity*: the fraction of generated counterfactuals classified by h as the target class. *Proximity (Prox.)*: mean ℓ_1 distance between the original and counterfactual on standard-scaled (z-score) continuous features. *Sparsity (Spars.)*: fraction of categorical features changed. *LOF*: Local Outlier Factor of the counterfactual relative to the training distribution, used as a plausibility proxy (lower is more in-distribution). *Diversity*: mean pairwise cosine distance between counterfactual change vectors, computed in a joint space of numerical and one-hot encoded categorical features. We also report training time *Train* and inference time *Inference* in seconds. Training time is not applicable to optimization-based methods, as they require no separate training stage.

Benchmark. Validity is effectively saturated in Table 1, so the meaningful comparison is how the methods balance proximity, sparsity, plausibility, and diversity. DICTUM combines the best

Table 1: Main benchmark across five tabular datasets (neural-network classifier; N=10 counterfactuals per query, averaged over 2000 queries). Training and inference times are reported for the entire training process and inference across all samples. **Best** and second-best per column highlighted.

Method	Validity ↑	Prox. ↓	Spars. ↓	LOF ↓	Diversity ↑	Train [s] ↓	Inference [s] ↓
			Adult Income				
DICE	**1.00 ± 0.00**	1.08 ± 0.03	**0.14 ± 0.00**	0.50 ± 0.01	0.57 ± 0.02	–	426 ± 12
CCHVAE	**1.00 ± 0.00**	0.34 ± 0.01	0.28 ± 0.03	0.40 ± 0.02	0.17 ± 0.02	–	18360 ± 1622
DiCoFlex	**1.00 ± 0.00**	0.89 ± 0.12	0.29 ± 0.01	0.45 ± 0.05	0.35 ± 0.05	23976 ± 2844	5 ± 0
DICTUM	**1.00 ± 0.00**	**0.23± 0.00**	0.26± 0.00	**0.17± 0.00**	**0.74± 0.00**	12620 ± 181	35 ± 0
			Bank Marketing				
DICE	**1.00 ± 0.00**	1.12 ± 0.02	**0.09 ± 0.00**	0.26 ± 0.01	0.58 ± 0.03	–	361 ± 6
CCHVAE	**1.00 ± 0.00**	0.50 ± 0.00	0.19 ± 0.03	**0.13 ± 0.01**	0.06 ± 0.02	–	10477 ± 1835
DiCoFlex	**1.00 ± 0.00**	0.89 ± 0.08	0.13 ± 0.01	0.22 ± 0.02	0.34 ± 0.02	19656 ± 468	4 ± 0
DICTUM	**1.00 ± 0.00**	**0.34± 0.00**	0.17± 0.00	0.14± 0.00	**0.72± 0.00**	10430 ± 178	46 ± 0
			Credit Default				
DICE	**1.00 ± 0.00**	1.18 ± 0.11	**0.10 ± 0.01**	0.83 ± 0.06	**0.80 ± 0.03**	–	588 ± 25
CCHVAE	**1.00 ± 0.00**	0.43 ± 0.01	0.38 ± 0.01	0.44 ± 0.03	0.08 ± 0.01	–	20444 ± 1260
DiCoFlex	0.99 ± 0.01	0.60 ± 0.03	0.25 ± 0.00	0.30 ± 0.02	0.34 ± 0.02	25164 ± 395	6 ± 0
DICTUM	**1.00 ± 0.00**	**0.27± 0.01**	0.19± 0.00	**0.17± 0.00**	0.75± 0.01	12982 ± 64	62 ± 0
			GMC				
DICE	**1.00 ± 0.00**	2.33 ± 0.21	0.21 ± 0.01	1.15 ± 0.04	0.83 ± 0.02	–	282 ± 3
CCHVAE	**1.00 ± 0.00**	0.37 ± 0.01	0.32 ± 0.07	0.54 ± 0.16	0.12 ± 0.03	–	16411 ± 2952
DiCoFlex	**1.00 ± 0.00**	0.36 ± 0.03	**0.03 ± 0.00**	0.68 ± 0.04	0.52 ± 0.01	19368 ± 5976	3 ± 0
DICTUM	**1.00 ± 0.00**	**0.24± 0.01**	0.14± 0.01	**0.20± 0.01**	**0.92± 0.00**	10772 ± 85	29 ± 0
			Lending Club				
DICE	**1.00 ± 0.00**	2.25 ± 0.29	**0.13 ± 0.01**	0.84 ± 0.08	0.46 ± 0.06	–	302 ± 19
CCHVAE	**1.00 ± 0.00**	0.57 ± 0.01	0.45 ± 0.02	0.16 ± 0.01	0.07 ± 0.00	–	10548 ± 2010
DiCoFlex	**1.00 ± 0.00**	0.67 ± 0.01	0.18 ± 0.01	0.10 ± 0.01	0.33 ± 0.02	21774 ± 4104	3 ± 0
DICTUM	**1.00 ± 0.00**	**0.37± 0.01**	0.25± 0.00	**0.07± 0.00**	**0.75± 0.01**	11017 ± 49	33 ± 0

proximity on all five datasets with the highest diversity and lowest LOF on four, indicating that its additional alternatives do not arise simply from moving farther away or leaving the data manifold. Credit Default makes this distinction clear – DiCE is slightly more diverse, but has substantially worse proximity and LOF. The runtime profile places DICTUM between one-shot DiCoFlex and costly per-query search. Diffusion adds moderate inference overhead, but remains much faster than DiCE and C-CHVAE and trains in roughly half the reported time of DiCoFlex.

Inference-time guidance for actionability constraints. A user can specify actionability constraints for features that must remain unchanged or change only within a given range. DICTUM enforces these constraints *at inference time* without retraining, shortening the overall processing pipeline. At each reverse step, the branching strategy propagates $M = 5$ candidate trajectories to $t = 0$ and scores them with a constraint-aware utility. The best constraint-satisfying candidate ($K = 1$) is added to the final set. To generate each subsequent counterfactual, the model starts a new reverse process from the initial state and repeats this procedure until it reaches the requested number of explanations. This evaluation is stricter than the unconstrained benchmark because the counterfactuals must satisfy externally imposed feature constraints while remaining on the data manifold.

Table 4 reports the actionability results for three increasingly strict constraint setups on *Default of Credit Card*. Setup 1 fixes the demographic core (`SEX`, `MARRIAGE`); Setup 2 additionally requires `AGE` and `EDUCATION` to move only upward; Setup 3 further locks the two most recent payment-history fields to a non-decreasing trajectory and two most recent bill-history as immutable features. Appendix C details the feature-level constraint configuration for each setup and provides experiments on other datasets.

Table 2: Constraint-strictness sweep on *Credit Default*. Setups 1–3 progressively freeze more features (see Appendix C). M=100 CFs per query. **Best** and second-best per column highlighted.

Method	Validity ↑	Prox.-Cont ↓	Spars.-Cat ↓	LOF ↓	Diversity ↑
Setup 1					
DICE	**1.00**	0.81	0.16	0.68	**0.89**
CCHVAE	**1.00**	0.54	0.20	0.32	0.17
DiCoFlex	0.85	0.94	0.17	0.64	0.22
DICTUM	**1.00**	**0.25**	**0.13**	**0.18**	0.70
Setup 2					
DICE	**1.00**	0.93	0.15	0.71	**0.87**
CCHVAE	**1.00**	0.64	0.18	0.43	0.17
DiCoFlex	0.22	4.19	**0.00**	1.28	0.12
DICTUM	**1.00**	**0.24**	0.14	**0.18**	0.67
Setup 3					
DICE	**1.00**	1.03	0.16	0.78	**0.84**
CCHVAE	**1.00**	1.28	0.14	**0.76**	0.21
DiCoFlex	0.23	5.58	**0.00**	1.05	0.24
DICTUM	**1.00**	**0.22**	0.14	**0.18**	0.68

It is evident from the constraint-strictness analysis that DICTUM preserves both proximity and plausibility as the constraints become more restrictive. While all baselines experience severe proximity degradation, DICTUM consistently achieves the best proximity and LOF across all setups. Notably, DiCoFlex struggles to handle the strict constraints even when it resorts to generating highly out-of-distribution samples.

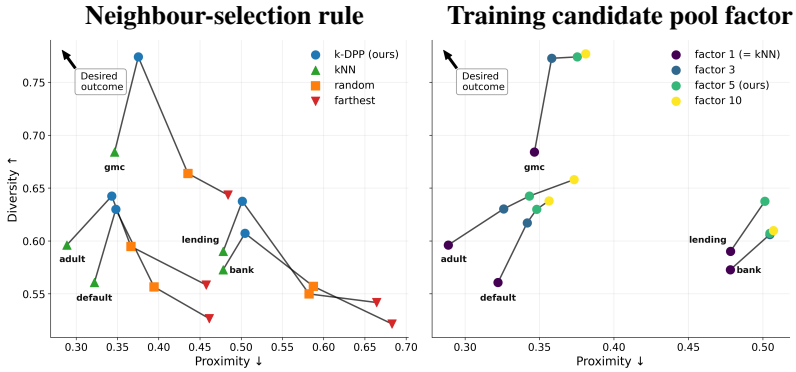


Figure 2: Impact of Neighbour-Selection Strategies on the Proximity–diversity trade-off across four datasets (grey lines join settings by dataset). **Left:** k-DPP improves diversity over kNN with a modest distance cost. **Right:** Larger candidate pools generally increase both diversity and query distance.

Component ablation. Figure 2 evaluates the impact of training-set curation strategies on the proximity-diversity trade-off. The left panel shows that while kNN selection reliably achieves the best proximity, our k-DPP selection significantly increases diversity across all datasets with only a moderate proximity cost. In contrast, random and farthest selections generally yield sub-optimal trade-offs, often degrading both metrics. The right panel demonstrates the effect of the candidate pool multiplier (factor). Expanding the pool by increasing this factor consistently boosts diversity but progressively degrades proximity. Consequently, we adopt a factor of 5 as our default configuration to establish a strong, practical balance between generating varied and localized counterfactuals.

k-DPP design parameters. Figure 3 sweeps the two design weights of the k-DPP post-hoc selection stage on Adult Income: the proximity weight σ_q and the repulsion weight σ_s. Increasing σ_q forces the algorithm to explore further from the original query, significantly raising diversity

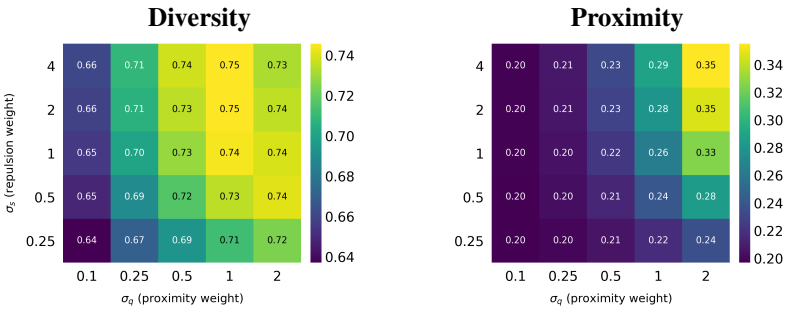


Figure 3: Post-hoc k-DPP sensitivity on Adult Income. Cells show diversity (left; higher is better) and proximity (right; lower is better) for proximity scale σ_q and repulsion scale σ_s. Larger σ_q generally improves diversity at a proximity cost.

but degrading proximity. Conversely, increasing the repulsion weight σ_s strongly penalises redundancy among candidates, further boosting diversity without drastically compromising proximity at lower σ_q levels. Together, these parameters cleanly control the trade-off between the generated set's diversity and its local concentration around the query.

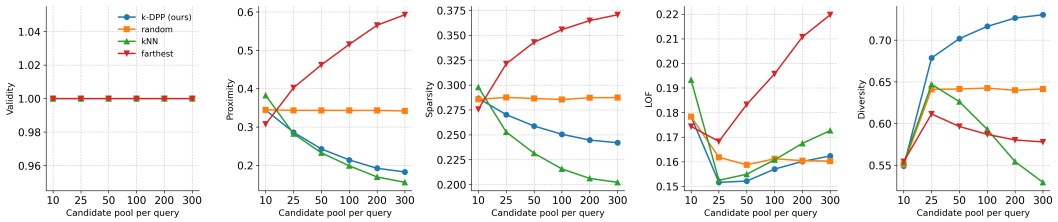


Figure 4: Sensitivity of DICTUM to the k-DPP postprocessing on Adult Income: each rule returns $k = 10$ counterfactuals from a per-query pool of 10–300. As the pool grows, k-DPP improves diversity while approaching kNN on proximity, sparsity, and LOF.

Candidate pool size. Figure 2 shows that k-DPP target selection improves diversity over kNN across four datasets approaching kNN proximity cost; random and farthest selection offer weaker trade-offs. Larger candidate pools increase diversity and query distance, motivating our default pool factor of 5.

5 Conclusion

We presented DICTUM, a conditional diffusion framework that treats diversity as a structural property of the learned distribution rather than a post-hoc score. The introduced greedy k-DPP curates training targets that are simultaneously close to the query and mutually repulsive while a Transformer-based mixed-type diffusion backbone with per-column learnable noise schedules turns this signal into a broad conditional distribution. Moreover, a branching generation strategy enforces user-specified actionability constraints at inference time without retraining. Across five benchmarks, DICTUM achieves the best continuous proximity on all five and the highest diversity and lowest LOF on four. The training-target analysis shows that DPP curation increases diversity at a moderate proximity cost, while the post-hoc analysis shows how kernel scales and candidate-pool size control the quality–diversity trade-off. The constraint sweep further shows that the complete pipeline preserves validity, proximity, and plausibility under increasingly restrictive requirements.

Limitations: The main residual weakness is categorical sparsity, where single-criterion optimisers remain stronger; promising future directions include per-criterion guidance (proximity- or sparsity-driven, rather than only actionability) and robustness to non-stationary classifiers.

AI USE STATEMENT

In this work, we used generative AI tools for retrieval and discovery (e.g., finding related work) and to aid or polish writing. We have not used generative AI tools for generating novel scientific claims, experimental data, or core algorithmic contributions. We have reviewed all AI-assisted work. Specifically, all AI-suggested references and related works were manually verified for existence, relevance, and correctness through independent literature searches. Furthermore, any text polished by AI was carefully checked. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses in the field of eXplainable AI (XAI) and relies exclusively on standard, publicly available tabular datasets. Our research does not involve human subjects, does not utilize sensitive personal data, and does not present direct risks of harmful societal applications. On the contrary, by advancing the generation of counterfactual explanations, the proposed methodology inherently aim to improve transparency, fairness, and user agency in machine learning systems. Therefore, the authors do not anticipate any negative ethical concerns arising from this submission.

REPRODUCIBILITY STATEMENT

To ensure full reproducibility of the experimental results presented in this work, we have included the complete source code as supplementary material alongside this submission. The provided repository contains the necessary scripts for data preprocessing, model training, and counterfactual generation. Furthermore, Section 3 provides a detailed description of our proposed method, while all relevant hyperparameter configurations and implementation details are thoroughly documented in Appendix 6.

REFERENCES

André Artelt and Barbara Hammer. Convex density constraints for computing plausible counterfactual explanations. In *International conference on artificial neural networks*, pp. 353–365. Springer, 2020.

Daniele Calandriello, Michał Dereziński, and Michal Valko. Sampling from a k-DPP without looking at all items. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature (PPSN)*, pp. 448–469, 2020.

Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018.

Michael Downs, Jonathan L Chu, Yaniv Yacoby, Finale Doshi-Velez, and Weiwei Pan. Cruds: Counterfactual recourse using disentangled subspaces. *ICML WHI*, 2020:1–23, 2020.

Tri Dung Duong, Qian Li, and Guandong Xu. Ceflow: A robust and efficient counterfactual explanation framework for tabular data using normalizing flows. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 133–144. Springer, 2023.

Oleksii Furman, Ulvi Movsum-zada, Patryk Marszalek, Maciej Zieba, and Marek Śmieja. Dicoflex: Model-agnostic diverse counterfactuals with flexible control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

John C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4): 857–871, 1971. doi: 10.2307/2528823. URL https://doi.org/10.2307/2528823.

Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking: R. guidotti. *Data Mining and Knowledge Discovery*, 38(5):2770–2824, 2024.

Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*, 2019.

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

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. 55(5), 2022. ISSN 0360-0300. doi: 10.1145/3527848. URL https://doi.org/10.1145/3527848.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pp. 17564–17579. PMLR, 2023.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, pp. 100–111, 2018.

Chaejeong Lee, Jayoung Kim, and Noseong Park. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*, pp. 18940–18956. PMLR, 2023.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Divyat Mahajan, Chenhao Tan, and Amit Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, pp. 607–617. ACM, 2020.

Emmanouil Panagiotou, Manuel Heurich, Tim Landgraf, and Eirini Ntoutsi. Tabcf: Counterfactual explanations for tabular data using a transformer-based vae. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 274–282, 2024.

Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference (WWW)*, pp. 3126–3132. ACM, 2020.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. FACE: Feasible and actionable counterfactual explanations. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, pp. 344–350, 2020.

Maximilian Schleich, Zixuan Geng, Yihong Zhang, and Dan Suciu. Geco: Quality counterfactual explanations in real time. *Proc. VLDB Endow.*, 14(9):1681–1693, 2021. doi: 10.14778/3461535.3461555. URL http://www.vldb.org/pvldb/vol14/p1681-schleich.pdf.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. TabDiff: A mixed-type diffusion model for tabular data generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Ilia Stepin, Jose M Alonso, Alejandro Catala, and Martín Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *Ieee Access*, 9:11974–12001, 2021.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Conference on Fairness, Accountability, and Transparency (FAT*)*, pp. 10–19, 2019.

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

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Madeleine Shroff. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*, 56(12):1–42, 2024.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31 (2):841–887, 2017.

Patryk Wielopolski, Oleksii Furman, Jerzy Stefanowski, and Maciej Zieba. Probabilistically plausible counterfactual explanations with normalizing flows. In *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pp. 954–961. IOS Press, 2024. doi: 10.3233/FAIA240584.

Wenzhuo Yang, Jia Li, Caiming Xiong, and Steven CH Hoi. Mace: An efficient model-agnostic framework for counterfactual explanation. *arXiv preprint arXiv:2205.15540*, 2022.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *International Conference on Learning Representations*, volume 2024, pp. 52829–52857, 2024.

A DATASETS

We evaluate our method across five widely used tabular benchmark datasets. To ensure consistency, we utilized representative subsets and specific feature configurations as follows:

Lending Club This dataset offers a practical scenario from the peer-to-peer lending sector, where the goal is to predict loan defaults. We use a subset of 30,000 instances that capture both individual borrower profiles (such as income and credit scores) and specific loan terms. The target variable representing a charged-off loan accounts for 22% of the data.
Available at: `https://www.kaggle.com/datasets/wordsforthewise/lending-club`.

Give Me Some Credit (GMC) Designed to predict the likelihood of serious financial distress (defined as being over 90 days past due), GMC provides a challenging benchmark due to its severe class imbalance—only 7% of the samples belong to the positive class. Our evaluation relies on a 30,000-sample subset encompassing key financial indicators like age and debt-to-income ratios.
Available at: `https://www.kaggle.com/c/GiveMeSomeCredit`.

Bank Marketing Tracking the outcomes of telemarketing campaigns run by a Portuguese bank, this task focuses on identifying clients likely to open a term deposit. We sampled 30,000 records from the original dataset, retaining its natural 11% positive class distribution, with features covering client demographics, macroeconomic indicators, and prior contact history.
Available at: `https://archive.ics.uci.edu/ml/datasets/bank+marketing`.

Credit Default Focusing on predicting next-month credit card defaults among clients in Taiwan, this dataset stands out due to its rich behavioral context. It includes 27,000 samples described by 23 features (14 numerical, 9 categorical) that capture complex financial habits, including rolling billing statements and historical payment delays.
Available at: `https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`.

Adult Income Derived from the 1994 U.S. Census, this classic benchmark is a staple in algorithmic fairness and explainability research due to the presence of sensitive attributes like race and gender. The objective is to classify whether a person earns over $50k annually based on their demographic and employment background. We utilize the standard 30,000 samples (with a 24% positive rate).
Available at: `https://archive.ics.uci.edu/ml/datasets/adult`.

B TRAJECTORY-BRANCHING INFERENCE

Algorithm 1 formalizes the branching procedure described in Section 3.4. Each of the P independent trajectories starts by noising the factual instance $\mathbf{x}$ to an intermediate step t_0. At every reverse step, the model draws K candidate transitions from the current state. For each candidate, the denoiser predicts a clean sample $\hat{\mathbf{x}}_0^{(j,c)}$, which is scored using the preference function

$$u(\hat{\mathbf{x}}_0; \mathbf{x}, y') = \min\big(\ell_{y'}(\hat{\mathbf{x}}_0), 2\big) - \alpha \|\hat{\mathbf{x}}_0^{\text{num}} - \mathbf{x}^{\text{num}}\|_2^2 - \beta \,\#\{\text{changed categorical features}\} - \gamma\, v(\hat{\mathbf{x}}_0; \mathbf{x}).$$

Here $\ell_{y'}$ is the target-class classifier logit, and v penalizes violations of immutability and monotonicity constraints. The scores $w_c = \lambda_t u(\hat{\mathbf{x}}_0^{(j,c)}; \mathbf{x}, y')$, with $\lambda_t = \lambda(t + 0.2)$, form the vector $\mathbf{w} = (w_1, \ldots, w_K)$. At intermediate steps, one branch is sampled according to $\mathrm{softmax}(\mathbf{w})$; at the final step, the branch with the highest score is chosen. The retained state is then projected via $\Pi_{\mathbf{x}}$ onto values admissible under the immutability and monotonicity constraints relative to $\mathbf{x}$. The completed states form a pool $\mathcal{P}$. Feasibility is evaluated on these completed samples $\mathbf{c} \in \mathcal{P}$, rather than on the denoiser predictions:

$$\mathcal{A}(\mathbf{c}; \mathbf{x}, y') = \mathbb{1}[\hat{y}(\mathbf{c}) = y' \wedge \ \mathbf{c} \text{ satisfies the user-specified actionability constraints relative to } \mathbf{x}] \,.$$

The feasible pool is $\mathcal{V} = \{\mathbf{c} \in \mathcal{P} : \mathcal{A}(\mathbf{c}; \mathbf{x}, y') = 1\}$. Then greedy k-DPP selection returns k diverse counterfactuals.

Algorithm 1 Counterfactual inference via guided pool sampling and k-DPP selection

Require: Factual instance $\mathbf{x}$, target class y', reverse process p_θ with denoiser D_θ, noise level $\eta \in (0, 1]$, number of steps T, pool size P, number of outputs k, branches per step K, preference score u, score scale λ_t, feasibility predicate $\mathcal{A}$, projection $\Pi_{\mathbf{x}}$

$t_0 \leftarrow \max\{1, \lfloor \eta(T-1) \rfloor\}$, $\quad \mathcal{P} \leftarrow [\,]$
for $j = 1, \ldots, P$ **do** ▷ run independent trajectories
  $\mathbf{x}_{t_0}^{(j)} \sim q(\mathbf{x}_{t_0} \mid \mathbf{x})$ ▷ noise the factual instance
  for $t = t_0, \ldots, 1$ **do** ▷ reverse diffusion
    for $c = 1, \ldots, K$ **do** ▷ branch the current state
      $\mathbf{z}_{t-1}^{(j,c)} \sim p_\theta(\cdot \mid \mathbf{x}_t^{(j)}, \mathbf{x}, y')$ ▷ draw one candidate transition
      $\hat{\mathbf{x}}_0^{(j,c)} \leftarrow D_\theta(\mathbf{z}_{t-1}^{(j,c)}, t-1; \mathbf{x}, y')$ ▷ predict its clean sample
      $w_c \leftarrow \lambda_t\, u(\hat{\mathbf{x}}_0^{(j,c)}; \mathbf{x}, y')$ ▷ score the prediction
    end for
    $\mathbf{w} \leftarrow (w_1, \ldots, w_K)$ ▷ collect branch scores
    if $t > 1$ **then**
      $c^* \sim \mathrm{Categorical}\big(\mathrm{softmax}(\mathbf{w})\big)$ ▷ sample a branch
    else
      $c^* \leftarrow \arg\max_{c \in \{1,\ldots,K\}} w_c$ ▷ choose greedily at the final step
    end if
    $\mathbf{x}_{t-1}^{(j)} \leftarrow \Pi_{\mathbf{x}}\big(\mathbf{z}_{t-1}^{(j,c^*)}\big)$ ▷ continue from the retained branch
  end for
  APPEND$(\mathcal{P}, \mathbf{x}_0^{(j)})$ ▷ add the completed sample to the pool
end for
$\mathcal{V} \leftarrow [\,\mathbf{c} \in \mathcal{P} : \mathcal{A}(\mathbf{c}; \mathbf{x}, y') = 1\,]$ ▷ keep feasible counterfactuals
return GREEDYKDPP$(\mathcal{V}, k, \mathbf{x})$ ▷ select k diverse counterfactuals

C ACTIONABILITY CONSTRAINTS

Generating feasible counterfactuals requires respecting real-world logical and business constraints. To evaluate the robustness of our SVDD method against increasing difficulty, we designed experiments with progressively stricter constraint setups. Table 3 details the specific actionability constraints applied at each difficulty level for the Default dataset.

Table 3: Actionability constraints across three datasets per experimental setup. Higher setup levels impose constraints on a larger number of features.

Setup Level	Immutable Num.	Immutable Cat.	Increasing Num.	Increasing Cat.
		Credit Default		
Setup 1	—	Sex, Marriage	—	—
Setup 2	—	Sex, Marriage	Age	Education
Setup 3	Bill_AMT1, Bill_AMT2	Sex, Marriage	Age	Education, Pay_0, Pay_2
		Adult		
Setup 1	—	Sex, Race	—	—
Setup 2	—	Sex, Race, Native_country	Age	—
Setup 3	—	Sex, Race, Native_country, Marital_status	Age	Education
		Lending Club		
Setup 1	Int_rate	Grade	—	—
Setup 2	Int_rate, Installment, Fico_range_low	Grade	—	—
Setup 3	Int_rate, Installment, Fico_range_low, Fico_range_high	Grade	—	Emp_length

D EVALUATION METRICS

To formally assess the quality, feasibility, and diversity of the generated counterfactuals, we define a comprehensive set of metrics. Let $N = |X_{\text{test}}|$ be the number of test samples, $M = 100$ the pool of generated candidates, and $K = 10$ the final number of selected counterfactuals. Furthermore, let $\mathcal{C}$ denote continuous feature indices, $\mathcal{G}$ categorical one-hot groups, $R_j = 1$ the training standard deviation in standard space, and $c_g(x)$ the selected category index for group g. Table 5 summarizes the mathematical formulations and interpretations of the metrics used in our evaluation.

E ADDITIONAL ABLATIONS

To gain deeper insights into the behavior of DICTUM, we conduct additional ablation studies analyzing the impact of two critical hyperparameters: the number of diffusion steps (T) (Figure 5) and the initial noise level introduced via the SDEdit mechanism (Figure 6).

Our findings indicate that the performance of the model converges rapidly with respect to the number of diffusion steps; specifically, all evaluation metrics become highly stable and show diminishing returns once T exceeds 50 steps. Furthermore, investigating the initial noise level reveals a clear trade-off inherent to the diffusion process: increasing the noise magnitude significantly improves the validity and plausibility of the generated counterfactuals, but this comes at the cost of reduced

Table 4: Constraint-strictness sweep on *Adult* and *Lending club* . Setups progressively freeze more features (see Table 3). M=100 CFs per query. **Best** and second-best per column highlighted.

Method	Validity ↑	Prox.-Cont ↓	Spars.-Cat ↓	LOF ↓	Diversity ↑
			Adult		
Setup 1					
DICE	**1.00**	1.17	**0.13**	0.46	0.60
CCHVAE	**1.00**	0.60	0.17	0.22	0.15
DiCoFlex	**1.00**	1.87	0.25	0.39	0.11
DICTUM	**1.00**	**0.24**	0.21	**0.17**	**0.69**
Setup 2					
DICE	**1.00**	1.17	**0.13**	0.46	0.58
CCHVAE	**1.00**	0.44	0.19	0.37	0.16
DiCoFlex	**1.00**	1.94	0.25	0.36	0.09
DICTUM	**1.00**	**0.22**	0.22	**0.18**	**0.67**
Setup 3					
DICE	**1.00**	1.30	**0.11**	0.46	0.53
CCHVAE	**1.00**	0.56	0.13	0.54	0.13
DiCoFlex	**1.00**	1.95	0.23	0.38	0.09
DICTUM	**1.00**	**0.32**	0.17	**0.22**	**0.57**
			Credit Default		
Setup 1					
DICE	**1.00**	0.81	0.16	0.68	**0.89**
CCHVAE	**1.00**	0.54	0.20	0.32	0.17
DiCoFlex	0.85	0.94	0.17	0.64	0.22
DICTUM	**1.00**	**0.25**	**0.13**	**0.18**	0.70
Setup 2					
DICE	**1.00**	0.93	0.15	0.71	**0.87**
CCHVAE	**1.00**	0.64	0.18	0.43	0.17
DiCoFlex	0.22	4.19	**0.00**	1.28	0.12
DICTUM	**1.00**	**0.24**	0.14	**0.18**	0.67
Setup 3					
DICE	**1.00**	1.03	0.16	0.78	**0.84**
CCHVAE	**1.00**	1.28	0.14	0.76	0.21
DiCoFlex	0.23	5.58	**0.00**	1.05	0.24
DICTUM	**1.00**	**0.22**	0.14	**0.18**	0.68
			Lending club		
Setup 1					
DICE	**1.00**	2.65	0.32	1.09	0.52
CCHVAE	**1.00**	0.81	0.27	0.19	0.12
DiCoFlex	0.91	7.39	**0.10**	0.34	0.03
DICTUM	**1.00**	**0.37**	0.24	**0.10**	**0.59**
Setup 2					
DICE	0.99	2.80	0.33	1.13	0.47
CCHVAE	**1.00**	0.74	0.31	0.44	0.23
DiCoFlex	0.84	7.17	**0.09**	0.43	0.01
DICTUM	0.96	**0.23**	0.27	**0.13**	**0.60**
Setup 3					
DICE	0.89	3.24	0.26	1.04	0.42
CCHVAE	0.86	0.59	0.28	0.32	0.18
DiCoFlex	0.74	7.16	**0.14**	0.34	0.02
DICTUM	**0.92**	**0.22**	0.24	**0.13**	**0.59**

sparsity, as the model explores a wider region of the feature space. Detailed results are presented in the subsequent figures.

Table 5: Summary of Evaluation Metrics

Metric	Interpretation	Formulation
Selected Validity	Proportion of valid CFs among the final K selected counterfactuals. Higher is better.	$\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K} v_{ik}$
Continuous Proximity	Mean absolute error (L1 distance) between the original sample and the CF for continuous features. Lower is better.	$\frac{1}{NK\lvert\mathcal{C}\rvert}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j\in\mathcal{C}}\lvert\tilde{s}_{ikj}-\tilde{x}_{ij}\rvert$
Categorical Sparsity	Average proportion of categorical features changed. Lower is better.	$\frac{1}{NK\lvert\mathcal{G}\rvert}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{g\in\mathcal{G}}\mathbb{1}[s_{ik,g}\neq x_{i,g}]$
LOF (Realism)	Median Local Outlier Factor log-score of the CFs relative to the training distribution. Lower means more realistic.	$\frac{1}{N}\sum_{i=1}^{N}\text{median}_{s\in S_i}\log(-q(s)+10^{-8})$ *where $q(s)$ is the LOF score.*
Diversity	Directional diversity: one minus the mean pairwise cosine similarity between the change vectors $\mathbf{v}_s=\phi(s)-\phi(\mathbf{x}_i)$ of a query's CFs. Higher means more diverse options.	$\frac{1}{N'}\sum_{i}\left(1-\frac{1}{\lvert V_i^+\rvert(\lvert V_i^+\rvert-1)}\sum_{\substack{s,s'\in V_i^+\\ s\neq s'}}\frac{\mathbf{v}_s^\top\mathbf{v}_{s'}}{\lVert\mathbf{v}_s\rVert\,\lVert\mathbf{v}_{s'}\rVert}\right)$ *where $V_i^+=\{s\in V_i:\mathbf{v}_s\neq\mathbf{0}\}$.*

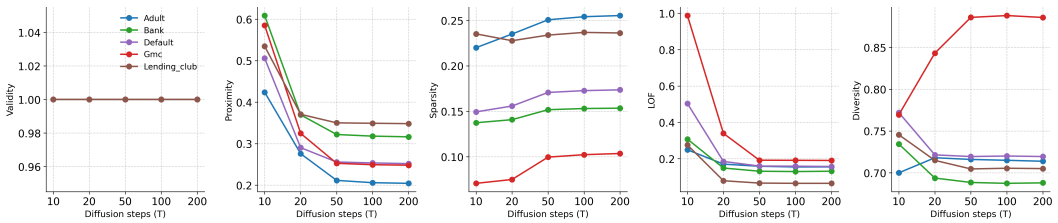


Figure 5: Effect of diffusion steps on the performance of DICTUM across evaluated datasets. Increasing the number of diffusion steps leads to greater stability across all metrics.

F Hyperparameters

F.1 Classifier

For every dataset we train a feed-forward neural network h (two hidden layers, 32 units each, ReLU activations, dropout 0.2, cross-entropy loss with inverse-frequency class weights and early stopping). Counterfactual explanations are computed with respect to this classifier: the target class label y' is fixed to the favorable class, and validity is measured by querying h on the generated counterfactuals. All generative methods are trained on classifier-relabeled data (Furman et al., 2025).

F.2 DICTUM

To ensure reproducibility and a fair baseline, our diffusion model architecture builds upon the foundational choices established in the original TabDiff implementation. Following their methodology, each feature column is individually projected to a $d = 4$ dimensional embedding, ensuring that all columns are treated with equal importance.

The core generative model utilizes a transformer-like UniMod architecture tailored for tabular data. Operating on the base parameters recommended in the original study, we configured the network to maintain a sufficiently wide bottleneck for capturing temporal dynamics during the diffusion process. Specifically, we set the temporal embedding dimension (dim_t) to 256 and the number of layers to 4. The attention mechanism operates with 1 head and a feed-forward expansion factor of 32. For the diffusion process itself, we utilize $T = 50$ timesteps with an initial noise level of 0.6 during the sampling phase. More details are described in 6.

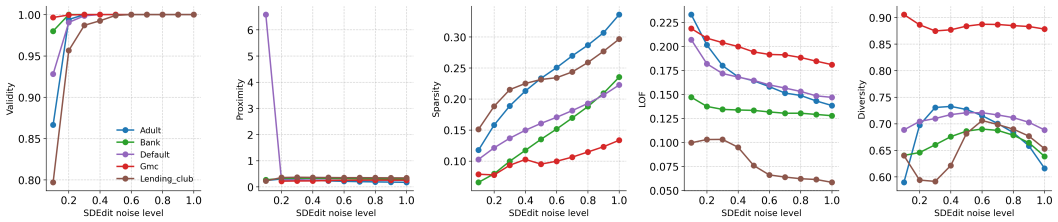


Figure 6: Impact of the initial noise level on the performance of DICTUM across different datasets. The results illustrate how the magnitude of added noise affects the quality and characteristics of the generated counterfactuals.

All experiments were conducted on a single NVIDIA A100 Tensor Core GPU. On average, the DICTUM training process spanning 4000 epochs required 3.5 hours to complete. To demonstrate the stability and convergence of our training procedure, Figure 7 illustrates the training loss over epochs. The curve, enhanced with Exponential Moving Average (EMA) smoothing, exhibits a graceful and consistent descent, confirming that the model converges effectively without significant oscillations.

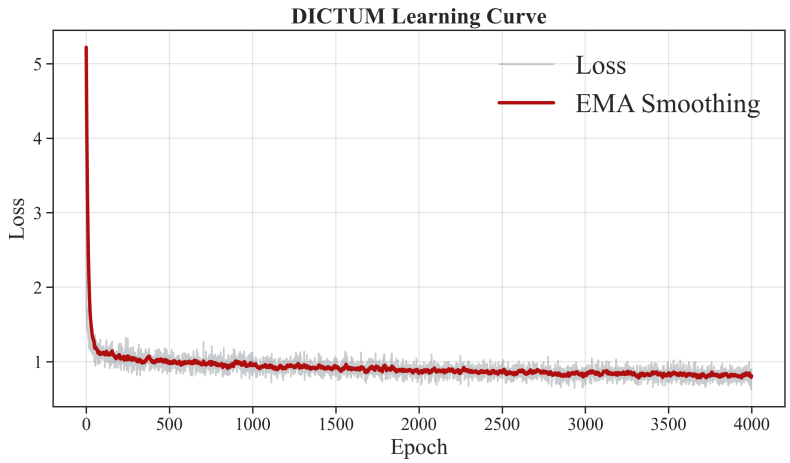


Figure 7: Training loss curve for DICTUM over 4000 epochs. The raw training loss is presented alongside an Exponential Moving Average (EMA) smoothed curve, illustrating a stable and continuous convergence over the training duration.

Table 6: Hyperparameters for Model Architecture, Diffusion, DICTUM Training, and Sampling.

Hyperparameter	Value
Model Architecture (UniMod)	
Number of Layers	4
Token Dimension (d_{token})	4
Number of Attention Heads	1
Time Embedding Dimension (dim_t)	256
Feed-Forward Factor	32
Condition Dropout Probability	0.25
Diffusion Process	
Timesteps (T)	50
Initial Noise Level (η)	0.6
Sampler	Stochastic, second-order correction
Training Targets (Opposite-Class Neighbours)	
Training Subsample Size	20,000
Feature Scaling (continuous)	Standardization
Candidate Pool (nearest, Gower distance)	$5 \times 150 = 750$
Neighbours Selected per Instance (greedy k-DPP)	150
Gower Categorical Weight	0.75
DICTUM Training	
Optimizer	Adam
Initial Learning Rate	1×10^{-3}
Learning Rate Scheduler	Cosine Annealing ($\eta_{\min} = 10^{-5}$)
Gradient Clipping (max norm)	1.0
Batch Size	512
Training Epochs	4000
Classifier	
Architecture	MLP, hidden layers $[32, 32]$
Optimizer / Learning Rate	Adam / 1×10^{-3}
Batch Size / Epochs	64 / 100
Post-Selection (k-DPP)	
Generated Pool Size (M)	100
Selected CFs per Instance (k)	10
Query Kernel Scale (σ_q)	0.5
Similarity Kernel Scale (σ_s)	0.5
Gower Categorical Weight	0.75
Branching Generation (SVDD, constrained setting)	
Candidates per Reverse Step (K)	5
Guidance Scale (λ)	20
Proximity Weight (α)	0.1
Categorical Change Weight (β)	1.0
Constraint Penalty Weight (γ)	100