COMPOSITIONAL WORLD KNOWLEDGE LEADS TO HIGH UTILITY SYNTHETIC DATA

Sachit Gaudi **Gautam Sreekumar** **Vishnu Naresh Boddeti**
Michigan State University
{gaudisac,sreekum1,vishnu}@msu.edu

ABSTRACT

Machine learning systems struggle with robustness, under subpopulation shifts. This problem becomes especially pronounced in scenarios where only a subset of attribute combinations is observed during training—a severe form of subpopulation shift, referred as *compositional shift*. To address this problem, we ask the following question: Can we improve the robustness by training on synthetic data, spanning all possible attribute combinations? We first show that training of conditional diffusion models on limited data lead to incorrect underlying distribution. Therefore, synthetic data sampled from such models will result in unfaithful samples and does not lead to improve performance of downstream machine learning systems. To address this problem, we propose COIND to reflect the compositional nature of the world by enforcing conditional independence through minimizing Fisher's divergence between joint and marginal distributions. We demonstrate that synthetic data generated by COIND is faithful and this translates to state-of-the-art worst-group accuracy on *compositional shift* tasks on CelebA. Our code is available at https://github.com/sachit3022/compositional-generation/

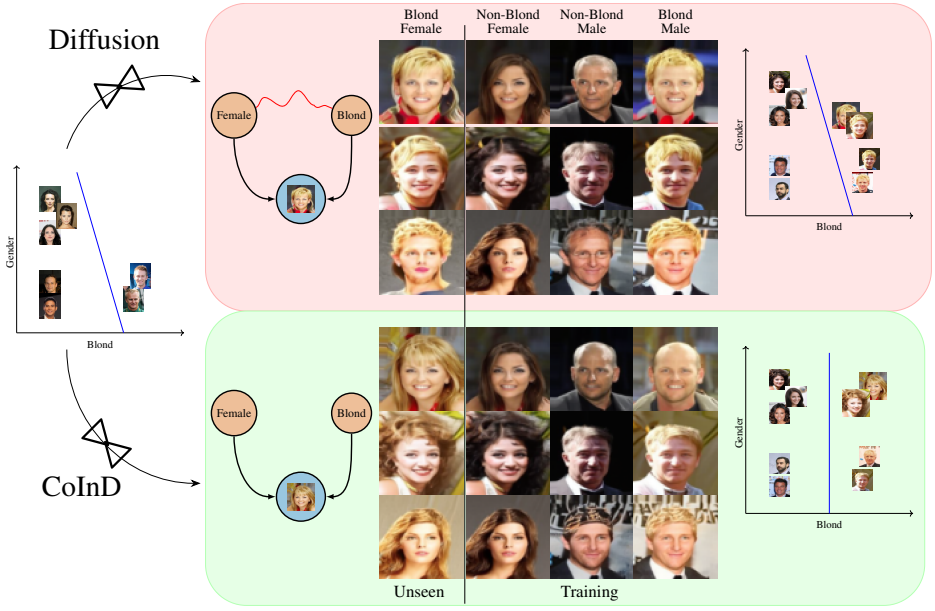


Figure 1: Training data comprises of all combinations except female blonds. Classifier trained to predict *blond* relies on information of *gender* due to their association in training data. Standard diffusion model trained on this data also learns this association leading to incorrect distribution, resulting in unfaithful generation of unseen combinations (female blondes).Consequently, synthetic data from this model fails to improve downstream classifier performance. In contrast, COIND leverages compositional world knowledge to learn the true distribution, facilitating accurate generation of unseen combinations. This leads to a robust classifier when trained on synthetic data from COIND.

1 INTRODUCTION

Consider CelebA Liu et al. (2015) dataset with 40 binary attributes would require over 1 trillion samples to span all combinations. Collecting such massive data for all attribute combinations is infeasible; Machine learning systems trained on subset of combinations will suffer from the problem of *compositional shift* (Mahajan et al., 2024). Consider a simplified base case of *compositional shift*, where *female blonds* are not observed during training but all the other combinations of gender, blond are observed as depicted in Fig. 1.

Humans can easily synthesize and compose attributes such as blond and gender, and therefore able to imagine *female blonds* from just observing *male blonds*, *female non-blonds*, and *male non-blonds* (Baroni & Lake, 2023). In this work, we want to learn generative models that have the capability to imagine. This imagination can be distilled to generate synthetic samples spanning all combinations. The success of the task depends on the models ability to faithfully generate unseen compositions. Standard diffusion models Ho et al. (2020a) are trained with an optimization designed to maximize likelihood, which results in memorization of training data rather than true generalization (Kamb & Ganguli, 2024). Diffusion models either fail to respect the gender, or learn incorrect interpolations between blond and gender attributes. The samples in Fig. 1(red box) reveal these limitations.

The difference between humans and diffusion models is that we understand the compositional nature of the world, allowing us to create complex composites from a set of primitive components Nye et al. (2020). However, diffusion models learn the associations from the data. In fact, we verify that diffusion models trained on limited data violate conditional independence, which is an important assumption in compositionality Nie et al. (2021); Liu et al. (2021). We point out that this violation stems from the incorrect objective of diffusion models under limited data. Therefore, we propose COIND to incorporate compositional world knowledge into diffusion model training by minimizing the fisher's divergence between conditional joint and product of conditional marginals, in addition to maximizing likelihood on the observed compositions.

COIND exhibits compositionality, effectively learning the true underlying data distribution. This results in the faithful generation of previously unseen attribute combinations, as clearly illustrated in the Fig. 1(green box). Notably, classifiers trained on this higher-quality synthetic data exhibit enhanced robustness and generalization capabilities, compared to standard diffusion models. These classifiers achieve SoTA results, significantly outperforming established baselines in subpopulation shift literature. Moreover, COIND offers a remarkably simple implementation, requiring only a few additional lines of code to standard diffusion model training.

2 COMPOSITIONAL WORLD

We study the model of learning compositional functions from limited data. As compositionality is a property of data generating process (Wiedemer et al., 2024). We consider the case, where the samples are generated by independently varying factors, and have access to the labels of these factors. Similar assumptions were discussed in (Wiedemer et al., 2024; Ahmed et al., 2020). In this section, we formally define the assumption in the data generation process.

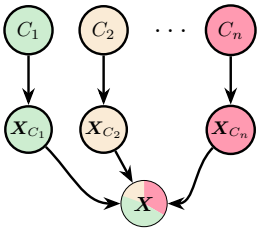


Figure 2: **Compositional world model**. $\boldsymbol{X}$ is generated by varying $C_1, C_2, \ldots, C_n$ independently in the underlying data generating process.

Notations. We use bold lowercase and uppercase characters to denote vectors (e.g., $\boldsymbol{a}$) and matrices (e.g., $\boldsymbol{A}$) respectively. Random variables are denoted by uppercase Latin characters (e.g., X). The distribution of a random variable X is denoted as $p(X)$, or as $p_{\boldsymbol{\theta}}(X)$ if the distribution is parameterized by a vector $\boldsymbol{\theta}$. With a slight abuse of notation, we refer to $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i)$ as marginal, and joint as $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C)$.

Data Generation Process. The data generation process consists of observed data $\boldsymbol{X}$ (e.g., images) and its attribute variables $C_1, C_2, \ldots, C_n$ (e.g., color, digit, etc.). To have explicit control over these attributes during generation, they should vary independently of each other. In this work, we limit our study to only those causal graphs in which the attributes are not causally related and can hence vary indepen-

dently, as shown in Fig. 2. Each C_i assumes values from a set $\mathcal{C}_i$ and their Cartesian product $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_n$ is referred to as the *attribute space*. Each attribute C_i generates its own observed component $\boldsymbol{X}_{C_i} = f_{C_i}(C_i)$, which together with unobserved exogenous variables $\boldsymbol{U}_{\boldsymbol{X}}$ form the composite observed data $\boldsymbol{X} = f(\boldsymbol{X}_{C_1}, \ldots, \boldsymbol{X}_{C_i}, \boldsymbol{U}_{\boldsymbol{X}})$ (see Fig. 2). We do not restrict f much except that it should not obfuscate individual observed components in $\boldsymbol{X}$ (Wiedemer et al., 2024). A simple example of f is the concatenation function. We also assume that all f_{C_i} are invertible and therefore it is possible to estimate $C_1, \ldots, C_n$ from $\boldsymbol{X}$. These assumptions together ensure that $C_1, \ldots, C_n$ are mutually independent given $\boldsymbol{X}$ despite being seemingly d-connected.

The attribute space in our problem statement has the following properties. **(1)** The attribute space observed during training $\mathcal{C}_{\text{train}}$ covers $\mathcal{C}$ in the following sense:

Definition 1 (Support Cover)**.** *Let $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_n$ be the Cartesian product of n finite sets $\mathcal{C}_1, \ldots, \mathcal{C}_n$. Consider a subset $\mathcal{C}_{train} \subset \mathcal{C}$. Let $\mathcal{C}_{train} = \{(c_{1j}, \ldots, c_{nj}) : c_{ij} \in \mathcal{C}_i, 1 \leq i \leq n, 1 \leq j \leq m\}$ and $\tilde{\mathcal{C}}_i = \{c_{ij} : 1 \leq j \leq m\}$ for $1 \leq i \leq n$. The Cartesian product of these sets is $\tilde{\mathcal{C}}_{train} = \tilde{\mathcal{C}}_1 \times \cdots \times \tilde{\mathcal{C}}_n$. We say $\mathcal{C}_{train}$ covers $\mathcal{C}$ iff $\mathcal{C} = \tilde{\mathcal{C}}_{train}$.*

Informally, this assumption implies that every possible value that C_i can assume is present in the training set, and open-set attribute compositions do not fall under this definition. **(2)** For every ordered tuple $c \in \mathcal{C}_{\text{train}}$, there is another $c' \in \mathcal{C}_{\text{train}}$ such that c and c' differ on only one attribute.

In the example discussed in § 1, C_1 represents the blond hair and C_2 encodes gender. Blond and gender independently generate an image, ($\boldsymbol{X}$), while both blond and gender can be reversely inferred from the image. We observe all attribute combinations except $(C_1, C_2) = (1, 0)$ (where $C_1 = 1$ denotes blond and $C_2 = 0$ indicates female), consistent with the assumptions outlined earlier.

Preliminaries on Score-based Models In this work, we train conditional score-based models (Song et al., 2021b) using classifier-free guidance (Ho & Salimans, 2022) to generate data corresponding by composing attributes. Score-based models learn the score of the observed data distributions $p_{\text{train}}(\boldsymbol{X})$ and $p_{\text{train}}(\boldsymbol{X} \mid C)$ through score matching (Hyvärinen, 2005). Once the score of a distribution is learned, samples can be generated using Langevin dynamics. Liu et al. (2022) proposed the following modifications during sampling to enable compositionality, assuming that the model learns the conditional independence relations from the data-generation process.

Compositional Sampling: $C_1 = c_1 \wedge C_2 = c_2$ generates data where attributes C_1 and C_2 takes values c_1 and c_2 respectively. Since, $p_{\boldsymbol{\theta}}(C_1 \wedge C_2 \mid \boldsymbol{X}) = p_{\boldsymbol{\theta}}(C_1 \mid \boldsymbol{X}) p_{\boldsymbol{\theta}}(C_2 \mid \boldsymbol{X})$ samples are generated for the composition $C_1 \wedge C_2$ by sampling from the following score:

$$\nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_1 \wedge C_2) = \nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_1) + \nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_2) - \nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}) \quad (1)$$

This formulation gives can me modified for additional flexibility of controlling for attributes, where γ controls the strength of attribute C_1 w.r.t C_2, $\nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_1 \wedge C_2)$ can be written as:

$$\gamma \nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_1) + \nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_2) - \gamma \nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}) \quad (2)$$

3 Diffusion models do not respect the compositional world

To achieve compositionality, diffusion models must preserve the properties of the underlying data distribution. As established in § 2, a critical property is conditional mutual independence (CI). Additionally, this property is used to sample data using compositional sampling.

We empirically verify whether diffusion models satisfy CI through two experimental configurations: **(1) Full support**, where the model observes all ordered pairs in the concept space $\mathcal{C}$, and **(2) Partial support**, limited to a training subset $\mathcal{C}_{\text{train}} \subset \mathcal{C}$ (visualized in Fig. 1).

Support	JSD ↓	CS↑
Full	0.37	96.58
Partial	0.62	41.80

Table 1: Jensen-Shannon divergence and Conformity Scores for CelebA dataset.

We measure CI violation as the disparity between the conditional joint distribution $p_{\boldsymbol{\theta}}(C \mid \boldsymbol{X})$ and the product of conditional marginal distributions $\prod_i^n p_{\boldsymbol{\theta}}(C_i \mid \boldsymbol{X})$ learned by the implicit classifier of diffusion model using Jensen-Shannon divergence as,

$$\text{JSD} = \mathbb{E}_{C, \boldsymbol{X} \sim p_{\text{data}}} \left[D_{\text{JS}} \left(p_{\boldsymbol{\theta}}(C \mid \boldsymbol{X}) \mid\mid \prod_i^n p_{\boldsymbol{\theta}}(C_i \mid \boldsymbol{X}) \right) \right] \quad (3)$$

where D_{JS} is the Jensen-Shannon divergence and $p_{\boldsymbol{\theta}}$ is obtained by following (Li et al., 2023) and evaluating the implicit classifier learned by the diffusion model. More details can be found in App. A.1.

A positive JSD value suggests that the model fails to adhere to the CI relation. As shown in Tab. 1, models trained on partially observed attribute compositions (partial support) exhibit substantially higher JSD values than those trained with full composition coverage. This result demonstrates that diffusion models directly propagate spurious dependencies present in the training data rather than learning conditionally independent attributes.

Violation in conditional independence originates from the standard training objective of diffusion models that maximize the likelihood of conditional generation. Under perfect loss, for every observed composition ($C \in \mathcal{C}_{train}$), the model accurately learns $p_{\text{train}}(\boldsymbol{X} \mid C)$, i.e., $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C) \approx p_{\text{train}}(\boldsymbol{X} \mid C) = p(\boldsymbol{X} \mid C)$, However, learn incorrect marginals, $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i) \approx p_{\text{train}}(\boldsymbol{X} \mid C_i) \neq p(\boldsymbol{X} \mid C_i)$. Refer to App. B.1 for complete proof. Informally, $p_{\text{train}}(\boldsymbol{X} \mid C_1 = 1)$, consists of only blond males. While, the underlying $p(\boldsymbol{X} \mid C_1 = 1)$ contains male, and females blonds. These incorrect marginals lead to violation in CI.

This violation leads to unfaithful generation, evident in lower conformity scores (CS) for partial-support models versus full-support (Tab. 1). CS quantifies alignment between generated attributes (inferred via ϕ_{C_i}) and conditioning attributes (C) (details: App. A.2). Samples generated under partial support (highlighted in red, Fig. 1) exhibit gender leakage —such as beards— when synthesizing unseen "blond females". Based on these observations, we propose CoInD to train diffusion models that explicitly enforce the conditional independence to encourage compositionality.

4 CoInD: Enforcing Conditional Independence Enables Compositionality

In the previous section, we observed that diffusion models violate conditional independence (CI) by learning incorrect marginals. To remedy this, CoInD uses a training objective that explicitly enforces the causal factorization to ensure that the trained diffusion models obey CI. Applying the assumption of $C_1 \perp\!\!\!\perp \ldots \perp\!\!\!\perp C_n \mid \boldsymbol{X}$ mentioned in § 2, we have $p(\boldsymbol{X} \mid C) = \frac{p(\boldsymbol{X})}{p(C)} \prod_i^n \frac{p(\boldsymbol{X}|C_i)p(C_i)}{p(\boldsymbol{X})}$. Note that the invariant $p(\boldsymbol{X} \mid C)$ is now expressed as the product of marginals employed for sampling. Therefore, training the diffusion model by maximizing this conditional likelihood is naturally more suited for learning accurate marginals for the attributes. We minimize the distance between the true conditional likelihood and the learned conditional likelihood as,

$$\mathcal{L}_{\text{comp}} = \mathcal{W}_2 \left(p(\boldsymbol{X} \mid C), \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X})}{p_{\boldsymbol{\theta}}(C)} \prod_i \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i) p_{\boldsymbol{\theta}}(C_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{X})} \right) \tag{4}$$

leveraging the Wasserstein distance upper bound via Fisher divergence (Kwon et al., 2022), we derive the following inequality:

$$\mathcal{L}_{\text{comp}} \leq K_1 \sqrt{\mathcal{L}_{\text{score}}} + K_2 \sqrt{\mathcal{L}_{\text{CI}}} \tag{5}$$

for constants $K_1, K_2 > 0$. A complete derivation of this bound is provided in App. B.2.

Distribution matching:

$$\mathcal{L}_{\text{score}} = \mathbb{E}_{p(\boldsymbol{X},C)} \| \nabla_{\boldsymbol{X}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C) - \nabla_{\boldsymbol{X}} \log p(\boldsymbol{X} \mid C) \|_2^2 \tag{6}$$

Conditional Independence:

$$\mathcal{L}_{\text{CI}} = \mathbb{E} \| \nabla_{\mathbf{X}} \log p_\theta(\mathbf{X} \mid C) - \nabla_{\mathbf{X}} \log p_\theta(\mathbf{X}) - \sum_i \left[\nabla_{\mathbf{X}} \log p_\theta(\mathbf{X} \mid C_i) - \nabla_{\mathbf{X}} \log p_\theta(\mathbf{X}) \right] \|_2^2 \tag{7}$$

Practical Implementation. A computational burden presented by $\mathcal{L}_{\text{CI}}$ in Eq. (7) is that the required number of model evaluations increases linearly with the number of attributes. To mitigate this burden, we approximate the mutual conditional independence with pairwise conditional independence (Hammond & Sun, 2006). Thus, the modified $\mathcal{L}_{\text{CI}}$ becomes,

$$\mathcal{L}_{\text{CI}} = \mathbb{E}_{p(\boldsymbol{X},C)}\mathbb{E}_{j,k}\|\nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X} \mid C_j, C_k) - \nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X} \mid C_j) - \nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X} \mid C_k) + \nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X})\|_2^2$$

The weighted sum of the square of the terms in Eq. (5) has shown stability. Therefore, CoInD's training objective:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{score}} + \lambda \mathcal{L}_{\text{CI}} \tag{8}$$

where λ is the hyper-parameter that controls the strength of conditional independence. The reduction to the practical version of the upper bound (Eq. (5)) is discussed in extensively in App. C. For guidance on selecting hyper-parameters in a principled manner, please refer to App. C.3. Finally, our proposed approach can be implemented with just a few lines of code, as outlined in Algorithm 1.

5 Results

We perform experiments on the CelebA dataset, downsampled to 64×64. pixels. As a remainder, diffusion model was trained on all compositions except for "blonde females." We followed similar settings for both the standard diffusion model and CoInD($\lambda = 100$). For detailed experimental settings, please refer to Appendix Section App. D

5.1 Compositionality leads to high utility synthetic data

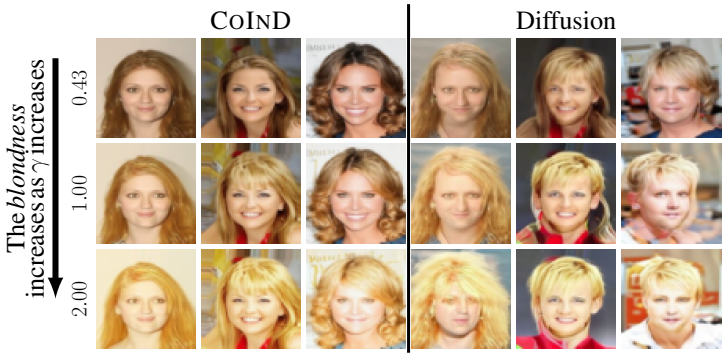


Figure 3: CoInD enables precise control (Eq. (2)) over blondness, while preserving gender attributes (left). Standard diffusion models exhibit gender bias by conflating blondness with specific genders due to training data correlations (right).

Our analysis, as shown in Tab. 2, demonstrates that CoInD significantly reduces violations of conditional independence, resulting in more faithful data generation. Fig. 1(highlighted in green) illustrates this improvement, showcasing CoInD's ability to accurately synthesize previously unseen "blonde females" compositions. CoInD adapts hairstyles from female celebrities and blends blonde shades from male counterparts, producing photorealistic blonde female figures - despite not being explicitly trained on these examples.

Method	JSD ↓	FID ↓	CS↑
Diffusion	0.62	41.80	64.77
CoInD	**0.16**	**21.64**	**81.05**

Table 2: JSD, CS, FID for CelebA dataset.

5.2 Synthetic Data Sampled from CoInD Yields Robust Classifier

Building on the demonstrated capability of CoInD to generate high-utility synthetic data, this section addresses the following question: Can leveraging this data enhance classifier robustness against compositional shifts?

Setup We generate 20,000 samples of synthetic data from the trained diffusion model by uniformly sampling (Eq. (2)) all possible compositions, including "blonde females." We then train a ResNet-18 classifier on this synthetic data to predict the "blonde" attribute. To evaluate performance, we measure the test accuracy, balanced test accuracy, and Worst Group Accuracy (WGA) on the downsampled (64×64) test set. We compare these results to the baselines borrowed from (Mahajan et al., 2024).

Discussion CoInD exhibits $\approx 5\%$ higher WGA compared to the baselines(Tab. 3), which makes it robust to compositional shift. However, other models exhibit high test accuracy due to the imbalance in the reference test set of CelebA, which contains many (84.6%) samples of the seen male and female non-blond, compared to that of (14.46%) challenging unseen female blonds.

Data	Method	Test Acc.	Balanced Acc.	Worst group Acc.
Real	ERM	87.0	59.3	4.0
	G-DRO (Sagawa et al., 2019)	91.7	**86.3**	71.7
	LC (Liu et al., 2023)	88.3	70.7	21.0
	sLA (Tsirigotis et al., 2024)	88.3	71.0	21.3
	CRM (Mahajan et al., 2024)	**93.0**	85.7	73.3
Synthetic	Diffusion	90.9	80.4	64.0
	CoInD	80.6	80.3	**76.9**

Table 3: **Classifier trained with synthetic data generated from CoInD achieves better Worst group Acc.** for classifying attribute "blond" attribute of CelebA dataset.

Method	Test Acc.	Balanced Acc.	Worst group Acc.
Diffusion	**61.19**	56.47	33.16
CoInD	56.39	**60.63**	**55.58**

Table 4: **Implicit classifier (generative classifier),** $p_{\boldsymbol{\theta}}(\text{blond} = \text{true} \mid \boldsymbol{X})$ **learned by CoInD achieves better Worst group Acc.** for classifying attribute "blond" attribute of CelebA dataset.

6 Insights

Implicit Classifier Learned by CoInD is Robust to Subpopulation Shift Li et al. demonstrate that the implicit classifier, which guides the generation in classifier-free diffusion models remains robust to subpopulation shifts in the absence of spurious labels. However, when spurious labels are available, traditional discriminative baselines that exploit these correlations retain an advantage—evident from the real data setting in Tab. 3—compared to the performance of implicit diffusion classifier reported in Tab. 4.

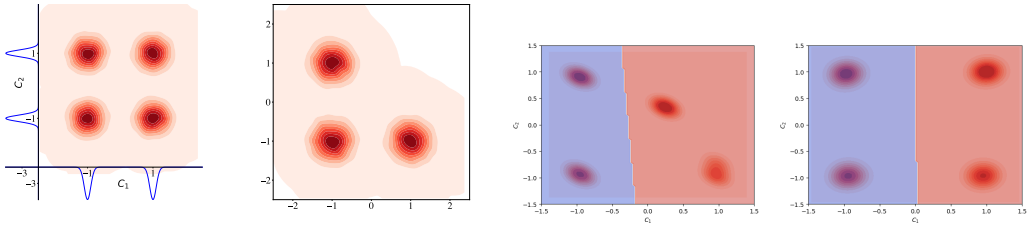
CoInD leverages the spurious labels to explicitly enforce conditional independence during training, resulting in a stronger implicit classifier that outperforms diffusion baseline under subpopulation shift (Tab. 4). However, they still trail behind the discriminator based approaches. This improved classifier compared to the standard diffusion models lead to more faithful image generation (Tab. 2), which in turn yields high-utility synthetic data. When used to train downstream classifiers, this data leads to improved robustness under distribution shift (Tab. 3).

Qualitatively Accurate Counterfactual Generation The ability to compose attributes is reflected in a model's capacity to generate meaningful counterfactuals—i.e., to answer "what if?" questions. In our experiments, we generate unseen blond female samples by posing counterfactual queries such as: *What if this non-blond female dyed her hair blond?* or *How would the model interpret a gender change from male to female when generating blond females from blond males?*

As illustrated in Fig. 1, Column 1 shows the counterfactual blond female generated from a non-blond female (Column 2). Notably, CoInD achieves this by modifying only the hair color while preserving all other facial attributes. In contrast, baseline diffusion model tend to generate shorter hair—a spurious correlation from the training distribution where short hair (male feature) correlates with blondes—thus yielding incorrect interpolations. This highlights CoInD's superior compositional ability to disentangle and manipulate attributes independently, which translates to a robust downstream classifier. More samples for inspection are provided in Fig. 6

Illustrative Setting: Theoretical Study Using a 2D Gaussian Mixture To elucidate the issue of incorrect interpolation, we derive the score for the mixture of Gaussians in closed form. As illustrated in Fig. 4, directly learning the marginal $p_\theta(X \mid C_0 = 1)$ from the training distribution (shown in panel (b)) results in a distribution that incorrectly represents a single Gaussian with mean $(1, -1)$ rather than the appropriate mixture comprising Gaussians with means $(1, -1)$ and $(1, 1)$. This error in modeling leads to an inaccurate samples when drawing from $p_\theta(X \mid C_0, C_1 = (1, 1))$, as depicted in panel (c). In contrast, CoInD leverages conditional independence to accurately learn $p_{\text{train}}(X \mid C_0 = 1)$ ensuring that the resulting distribution is faithful to the true data. Consequently,

this approach yields a robust classifier when trained on the correct data representation, as evidenced in panel (d). More details can be found in App. E.1.



(a) True underlying data distribution (b) Training data: Orthogonal Support (c) Conditional distribution learned by vanilla diffusion objective (d) Conditional distribution learned by CoInD

Figure 4: CoInD respects underlying independence conditions thereby generating true data distribution (d).

7 Related Work

Subpopulation shift is a long studied topic. Group DRO (Sagawa et al., 2019), a competitive baseline method, minimizes worst-group error as a proxy for generalization. In contrast, CoInD, enforces independence between spurious and target features in data generation—a constraint that defines spurious correlation. Compositional Risk Minimization (CRM) (Mahajan et al., 2024) is closely related to our work. While CRM enforces compositional constraints on the classifier's output, CoInD applies these constraints in the data-generating space, where they originate. Our experimental evaluation directly borrows baseline results from Mahajan et al. (2024).

Synthetic data for improving downstream models has been discouraged due to the negative evidence from the modal collapse Alemohammad et al. (2024). This can be explained because maximum likelihood objective encourages memorization Kamb & Ganguli (2024). Consequently, no new information is provided for classifiers to improve performance. While recent studies (Azizi et al., 2023; Tian et al., 2024; Chen et al., 2024) demonstrated performance improvements using pre-trained generative models for classification. Critically, these approaches often suffer from potential test information leakage, whereas our method achieves robust classifier performance without external information.

Compositionality in generative models Our work concerns compositional generalization in generative models, where the goal is to generate data with unseen attribute compositions. One class of approaches seek to achieve compositionality by combining distinct models trained for each attribute (Du et al., 2020; Liu et al., 2021; Nie et al., 2021; Du et al., 2023), which is expensive and scaling linearly with the number of attributes also suffer from incorrect marginals. In contrast, we are interested in monolithic compositional diffusion models that learn compositionality. Liu et al. (2022) studied compositionality broadly and proposed methods to represent compositions in terms of marginal probabilities obtained through factorization of the joint distribution. However, these factorized sampling methods fail since the underlying generative model learns inaccurate marginals.

8 Conclusion

In this work, we demonstrate that diffusion models trained on partial data, where not all attribute compositions are observed, fail to respect underlying conditional independence relations, thus compromising their ability to exhibit compositionality. To address this issue, we propose a novel synthetic data generation algorithm, CoInD, which enhances compositional capabilities and allows for enforcing constraints in the data generation process. This approach leads to improved synthetic data quality and enables fine-grained control over the generation process.

The capabilities of CoInD naturally translate to enhanced robustness of downstream classifiers trained on synthetic data from CoInD against compositional shift. For a more comprehensive discussion and analysis of CoInD, including 2D Gaussian experiments App. E.1, extension to flow-based models App. E.2, and its limitations App. C.4, readers are directed to the appendix.

REFERENCES

Faruk Ahmed, Yoshua Bengio, Harm Van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-Consuming Generative Models Go MAD. In *International Conference on Learning Representations*, 2024.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023.

Marco Baroni and Brenden M Lake. Human-like systematic generalization through a meta-learning neural network. *Nature. 2023 Nov; 623 (7985): 115–21.*, 2023.

Changjian Chen, Fei Lv, Yalong Guan, Pengcheng Wang, Shengjie Yu, Yifan Zhang, and Zhuo Tang. Human-guided image generation for expanding small-scale training image datasets. *arXiv preprint arXiv:2412.16839*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional Visual Generation with Energy Based Models. In *Advances in Neural Information Processing Systems*, 2020.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, Reuse, Recycle: Compositional Generation with Energy-Based Diffusion Models and MCMC. In *International Conference on Machine Learning*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Peter J Hammond and Yeneng Sun. The essential equivalence of pairwise and mutual conditional independence. *Probability Theory and Related Fields*, 135(3):415–427, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022. URL `https://arxiv.org/abs/2207.12598`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, 2020a.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, 2020b.

Aapo Hyvärinen. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.

Dohyun Kwon, Ying Fan, and Kangwook Lee. Score-based Generative Modeling Secretly Minimizes the Wasserstein Distance. In *Advances in Neural Information Processing Systems*, 2022.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying Guidance in a Limited Interval Improves Sample and Distribution Quality in Diffusion Models. In *Advances in Neural Information Processing Systems*, 2024.

Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your Diffusion Model is Secretly a Zero-Shot Classifier. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Alexander Cong Li, Ananya Kumar, and Deepak Pathak. Generative classifiers avoid shortcut solutions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oCUYc7BzXQ.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow Matching Guide and Code, 2024. URL https://arxiv.org/abs/2412.06264.

Nan Liu, Shuang Li, Yilun Du, Josh Tenenbaum, and Antonio Torralba. Learning to Compose Visual Relations. In *Advances in Neural Information Processing Systems*, 2021.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional Visual Generation with Composable Diffusion Models. In *European Conference on Computer Vision*, 2022.

Sheng Liu, Xu Zhang, Nitesh Sekhar, Yue Wu, Prateek Singhal, and Carlos Fernandez-Granda. Avoiding spurious correlations via logit correction, february 2023. *URL http://arxiv.org/abs/2212.01433*, 2023.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *IEEE/CVF International Conference on Computer Vision*, 2015.

Divyat Mahajan, Mohammad Pezeshki, Ioannis Mitliagkas, Kartik Ahuja, and Pascal Vincent. Compositional risk minimization. *arXiv preprint arXiv:2410.06303*, 2024.

Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and Compositional Generation with Latent-Space Energy-Based Models. In *Advances in Neural Information Processing Systems*, 2021.

Maxwell Nye, Armando Solar-Lezama, Josh Tenenbaum, and Brenden M Lake. Learning Compositional Rules via Neural Program Synthesis . In *Advances in Neural Information Processing Systems*, 2020.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Pablo Sánchez-Moreno, Alejandro Zarzo, and Jesús S Dehesa. Jensen divergence based on Fisher's information. *Journal of Physics A: Mathematical and Theoretical*, 45(12):125305, 2012.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021a.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021b.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. StableRep: Synthetic Images from Text-to-Image Models Make Strong Visual Representation Learners. In *Advances in Neural Information Processing Systems*, 2024.

Christos Tsirigotis, Joao Monteiro, Pau Rodriguez, David Vazquez, and Aaron C Courville. Group robust classification without any group information. *Advances in Neural Information Processing Systems*, 36, 2024.

Max Welling and Yee W Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *International Conference on Machine Learning*, 2011.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems*, 2024.

Appendix

Table of Contents

A PRELIMINARIES OF SCORE-BASED MODELS

Score-based models Score-based models (Song et al., 2021b) learn the score of the observed data distribution, $P_{\text{train}}(\boldsymbol{X})$ through score matching (Hyvärinen, 2005). The score function $s_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})$ is learned by a neural network parameterized by $\boldsymbol{\theta}$.

$$L_{\text{score}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{train}}}\left[\|s_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{train}}(\mathbf{x})\|_2^2\right] \tag{9}$$

During inference, sampling is performed using Langevin dynamics:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\eta}{2}\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}) + \sqrt{\eta}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1) \tag{10}$$

where $\eta > 0$ is the step size. As $\eta \to 0$ and $T \to \infty$, the samples $\mathbf{x}_t$ converge to $p_{\boldsymbol{\theta}}(\boldsymbol{X})$ under certain regularity conditions (Welling & Teh, 2011).

Diffusion models Song & Ermon (2019) proposed a scalable variant that involves adding noise to the data Ho et al. (2020b) has shown its equivalence to Diffusion models. Diffusion models are trained by adding noise to the image $\mathbf{x}$ according to a noise schedule, and then neural network, ϵ_θ is used to predict the noise from the noisy image, $\mathbf{x_t}$. The training objective of the diffusion models is given by:

$$L_{\text{score}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{train}}}\mathbb{E}_{t\sim[0,T]}\left\|\epsilon - \epsilon_\theta\left(\mathbf{x}_t, t\right)\right\|^2 \tag{11}$$

Here, the perturbed data $\mathbf{x}_t$ is expressed as: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x} + \sqrt{1-\bar{\alpha}_t}\epsilon$ where $\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$, for a pre-specified noise schedule α_t. The score can be obtained using,

$$s_\theta(\mathbf{x}_t, t) \approx -\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}} \tag{12}$$

Langevin dynamics can be used to sample from the $s_\theta(\mathbf{x}_t, t)$ to generate samples from $p(\boldsymbol{X})$. The conditional score (Dhariwal & Nichol, 2021) is used to obtain samples from the conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C)$ as:

$$\nabla_{\boldsymbol{X}_t} \log p(\boldsymbol{X}_t \mid C) = \underbrace{\nabla_{\boldsymbol{X}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}_t)}_{\text{Unconditional score}} + \gamma \nabla_{\boldsymbol{X}_t} \log \underbrace{p_{\boldsymbol{\theta}}(C \mid \boldsymbol{X}_t)}_{\text{noisy classifier}}$$

where γ is the classifier strength. Instead of training a separate noisy classifier, Ho & Salimans have extended to conditional generation by training $\nabla_{\boldsymbol{X}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}_t \mid C) = s_\theta(\boldsymbol{X}_t, t, C)$. The sampling can be performed using the following equation:

$$\nabla_{\boldsymbol{X}_t} \log p(\boldsymbol{X}_t \mid C) = (1-\gamma)\nabla_{\boldsymbol{X}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}_t) + \gamma \nabla_{\boldsymbol{X}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}_t \mid C) \tag{13}$$

However, the sampling needs access to unconditional scores as well. Instead of modelling $\nabla_{\boldsymbol{X}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}_t)$, $\nabla_{\boldsymbol{X}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}_t|C)$ as two different models Ho & Salimans have amortize training a separate classifier training a conditional model $s_\theta(\mathbf{x}_t, t, \mathbf{c})$ jointly with unconditional model trained by setting $c = \varnothing$.

In the general case of classifier-free guidance, a single model can be effectively trained to accommodate all subsets of attribute distributions. During the training phase, each attribute c_i is randomly set to $\varnothing$ with a probability p_{uncond}. This approach ensures that the model learns to match all possible subsets of attribute distributions. Essentially, through this formulation, we use the same network to model all the possible subsets of conditional probability.

Once trained, the model can generate samples conditioned on specific attributes, such as c_i and c_j, by setting all other conditions to $\varnothing$. The conditional score is then computed as, $\nabla_{\boldsymbol{X}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}_t|c_i, c_j) = s_\theta(\mathbf{x}_t, \mathbf{c}^{i,j})$, where $\mathbf{c}^{i,j}$ represents the condition vector with all values other than i and j set to $\varnothing$. This method allows for flexible and efficient sampling across various attribute combinations.

Estimating Guidance Once the diffusion model is trained, we investigate the implicit classifier, $p_\theta(C|\boldsymbol{X})$, learned by the model. This will give us insights into the learning process of the diffusion models. (Li et al., 2023) have shown a way to calculate $p_\theta(C_i = c_i \mid \boldsymbol{X} = \mathbf{x})$, borrowing equation (5), (6) from (Li et al., 2023).

$$p_\theta(C_i = c_i \mid \mathbf{x}) = \frac{p(c_i)\, p_\theta(\mathbf{x} \mid c_i)}{\sum_k p(c_k)\, p_\theta(\mathbf{x} \mid c_k)}$$

$$p_\theta(C_i = c_i \mid \mathbf{x}) = \frac{\exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}^i)\|^2]\}}{\mathbb{E}_{C_i}\left[\exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}^i)\|^2]\}\right]} \tag{14}$$

Likewise, we can extend it to joint distribution by

$$p_\theta(C_i = c_i, C_j = c_j \mid \mathbf{x}) = \frac{\exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}^{i,j})\|^2]\}}{\mathbb{E}_{C_i,C_j}\left[\exp\{-\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}^{i,j})\|^2]\}\right]} \tag{15}$$

Practical Implementation The authors Li et al.. have showed many axproximations to compute $\mathbb{E}_{t,\epsilon}$. However, we use a different approximation inspired by Kynkäänniemi et al. (2024), where we sample 5 time-steps between [300,600] instead of these time-steps spread over the [0, T].

A.1 Computing JSD

We are interested in understanding the causal structure learned by diffusion models. Specifically, we aim to determine whether the learned model captures the conditional independence between attributes, allowing them to vary independently. This raises the question: *Do diffusion models learn the conditional independence between attributes?* The conditional independence is defined by:

$$p_{\boldsymbol{\theta}}(C_i, C_j \mid \boldsymbol{X}) = p_{\boldsymbol{\theta}}(C_i \mid \boldsymbol{X}) p_{\boldsymbol{\theta}}(C_j \mid \boldsymbol{X}) \tag{16}$$

We aim to measure the violation of this equality using the Jensen-Shannon divergence (JSD) to quantify the divergence between two probability distributions:

$$\text{JSD} = \mathbb{E}_{p_{\text{data}}}\left[D_{\text{JS}}\left(p_{\boldsymbol{\theta}}(C \mid \boldsymbol{X}) \;||\; p_{\boldsymbol{\theta}}(C_i \mid \boldsymbol{X}) p_{\boldsymbol{\theta}}(C_j \mid \boldsymbol{X})\right)\right] \tag{17}$$

The joint distribution, $p_{\boldsymbol{\theta}}(C_i, C_j \mid \boldsymbol{X})$, and the marginal distributions, $p_{\boldsymbol{\theta}}(C_i \mid \boldsymbol{X})$ and $p_{\boldsymbol{\theta}}(C_j \mid \boldsymbol{X})$, are evaluated at all possible values that C_i and C_j can take to obtain the probability mass function (pmf). The probability for each value is calculated using Equation Eq. (15) for the joint distribution and Equation Eq. (14) for the marginals.

Practical Implementation For the diffusion model with multiple attributes, the violation in conditional *mutual* independence should be calculated using all subset distributions. However, we focus on pairwise independence. We further approximate this in our experiments by computing JSD between the first two attributes, C_1 and C_2. We have observed that computing JSD between any attribute pair does not change our examples' conclusion.

A.2 Conformity Score (CS)

To measure the CS, we first infer attributes $(\hat{c}_1, \ldots, \hat{c}_n)$ from the generated images $\hat{\boldsymbol{X}}$ using attribute-specific classifiers ϕ_{C_i} and compare them against the expected attributes from the input composition $(c_1, \ldots, c_n)$. We refer to this accuracy as *conformity score* (CS) and is given by

$$\text{CS}(g) = \mathbb{E}_{p(C)p(U)}\left[\prod_i^n \mathbb{1}(C_i, \phi_{C_i}(g(C, U)))\right] \tag{18}$$

where $\mathbb{1}(\cdot, \cdot)$, g, and U are the indicator function, diffusion model, and the stochastic noise in the generation process respectively

To obtain a attribute-specific classifier, we train a single ResNet-18 (He et al., 2016) classifier with multiple classification heads, one corresponding to each attribute, and trained on the full support. The effectiveness of the classifier in predicting the attributes is reported in App. D.3

B Proofs for Claims

In this section, we detail the mathematical derivations for violation of conditional independence in diffusion models in App. B.1, and then derive the final loss function of CoInD in App. B.2.

B.1 STANDARD DIFFUSION MODEL OBJECTIVE IS NOT SUITABLE FOR COMPOSITIONALITY

This section proves that the violation in conditional independence in diffusion models is due to learning incorrect marginals, $p_{\text{train}}(\boldsymbol{X} \mid C_i)$ under $C_i \not\perp\!\!\!\perp C_j$. We leverage the causal invariance property: $p_{\text{train}}(\boldsymbol{X} \mid C) = p_{\text{true}}(\boldsymbol{X} \mid C)$, where p_{train} is the training distribution and p_{true} is the true underlying distribution.

Consider the training objective of the score-based models in classifier free formulation Eq. (9). For the classifier-free guidance, a single model $s_{\boldsymbol{\theta}}(\mathbf{x}, C)$ is effectively trained to match the score of all subsets of attribute distributions. Therefore, the effective formulation for classifier-free guidance can be written as,

$$L_{\text{score}} = \mathbb{E}_{\mathbf{x}\sim p_{\text{train}}}\mathbb{E}_S \left[\|\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x} \mid c_S) - \nabla_{\mathbf{x}} \log p_{\text{train}}(\mathbf{x} \mid c_S)\|_2^2 \right] \tag{19}$$

where S is the power set of attributes.

From the properties of Fisher divergence, $L_{\text{score}} = 0$ iff $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid c_S) = p_{\text{train}}(\boldsymbol{X} \mid c_S), \forall S$. In the case of marginals, $p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i)$ i.e. $S = \{C_i\}$ for some $1 \leq i \leq n$,

$$\begin{aligned}
p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i) &= p_{\text{train}}(\boldsymbol{X} \mid C_i) \\
&= \sum_{C_{-i}} p_{\text{train}}(\boldsymbol{X} \mid C_i, C_{-i}) p_{\text{train}}(C_{-i} \mid C_i) \\
&= \sum_{C_{-i}} p_{\text{true}}(\boldsymbol{X} \mid C_i, C_{-i}) p_{\text{train}}(C_{-i} \mid C_i) \\
&\neq \sum_{C_{-i}} p_{\text{true}}(\boldsymbol{X} \mid C_i, C_{-i}) p_{\text{true}}(C_{-i}) = p_{\text{true}}(\boldsymbol{X} \mid C_i) \\
\implies p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i) &\neq p_{\text{true}}(\boldsymbol{X} \mid C_i)
\end{aligned} \tag{20}$$

Where $C_{-i} = \prod_{\substack{j=1 \\ j\neq i}}^{n} C_j$, which is every attribute except C_i. Therefore, the objective of the score-based models is to maximize the likelihood of the marginals of training data and not the true marginal distribution, which is different from the training distribution when $C_i \not\perp\!\!\!\perp C_j$.

B.2 STEP-BY-STEP DERIVATION OF COIND IN § 4

The objective is to train the model by explicitly modeling the joint likelihood following the causal factorization. The minimization for this objective can be written as,

$$\mathcal{L}_{\text{comp}} = \mathcal{W}_2 \left(p(\boldsymbol{X} \mid C), \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X})}{p_{\boldsymbol{\theta}}(C)} \prod_i \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i) p_{\boldsymbol{\theta}}(C_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{X})} \right) \tag{21}$$

where $\mathcal{W}_2$ is 2-Wasserstein distance. Applying the triangle inequality to Eq. (21) we have,

$$\mathcal{L}_{\text{comp}} \leq \underbrace{\mathcal{W}_2 \left(p(\boldsymbol{X} \mid C), p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C) \right)}_{\text{Distribution matching}} + \underbrace{\mathcal{W}_2 \left(p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C), \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X})}{p_{\boldsymbol{\theta}}(C)} \prod_i^n \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i) p_{\boldsymbol{\theta}}(C_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{X})} \right)}_{\text{Conditional Independence}} \tag{22}$$

(Kwon et al., 2022) showed that under some conditions, the Wasserstein distance between $p_0(\boldsymbol{X}), q_0(\boldsymbol{X})$ is upper bounded by the square root of the score-matching objective. Rewriting Equation 16 from (Kwon et al., 2022)

$$\mathcal{W}_2 \left(p_0(\boldsymbol{X}), q_0(\boldsymbol{X}) \right) \leq K \sqrt{\mathbb{E}_{p_0(\boldsymbol{X})} \left[||\nabla_{\boldsymbol{X}} \log p_0(\boldsymbol{X}) - \nabla_{\boldsymbol{X}} \log q_0(\boldsymbol{X})||_2^2 \right]} \tag{23}$$

Distribution matching Following Eq. (23) result, the first term in Eq. (22), replacing p_0 as p and q_0 as p_θ will result in

$$\begin{aligned}
\mathcal{W}_2 \left(p(\boldsymbol{X} \mid C), p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C) \right) &\leq K_1 \sqrt{\mathbb{E}_{p_0(\boldsymbol{X})} \left[||\nabla_{\boldsymbol{X}} \log p(\boldsymbol{X} \mid C) - \nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X})||_2^2 \right]} \\
&= K_1 \sqrt{\mathcal{L}_{\text{score}}}
\end{aligned} \tag{24}$$

Conditional Independence Following Eq. (23) result, the second term in Eq. (22), replacing p_0 as p_θ and $q_0(\boldsymbol{X})$ as $\frac{p_{\boldsymbol{\theta}}(\boldsymbol{X})}{p_{\boldsymbol{\theta}}(C)} \prod_i^n \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X}|C_i)p_{\boldsymbol{\theta}}(C_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{X})}$

$$\mathcal{W}_2 \left(p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C), \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X})}{p_{\boldsymbol{\theta}}(C)} \prod_i^n \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i)p_{\boldsymbol{\theta}}(C_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{X})} \right)$$

$$\leq \sqrt{\mathbb{E} \| \nabla_{\mathbf{X}} \log p_\theta(\boldsymbol{X} \mid C) - \nabla_{\mathbf{X}} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X})}{p_{\boldsymbol{\theta}}(C)} \prod_i^n \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i)p_{\boldsymbol{\theta}}(C_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{X})} \|_2^2}$$

Further simplifying and incorporating $\nabla_{\mathbf{X}} \log p_\theta(C_i) = 0$ and $\nabla_{\mathbf{X}} \log p_\theta(C) = 0$ will result in

$$\mathcal{W}_2 \left(p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C), \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X})}{p_{\boldsymbol{\theta}}(C)} \prod_i^n \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X} \mid C_i)p_{\boldsymbol{\theta}}(C_i)}{p_{\boldsymbol{\theta}}(\boldsymbol{X})} \right)$$

$$\leq K_2 \sqrt{\underbrace{\mathbb{E} \| \nabla_{\mathbf{X}} \log p_\theta(\boldsymbol{X} \mid C) - \nabla_{\mathbf{X}} \log p_\theta(\boldsymbol{X}) - \sum_i \left[\nabla_{\mathbf{X}} \log p_\theta(\boldsymbol{X} \mid C_i) - \nabla_{\mathbf{X}} \log p_\theta(\boldsymbol{X}) \right] \|_2^2}_{\mathcal{L}_{\text{CI}}}}$$

$$= K_2 \sqrt{\mathcal{L}_{\text{CI}}} \tag{25}$$

Substituting Eq. (24), Eq. (25) in Eq. (22) will result in our final learning objective

$$\mathcal{L}_{\text{comp}} \leq K_1 \sqrt{\mathcal{L}_{\text{score}}} + K_2 \sqrt{\mathcal{L}_{\text{CI}}} \tag{26}$$

where K_1, K_2 are positive constants, i.e., the conditional independence objective $\mathcal{L}_{\text{CI}}$ is incorporated alongside the existing score-matching loss $\mathcal{L}_{\text{score}}$.

Note that Eq. (25) is the Fisher divergence between the joint $p_\theta(\boldsymbol{X} \mid C)$ and the causal factorization $\frac{p_\theta(X)}{p_\theta(C)} \prod_i \frac{p_\theta(\boldsymbol{X}|C_i)p_\theta(C_i)}{p_\theta(\boldsymbol{X})}$. From the properties of Fisher divergence (Sánchez-Moreno et al., 2012), $\mathcal{L}_{\text{CI}} = 0$ iff $p_\theta(\boldsymbol{X} \mid C) = \frac{p_\theta(\boldsymbol{X})}{p_\theta(C)} \prod_i^n \frac{p_\theta(\boldsymbol{X}|C_i)p_\theta(C_i)}{p_\theta(\boldsymbol{X})}$ and further implying, $\prod_i p_\theta(C_i \mid \boldsymbol{X}) = p_{\text{train}}(C \mid \boldsymbol{X})$

When $L_{\text{comp}} = 0$: $P_\theta(\boldsymbol{X} \mid C) = P_{\text{train}}(\boldsymbol{X} \mid C) = P(\boldsymbol{X} \mid C)$, and $\prod_i p_\theta(C_i \mid \boldsymbol{X}) = p_{\text{train}}(C \mid \boldsymbol{X})$. This implies that the learned marginals obey the causal independence relations from the data-generation process, leading to more accurate marginals.

B.3 Connection to compositional generation from first principles

Compositional generation from first principles Wiedemer et al. (2024) have shown that restricting the function to a certain compositional form will perform better than a single large model. In this section, we show that, by enforcing conditional independence, we restrict the function to encourage compositionality.

Let $c_1, c_2, \ldots, c_n$ be independent components such that $c_1, c_2, \ldots, c_n \in \mathbb{R}$. Consider an injective function $f : \mathbb{R}^n \to \mathbb{R}^d$ defined by $f(c) = x$. If the components, c are conditionally independent given x the cumulative functions, F must satisfy the following constraint:

$$F_{C_i, C_j, \ldots, C_n | X=x}(c_i, c_j, \ldots, c_n) = \prod_i F_{C_i | X=x}(c_i) \tag{27}$$

$F^{-1}_{C_i, C_j, \ldots, C_n | X=x}(x) = \inf\{c_i, c_j, \ldots, c_n \mid F(c_i, c_j, \ldots, c_n) \geq x\}$, where $F^{-1}_{c_i, c_j, \ldots, C_n | X=x}$ is a generalized inverse distribution function.

$$\begin{aligned} f(c_i, c_j, \ldots, c_n) &= (f \circ F^{-1}_{c_i, c_j, \ldots, C_n | X=x})(\prod_i F_{C_i | X=x}(c_i)) \\ &= (f \circ F^{-1}_{c_i, c_j, \ldots, C_n | X=x} \circ e)(\sum_i \log F_{C_i | X=x}(c_i)) \\ &= g(\sum_i \phi_i(c_i)) \end{aligned}$$

Therefore, we are restricting f to take a certain functional form. However, it is difficult to show that the data generating process, f, meets the rank condition on the Jacobian for the sufficient support assumption Wiedemer et al. (2024), which is also the limitation discussed in their approach. Therefore, we cannot provide guarantees. However, this section provides a functional perspective of CoInD.

C Practical Considerations

To facilitate scalability and numerical stability for optimization, we introduce two approximations to the upper bound of our objective function Eq. (5).

C.1 Scalability of $\mathcal{L}_{\text{CI}}$

A key computational challenge posed by Eq. (7) is that the number of model evaluations grows linearly with the number of attributes. The Eq. (7) is derived from conditional independence formulation as follows:

$$p_\theta(C \mid X) = \prod_i p_\theta(C_i \mid X). \tag{28}$$

By applying Bayes' theorem to all terms, we obtain,

$$\frac{p_\theta(\boldsymbol{X} \mid C) p_\theta(C)}{p_\theta(X)} = \prod_i \frac{p_\theta(\boldsymbol{X} \mid C_i) p_\theta(C_i)}{p_\theta(\boldsymbol{X})} \tag{29}$$

Note that this formulation is equal to the causal factorization. From this, by applying logarithm and differentiating w.r.t. $\boldsymbol{X}$, we derive the score formulation.

$$\nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X} \mid C) = \nabla_{\boldsymbol{X}} \log \sum_i p_\theta(\boldsymbol{X} \mid C_i) - \nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X}) \tag{30}$$

The L_2 norm of the difference between LHS and RHS of the objective in Eq. (30) is given by, which forms our $\mathcal{L}_{\text{CI}}$ objective.

$$\mathcal{L}_{\text{CI}} = \|\nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X} \mid C) - \left(\nabla_{\boldsymbol{X}} \log \sum_i p_\theta(\boldsymbol{X} \mid C_i) - \nabla_{\boldsymbol{X}} \log p_\theta(\boldsymbol{X}) \right) \|_2^2 \tag{31}$$

Due to the $\sum_i$, in the equation, the number of model evaluations grows linearly with the number of attributes (n). This $\mathcal{O}(n)$ computational complexity hinders the approach's applicability at scale. To address this, we leverage the results of (Hammond & Sun, 2006), which shows conditional independence is equivalent to pairwise independence under large n to reduce the complexity to $\mathcal{O}(1)$ in expectation. This allows for a significant improvement in scalability while maintaining computational efficiency. Using this result, we modify Eq. (28) to:

$$p_\theta(C_i, C_j \mid X) = p_\theta(C_i \mid X) p_\theta(C_j \mid X). \quad \forall i, j$$

Accordingly, we can simplify the loss function for conditional independence as follows:

$$\mathcal{L}_{\text{CI}} = \mathbb{E}_{p(\boldsymbol{X}, C)} \mathbb{E}_{j,k} \|\nabla_X [\log p_\theta(X|C_j, C_k) - \log p_\theta(X|C_j) - \log p_\theta(X|C_k) + \log p_\theta(X)]\|_2^2. \tag{32}$$

In score-based models, which are typically neural networks, the final objective is given as:

$$\mathcal{L}_{\text{CI}} = \mathbb{E}_{p(\boldsymbol{X}, C)} \mathbb{E}_{j,k} \| s_{\boldsymbol{\theta}}(\boldsymbol{X}, C_j, C_k) - s_{\boldsymbol{\theta}}(\boldsymbol{X}, C_j) - s_{\boldsymbol{\theta}}(\boldsymbol{X}, C_k) + s_{\boldsymbol{\theta}}(\boldsymbol{X}, \varnothing) \|_2^2 \tag{33}$$

where $s_{\boldsymbol{\theta}}(\cdot) \coloneqq \nabla_{\boldsymbol{X}} \log p_\theta(\cdot)$ is the score of the distribution modeled by the neural network. We leverage classifier-free guidance to train the conditional score $s_{\boldsymbol{\theta}}(\boldsymbol{X}, C_i)$ by setting $C_k = \varnothing$ for all $k \neq i$, and likewise for $s_{\boldsymbol{\theta}}(\boldsymbol{X}, C_i, C_j)$, we set $C_k = \varnothing$ for all $k \notin \{i, j\}$.

C.2 Simplification of Theoretical Loss

In Eq. (5), we showed that the 2-Wassertein distance between the true joint distribution $p(\boldsymbol{X} \mid C)$ and the causal factorization in terms of the marginals $p(\boldsymbol{X} \mid C_i)$ is upper bounded by the weighted sum of the square roots of $\mathcal{L}_{\text{score}}$ and $\mathcal{L}_{\text{CI}}$ as $\mathcal{L}_{\text{comp}} \leq K_1\sqrt{\mathcal{L}_{\text{score}}} + K_2\sqrt{\mathcal{L}_{\text{CI}}}$. In practice, however, we minimized a simple weighted sum of $\mathcal{L}_{\text{score}}$ and $\mathcal{L}_{\text{CI}}$, given by $\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{score}} + \lambda\mathcal{L}_{\text{CI}}$ as shown in Eq. (8) instead of Eq. (5). We used Eq. (8) to avoid the instability caused by larger gradient magnitudes (due to the square root). Eq. (8) also provided the following practical advantages: **(1)** the simplicity of the loss function that made hyperparameter tuning easier, and **(2)** the similarity of Eq. (8) to the loss functions of pre-trained diffusion models allowing us to reuse existing hyperparameter settings from these models.

C.3 Choice of Hyperparameter λ

The value for the hyperparameter λ is chosen such that the gradients from the score-matching objective L_{score} and the conditional independence objective L_{CI} are balanced in magnitude. One way to choose λ is by training a vanilla diffusion model and setting $\lambda = \frac{L_{score}}{L_{CI}}$. As a rule of thumb, we recommend the simplified setting: $\lambda = L_{score} \times 4000$.

C.4 Limitations

This paper considered compositions of a closed set of attributes. As such, CoInD requires predefined attributes and access to data labeled with the corresponding attributes. Moreover, CoInD must be enforced during training, which requires retraining the model whenever the attribute space changes to include additional values. Instead, state-of-the-art generative models seek to operate without pre-defined attributes or labeled data and generate open-set compositions. Despite the seemingly restricted setting of our work, our findings provide valuable insights into a critical limitation of current generative models, namely their failure to generalize for unseen compositions, by identifying the source of this limitation and proposing an effective solution to mitigate it.

D Experiment Details

D.1 CoInD Algorithm

Algorithm 1 CoInD Training

repeat
  $(\mathbf{c}, \mathbf{x}_0) \sim p_{\text{train}}(\mathbf{c}, x)$
  $c_k \leftarrow \varnothing$ with probability p_{uncond} ▷ Set element of index,k i.e, c_k to $\varnothing$ with $p_{uncond} \forall k \in [0, N]$ probability
  $i \sim \text{Uniform}(\{0, \ldots, N\}), j \sim \text{Uniform}(\{0, \ldots, N\} \setminus \{i\})$ ▷ Select two random attribute indices
  $t \sim \text{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $x_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
  $\mathbf{c}^i, \mathbf{c}^j, \mathbf{c}^{i,j} \leftarrow \mathbf{c}$
  $\mathbf{c}^i \leftarrow \{c_k = \varnothing \mid k \neq i\}, \mathbf{c}^j \leftarrow \{c_k = \varnothing \mid k \neq j\}, \mathbf{c}^{i,j} \leftarrow \{c_k = \varnothing \mid k \notin \{i, j\}\}, \mathbf{c}^{\varnothing} \leftarrow \varnothing$
  $L_{CI} = ||\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^i) + \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^j) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^{i,j}) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^{\varnothing})||_2^2$
  Take gradient descent step one
    $\nabla_\theta[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})\|^2 + \lambda L_{CI}]$
until converged

To compute pairwise independence in a scalable fashion, we randomly select two attributes, i and j, for a sample in the batch and enforce independence between them. As the score in Eq. (12) is given by $\frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}$. The final equation for enforcing $\mathcal{L}_{\text{CI}}$ will be:

$$L_{CI} = \frac{1}{1-\bar{\alpha}_t} \left\| \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^i) + \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^j) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^{i,j}) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}^{\varnothing}) \right\|_2^2$$

We follow Ho et al. (2020b) to weight the term by $1-\bar{\alpha}_t$. This results in an algorithm for CoInD, requiring only a few modifications of lines from (Ho & Salimans, 2022), highlighted below. **Practical Implementation** In our experiments, we have used $p_{\text{uncond}} = 0.3$

D.2 Training details, Architecture, and Sampling

To generate CelebA images, we scale the image size to 64×64. We use the latent encoder of Stable Diffusion 3 (SD3) (Esser et al., 2024) to encode the images to a latent space and perform diffusion in the latent space. The model uses an AdamW optimizer with a learning rate of 1.0×10^{-4} and trains for 500,000 steps. It employs a DDPM train noise scheduler with a cosine noise schedule and 1000 train noise steps. The architecture is based on a U-Net model with 2 layers per block and, with block out channels of [224, 448, 672, 896]. It has an attention head dimension of 8, and 8 norm groups. trained on a A6000 GPU.

Sampling

To generate samples for a composition, we sample from the referenced equation Eq. (2) using $\gamma = 0.46$ to achieve more pronounced gender features, employing the Denoising Diffusion Implicit Models (DDIM) (Song et al., 2021a) for 250 steps. We apply identical sampling settings across both standard diffusion models and our proposed method. This configuration was selected because it consistently produces high-fidelity samples, demonstrating robust performance across both model architectures. By maintaining consistent sampling parameters, we ensure a fair and comparable evaluation of the generative capabilities of CoInD.

Evaluation : We evaluate our method by computing the Fréchet Inception Distance (FID). Specifically, we sample 5,000 generated images of unseen blond females following method described above and compare them against a reference set of celebA blond female images.

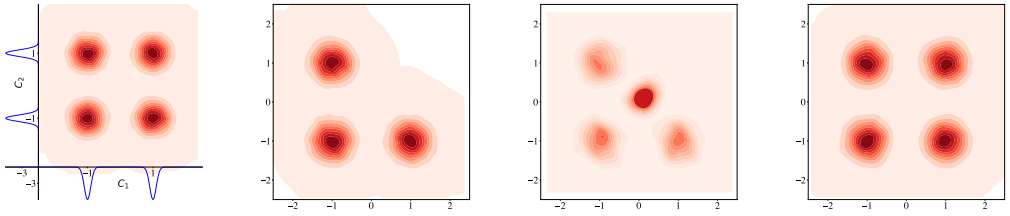
D.3 Accuracy of classifiers for Conformity Score (CS)

Table 5: ResNet-18 accuracy of classifying attributes on CelebA Dataset

Feature	Attributes	Possible Values	Accuracy
C_1	Blond	0,1	95.1
C_2	color	0,1	98.0

E Discussion

E.1 2D Gaussian: Workings of CoInD in closed form



(a) True underlying data distribution (b) Training data Orthogonal Support (c) Conditional distribution learned by vanilla diffusion objective (d) Conditional distribution learned by CoInD

Figure 5: CoInD respects underlying independence conditions thereby generating true data distribution (d).

The underlying data is generated by two independent attributes, C_1 and C_2. The observed variable $\mathbf{X}$ is defined as:

$$\mathbf{X} = f(\mathbf{C}_1) + f(\mathbf{C}_2) \tag{34}$$

where $f(c_i) = c_i + \sigma\epsilon$, and $\epsilon \sim \mathcal{N}(0, I)$ represents Gaussian noise. For simplicity, C_1 and C_2 are binary variables taking values in $\{-1, +1\}$. The function $f(C_1)$ results in a mixture of Gaussians with means $[-1 \quad 0]$ and $[+1 \quad 0]$. These are represented along the x-axis in Figure 5a. Similarly, $f(C_2)$ produces a mixture of Gaussians with means:$[0 \quad -1]$ and $[0 \quad +1]$. These are displayed along the y-axis in Figure 5a. The combination of C_1 and C_2 independently generates as Eq. (34) This results in a two-dimensional Gaussian mixture, as illustrated in Figure 5. We consider orthogonal support, where attribute combinations of $(C_1, C_2) \in \{(-1, -1), (-1, +1), (+1, -1)\}$, and the model is tasked to generate unseen combination of $(+1, +1)$. Also as a reminder that assumptions mentioned in § 2 are satisfied. (1) C_1, C_2 independently generate $\boldsymbol{X}$, and (2) all possibles values for every attribute are present at-least observed during training. Let score is given by $s_{+1,+1}$ represents $\nabla_{\mathbf{X}} \log p(X \mid C_1 = 1, C_2 = 1)$ and likewise $s_{1,\varnothing}$ represents $\nabla_{\mathbf{X}} \log p(X \mid C_1)$ To sample for the unseen compositions of (1,1) we use Eq. (1) to

$$s_{1,1}(x) = s_{1,\varnothing}(x) + s_{\varnothing,1}(x) - s_{\varnothing,\varnothing}(x) \tag{35}$$

Training diffusion model (score) objective involves computing score functions from the training data, which will give us the following terms in closed form. For example $s_{1,\varnothing}(x)$ is training using only $+1, -1$ combinations present during training. which is nothing but a gaussian at $+1, -1$ and the score of the gaussian is given in closed form.

$$s_{1,\varnothing}(x) = \frac{\mu_{1,-1} - x}{\sigma^2}$$
$$s_{\varnothing,1}(x) = \frac{\mu_{1,-1} - x}{\sigma^2}$$

$s_{\varnothing,\varnothing}(x)$ is a mixture of gaussian with means around 3 Gaussians present during training. The score of the mixture gaussian as:

$$s_{\varnothing,\varnothing}(x) = \frac{\sum_i \mathcal{N}(x; \mu_i, \sigma^2 I) \left[\frac{\mu_i - x}{\sigma^2}\right]}{\sum_i \mathcal{N}(x; \mu_i, \sigma^2 I)}$$

Now leveraging Langevin dynamics Eq. (10) will generate the Fig. 5c as the distribution of $P(X \mid C_1 = +1, C_2 = +1)$ is incorrect (strong red blob between the $(+1, -1), (-1, +1)$ instead of gaussian at (+1,+1)). This is due to incorect modelling of the distributions $s_{1,\varnothing}(x), s_{\varnothing,1}(x), s_{\varnothing,\varnothing}(x)$. However, CoInD does not explicitly model $s_{1,\varnothing}$, instead learn joint $s_{-1,-1}(x), s_{+1,-1}(x), s_{-1,+1}(x)$ as Gaussians and then combine them using pairwise conditional independence conditions given as:

$$\begin{aligned}
s_{-1,-1}(x) &= s_{+1,\varnothing}(x) + s_{\varnothing,+1}(x) - s_{\varnothing,\varnothing}(x) \\
s_{+1,-1}(x) &= s_{+1,\varnothing}(x) + s_{\varnothing,-1}(x) - s_{\varnothing,\varnothing}(x) \\
s_{-1,+1}(x) &= s_{-1,\varnothing}(x) + s_{\varnothing,+1}(x) - s_{\varnothing,\varnothing}(x) \\
s_{+1,1}(x) &= s_{+1,\varnothing}(x) + s_{\varnothing,+1}(x) - s_{\varnothing,\varnothing}(x) \\
&= s_{+1,-1}(x) + s_{-1,+1}(x) - s_{-1,-1}(x) \\
&= \frac{[\mu_{+1,-1} + \mu_{+1,-1} - \mu_{-1,-1}] - x}{\sigma^2}
\end{aligned}$$

This shows the workings of CoInD and also demonstrates that conditional independence constraints are necessary to learn the underlying distribution and alos with these constraints, diffusion models generate incorrect interpolation for unseen data distributions as shown in Fig. 5c.

E.2 Extension to Gaussian source flow models

Diffusion models can be viewed as a specific case of flow-based models where: (1) the source distribution is Gaussian, and (2) the forward process follows a predetermined noise schedule.(Lipman et al., 2024). Can we reformulate CoInD in terms of velocity rather than score, thereby generalizing it to accommodate arbitrary source distributions and schedules? When the source distribution

is gaussian, score and velocity are related by affine transformation as detailed in Tab. 1 of (Lipman et al., 2024).

$$s_\theta^t(x, C_1, C_2) = a_t x + b_t u_\theta^t(x, C_1, C_2) \tag{36}$$

replacing $s_\theta^t(\cdot)$ into Eq. (33)

$$\begin{aligned}\mathcal{L}_{\text{CI}} &= \mathbb{E}_{p(\boldsymbol{X},C),t\sim U[0,1]}\mathbb{E}_{j,k}\|s_{\boldsymbol{\theta}}^t(x, C_j, C_k) - s_{\boldsymbol{\theta}}^t(x, C_j) - s_{\boldsymbol{\theta}}^t(x, C_k) + s_{\boldsymbol{\theta}}^t(x)\|_2^2 \\ &= \mathbb{E}_{p(\boldsymbol{X},C),t\sim U[0,1]}\mathbb{E}_{j,k}\left[b_t^2\|u_{\boldsymbol{\theta}}^t(x, C_j, C_k) - s_{\boldsymbol{\theta}}^t(x, C_j) - u_{\boldsymbol{\theta}}^t(x, C_k) + u_{\boldsymbol{\theta}}^t(x)\|_2^2\right]\end{aligned}$$

However we can ignore b_t^2, weighting for the time step t.

$$\mathcal{L}_{\text{CI}} = \mathbb{E}_{p(\boldsymbol{X},C),t\sim U[0,1]}\mathbb{E}_{j,k}\left[\|u_{\boldsymbol{\theta}}^t(x, C_j, C_k) - u_{\boldsymbol{\theta}}^t(x, C_j) - u_{\boldsymbol{\theta}}^t(x, C_k) + u_{\boldsymbol{\theta}}^t(x)\|_2^2\right] \tag{37}$$

Therefore, if the source distribution is gaussian and for any arbitrary noise schedule, constraint in score translates directly to velocity constraint as given as Eq. (37).

F QUALITATIVE SAMPLES

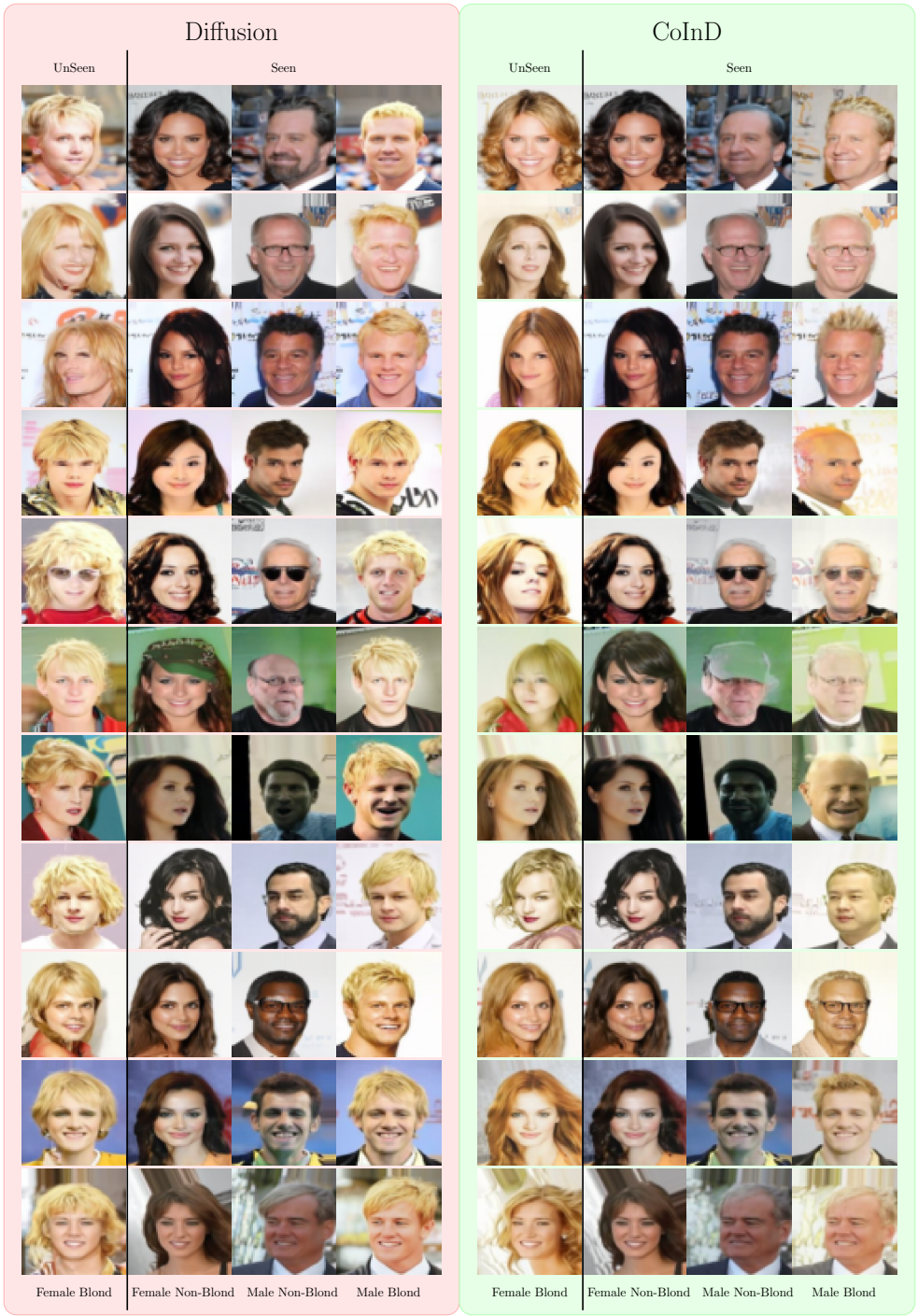


Figure 6: For unseen female blond generation, COIND (green box, Column 1) selectively modifies the hair color while preserving all other features from the reference sample (green box, Column 2). This behavior is achieved by enforcing conditional independence, ensuring that only the hair color is altered. In contrast, diffusion models (red box, Column 1) inadvertently leak gender and other extraneous attributes during generation, resulting in less accurate outputs.