# STRUCTURAL EXTRAPOLATIVE DATA GENERATION: HARDNESS, FEASIBILITY, AND FORMAL CONDITIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Achieving reliable extrapolation, specifically, extrapolative data generation beyond the support of the training distribution, is a fundamental challenge and inherently requires strong assumptions. This paper aims to address this problem and proposes a framework for Structural Extrapolative Data GEneration (SEDGE) based on suitable assumptions on the underlying data-generating process. We provide conditions under which data satisfying novel specifications can be generated reliably, together with the approximate identifiability of the distribution of such data under certain "conservative" assumptions, as well as the inherent non-identifiability of this distribution without such assumptions. On the algorithmic side, we develop two methods for extrapolative data generation: structure-informed likelihood-weighted resampling (LWR) and likelihood-guided diffusion (LGD). We verify the extrapolation performance on synthetic data and also consider extrapolative property-guided molecule design and extrapolative text-guided image generation as real-world scenarios to illustrate the validity of the proposed framework.

## 1 INTRODUCTION

Generative AI, designed to generate new content in the form of images, text, audio, or videos, has seen major breakthroughs and attracted much attention in recent years. For instance, in the image domain, well-known generative AI tools, including the latent diffusion model (Rombach et al., 2021), DALL-E 3 by OpenAI (Betker et al., 2023), Google's Imagen 3 (Imagen Team, Google, 2024), can generate high-quality images in a controllable manner.

However, in many real-world problems, we expect generative AI tools to be able to produce novel content and automate creative tasks in a reliable way. That is, they are required not only to interpolate within known regimes, but also to extrapolate beyond the support of available data. For instance, in AI-generated images for creative design, the desired content or visual specifications often lie outside the support of the training data. Similarly, in drug discovery, candidate molecules are optimized toward target properties to address new diseases or previously unseen combinations of diseases, for which no direct training examples exist. In such scenarios, the ability to generate reliable extrapolative data is essential, but as a classical problem in machine learning, especially in deep learning, it still remains a fundamental challenge.

Extrapolative data generation differs in theory from standard data augmentation (Shorten & Khoshgoftaar, 2019) or interpolation (estimating unknown data points within the range of a given dataset). Interpolation can be achieved, for instance, by making use of suitable smoothness assumptions on the function to be estimated from data; in contrast, extrapolation often requires a deeper understanding of the process behind the given data and applying it to novel scenarios. Otherwise, if one directly uses the learned flexible, deep data-generation model for extrapolation, the value of the input of the model is out of support of the training data, and the output of the model often does not have correctness guarantees. In fact, existing generative models, despite their expressive power, frequently fail when tasked with producing data that lie outside the training distribution (Gokhale et al., 2022; Luo et al., 2024; Vatsa et al., 2025).

Researchers have developed several types of assumptions or constraints to deal with the extrapolative generation issues in machine learning. A typical one is to directly constrain the functional class of the prediction function to be linear (or parametric) or with an additive form. Some other methods exploit compositional properties of generative functions to reduce instability during generation (Liu



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.

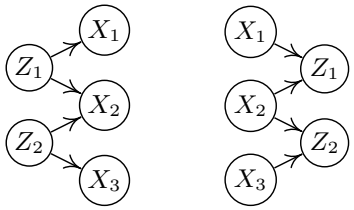
et al., 2022; Wiedemer et al., 2023; Lachapelle et al., 2023). Some work derives extrapolation guarantees from regularities on distribution shifts, for instance, smoothness (Kong et al., 2024). These developments do not provide a general framework for extrapolative data generation given novel specifications when properties of the involved functions are unknown.

In this paper, we study extrapolative data generation from a principled structural perspective, without directly imposing rather strong assumptions on the underlying functions. First, we provide an open analysis of two different data-generating processes, with graphical representation, to formulate the relationship between *specifications* and *features* of the data to be generated: modeling specifications as conditioning variables that generate features, as commonly adopted in conditional generation frameworks (as seen in Fig. 1(a), in which $Z_i$ and $X_i$ denote specifications and features, respectively), or modeling the process as features generating or satisfying specifications (see Fig. 1(b)). We show that the latter structure, in which features point to specifications in the graphical representation, offers a more natural explanation for unseen data. We further improve the graphical representation to explain why the novel scenarios do not appear in the training data, which also enables a principled treatment to go beyond such data. Interestingly, this structural view reveals a key source of difficulty in extrapolation: features directly involved in multiple specifications are inherently harder to extrapolate, since each specification imposes additional constraints on the admissible data. In contrast, in scenarios where such shared features are absent, extrapolation becomes substantially more straightforward.

Accordingly, the contributions of this work are summarized as follows. 1) We propose a structural view of the extrapolative data generation problem and conduct a systematic investigation to characterize the conditions under which valid extrapolated data can be generated and those that enable the whole distribution of the extrapolated data to be approximately constructed. 2) We develop two practical approaches to generate data for novel combinations of specifications: structure-informed likelihood-weighted resampling (LWR) and likelihood-guided diffusion (LGD). 3) We further demonstrate the effectiveness of the proposed framework on synthetic extrapolation tasks, extrapolative property-guided molecule design, and extrapolative text-guided image generation.

## 2 Initial Thoughts

For clarity, let us use a simple example to illustrate the setting and our idea. Suppose we aim to generate a plausible image of a peacock eating ice cream, even though no such image appears in the training set. For illustrative purposes, suppose the task involves only two specifications: "peacock" ($Z_1 = 1$) and "eating ice cream" ($Z_2 = 1$).[1] As an initial idea, one may adopt the graphical representation in Fig. 1(a) to describe the dependence relations, where features $\mathbf{X} = (X_1, X_2, X_3)$ are generated by specifications $\mathbf{Z} = (Z_1, Z_2)$. Here, we may think of $X_2$ as indicating the presence of a "beak". Under this model, given a novel combination of $\mathbf{Z}$ values, say, $(Z_1 = 1, Z_2 = 1)$, which never happened in the training data, unless strong assumptions are enforced on the functions (for instance, a parametrtic model is assumed for $p(\mathbf{x} \mid z_1, z_2)$), the distribution of feature $X_2$, which is adjacent to both $Z_1$ and $Z_2$, is *undefined*, and hence we cannot find the distribution or values of $\mathbf{X}$ to satisfy the novel combination of $\mathbf{Z}$. In other words, the model does not know how to generate the feature "beak" for the unseen combination "peacock eating ice cream", since this combination is not present in the training data. With this graphical representation, extrapolative data generation for novel specifications is impossible without additional assumptions on the functions mapping $\mathbf{Z}$ to $\mathbf{X}$, when $Z_i$ with novel value combinations directly influence the same features.




(a) Specification generates features. (b) Features produce specifications.

Figure 1: The two generating processes as initial thoughts.

In contrast, Fig. 1(b) offers a more plausible representation of the relationship between $\mathbf{X}$ and $\mathbf{Z}$, with arrows pointing from features to specifications. This structure implies the conditional independence $Z_1 \perp\!\!\!\perp Z_2 \,|\, X_2$. For example, upon observing a "beak" ($X_2 = 1$), inferring whether the object is "eating ice cream" ($Z_2 = 1$) does not depend on whether it is a "peacock" ($Z_1 = 1$) or a chicken. Such a conditional independence relationship is not captured by the graphical model in Fig. 1(a), which instead generally implies conditional dependence between $Z_1$ and $Z_2$ given $X_2$. However, if

[1] In practice, this example may involve finer-grained specifications, such as "eating" and "ice cream"; here, we combine them into a single macro specification.

one adopts the graphical representation in Fig. 1(b) for the data-generating process, why is there no such scenario in the training data? How can we go beyond the training data in a principled way?

Now we are ready to introduce our framework, which models the different distributions across the training data and the data to be generated, figures out their connections, and allows us to go from training data to the construction of new data satisfying novel combinations of the specifications.

## 3 Structural Extrapolative Data Generation

We now demonstrate how the structural properties can be leveraged for data generation under novel specifications. We first formalize the data generating process in Sec. 3.1, and then study the case of two specifications in Sec. 3.2 to illustrate our key ideas of Structural Extrapolative Data GEneration (SEDGE), including how to generate novel samples and recover the novel data distribution. Lastly, we present theoretical results, together with the involved assumptions, for the general setting with multiple specifications in Sec. 3.3.

### 3.1 Formulation

Let $\mathbf{X} = (X_1, \ldots, X_d)$ denote the feature variables to be generated, and let $\mathbf{Z} = (Z_1, \ldots, Z_n)$ denote the specification variables. Let $S$ be a selection variable indicating whether a sample is contained in the given dataset (Heckman, 1979). A sample is contained in the given dataset if and only if $S = 1$; otherwise, $S = 0$. The given training data correspond to the distribution $p(\mathbf{X} \mid S = 1)$. We assume that the variables $\mathbf{X}$, $\mathbf{Z}$, and $S$ form a Bayesian network with respect to a directed acyclic graph (DAG) $\mathcal{G}$ (Pearl, 1985; Koller & Friedman, 2009). That is, the Markov property is satisfied: each variable is conditionally independent of its non-descendants in $\mathcal{G}$ given its parents in $\mathcal{G}$. Consistent with the discussion in Sec. 2, we focus on the case when selection variable $S$ has no children, and its parents are restricted to the specifications $\mathbf{Z}$. Importantly, we assume that there are no edges among the specifications $\mathbf{Z}$, that is, they are conditionally independent given the features $\mathbf{X}$. As we show below, this condition enables reliable extrapolative data generation; moreover, if it is violated, one can construct a scenario in which such extrapolation is impossible, highlighting the fundamental difficulty of extrapolative data generation without this condition.

Given data points observed as samples from distribution $p(\mathbf{X} \mid S = 1)$, our goal is to generate samples from $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{z})$, where the combination of the specifications values $\mathbf{Z} = \mathbf{z}$ might be novel, i.e., unseen in the given/training data. Note that if the combination of their values already appears in the training data, then this conditional distribution is already implied by the training data, since we have $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{z}) = p(\mathbf{X} \mid \mathbf{Z} = \mathbf{z}, S = 1)$, implied by the conditional independence between $S$ and $\mathbf{X}$ given $\mathbf{Z}$ according to our structural assumptions discussed above.

Our central question is therefore: can we generate new data that satisfy novel combinations of specifications, and, if so, under what conditions and by what methods? Moreover, under what conditions can the corresponding conditional distributions of $\mathbf{X}$ be constructed?

In this section, we assume that both the features $\mathbf{X}$ and the specifications $\mathbf{Z}$ are given. Our framework, however, is general. For instance, the features $\mathbf{X}$ can represent variables from tabular data or latent representations learned from images, while the specifications $\mathbf{Z}$ can represent textual concepts extracted from text. For simplicity, we also assume that the specifications $\mathbf{Z}$ are binary, which is natural when they indicate the presence or absence of concepts. The framework extends naturally to continuous specifications, as demonstrated in our experiments. In practice, one may need to learn $\mathbf{X}$ and $\mathbf{Z}$ from data; their identifiability and stimation procedure are discussed in Appendix D.

**Notations.** We use $\mathbf{1}$ to denote an all-ones vector of appropriate dimension. To simplify notation, and without loss of generality, we treat $\mathbf{Z} = \mathbf{1}$ as the novel specification combination to which we aim to extrapolate in this section; the analysis extends straightforwardly to other specification values. We use uppercase letters to denote random variables and lowercase letters to denote their realizations. When the random variables are unambiguous from context, we omit variables in density functions, e.g., writing $p(\mathbf{x})$ instead of $p(\mathbf{X} = \mathbf{x})$.[2] We use $p^{\mathcal{D}}(\cdot)$ to denote the distribution implied by the given/training data, which satisfies $S = 1$, e.g., $p^{\mathcal{D}}(\mathbf{X} \mid \mathbf{Z} = \mathbf{1}) \coloneqq p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1}, S = 1)$. We occasionally abuse notation by treating a vector as the corresponding set of its entries when the

[2]We use uppercase $P$ to denote probabilities for discrete variables, even when conditioning on continuous variables, and lowercase $p$ to denote densities of continuous variables.

meaning is clear from context. We also write $[n] \coloneqq \{1, \ldots, n\}$ for integer $n$ and use $\propto$ to denote equality up to a positive multiplicative constant.

### 3.2 Extrapolation with Two Specifications

We start with the case of two specification variables, $\mathbf{Z} = (Z_1, Z_2)$, to illustrate our key idea of leveraging structural properties to generate data that satisfy novel combinations of specifications. In Sec. 3.3, we extend this setting to the general case of more than two specification variables which can also be continuous.

To be more precise about novel specifications, suppose we observe only $(Z_1 = 1, Z_2 = 0)$ and $(Z_1 = 0, Z_2 = 1)$, and possibly $(Z_1 = 0, Z_2 = 0)$, in the training data, and aim to generate sensible data $\mathbf{X}$ for $(Z_1 = 1, Z_2 = 1)$, i.e., $P(Z_1 = 1, Z_2 = 1 \mid S = 1) = 0$. The novel combination has positive probability (and thus is feasible) in the underlying population, i.e., $P(Z_1 = 1, Z_2 = 1) > 0$. If either specification value, $Z_1 = 1$ or $Z_2 = 1$, never appears in the training data, i.e., $P(Z_i = 1 \mid S = 1) = 0$ for some $i \in 1, 2$, it is impossible to generate the corresponding samples, since $p(\mathbf{x} \mid Z_i = 1)$ cannot be identified from the training data. It is thus necessary for each individual specification value to appear in the training data. This is formalized in the following assumption.

**Assumption 1** (Basic conditions for specifications and selection)**.** *We have*

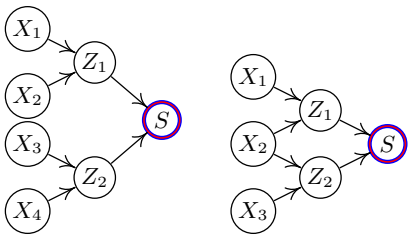
$$P(S = 1) > 0, \quad P(Z_1 = 1, Z_2 = 1) > 0, \quad \textit{and } P(Z_i = 1 \mid S = 1) > 0, \;\; i = 1, 2. \tag{1}$$

#### 3.2.1 Identifiable Case: Specifications do not Share Common Features

Let us first consider the generating process in Fig. 2(a). in which the specifications $Z_1$ and $Z_2$ do not share common features. It is worth noting the framework also applies when the features generating the same specification variable, such as $X_1$ and $X_2$, which generate $Z_1$, have dependence relations in between. In this case, one can successfully recover the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ from the given data, as seen below.

**Proposition 1.** *Suppose that Assumption 1 holds and the data generating process follows Fig. 2(a). Then, the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is identifiable from the given data.*

A proof is given in Appendix C.1. The idea is to make use of specific conditional independence relations implied by the structural properties of Fig. 2(a), where the specifications do not share common features, i.e., $Z_1 \perp\!\!\!\perp Z_2$.




(a) The generating process where $Z_1$ and $Z_2$ do not share common features.

(b) The generating process where $Z_1$ and $Z_2$ share $X_2$ as a common parent.

Figure 2: The generating processes where features generate the specification and the observed data satisfy the constraint that the selection variable $S = 1$.

Note that after recovering the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$, one can generate data points with the novel specifications $(Z_1 = 1, Z_2 = 1)$, thus achieving extrapolative data generation.

#### 3.2.2 Non-Identifiable Case: Specifications Share Common Features

We now consider the scenario in Fig. 2(b), where the specifications $Z_1$ and $Z_2$ share a common parent $X_2$. This case is more complex than the previous scenario in Fig. 2(a)–here $X_2$ is a direct parent of both $Z_1$ and $Z_2$.

**Non-identifiability of novel data distribution.** We show in the following proposition that the distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is, in general, non-identifiable for the generating process in Fig. 2(b). Intuitively, this is because $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ involves distributions conditioning on $S = 0$, which are unknown. A full proof is given in Appendix C.2.

**Proposition 2.** *Suppose that Assumption 1 holds and the data generating process follows Fig. 2(b). Then, the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is not identifiable from the given data.*

**Existence of novel data sample.** Although the distribution corresponding to novel specifications, $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$, is not identifiable, one may still ask whether it is possible to generate samples that satisfy these specifications. Surprisingly, this is indeed possible under mild conditions. We begin by stating the following assumption.

**Assumption 2** (Common features in separate specifications)**.** *There exists value $\tilde{x}_2$ of $X_2$ such that*

$$p^{\mathcal{D}}(X_2 = \tilde{x}_2 \mid Z_i = 1) > 0, \quad i = 1, 2.$$

This assumption requires that there exists at least one value of the common feature $X_2$ that has a positive density under each specification individually in the given data. Intuitively, it ensures that the marginal distributions of the specifications overlap in the space of the shared feature, making it possible to generate samples satisfying novel combinations of $Z_1$ and $Z_2$. Under this assumption, one can naturally construct data points satisfying the novel specifications from the given data. A full proof deferred to Appendix C.3.

**Proposition 3.** *Suppose Assumptions 1 and 2 hold, and that the data generating process follows Fig. 2(b). Then, there exists a value $\tilde{\mathbf{x}}$ of $\mathbf{X}$ that can be constructed from the given data such that*

$$p(\mathbf{X} = \tilde{\mathbf{x}} \mid Z_1 = 1, Z_2 = 1) > 0.$$

Intuively, it is because it can be derived that

$$p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1) = \frac{p^{\mathcal{D}}(X_2 \mid Z_1 = 1) p^{\mathcal{D}}(X_2 \mid Z_2 = 1)}{p(X_2)} \cdot p^{\mathcal{D}}(X_1 \mid X_2, Z_1 = 1) p^{\mathcal{D}}(X_3 \mid X_2, Z_2 = 1),$$

and all the terms involved above are positive for $\mathbf{X} = \tilde{\mathbf{x}}$. Specifically, given estimates of $p^{\mathcal{D}}(X_3 \mid X_2, Z_2 = 1)$, $p^{\mathcal{D}}(X_1 \mid X_2, Z_1 = 1)$, and $p^{\mathcal{D}}(X_2 \mid Z_i = 1), i = 1, 2$, all of which can be learned from the given/training data, one can explicitly construct a data point $\tilde{\mathbf{x}}$ that satisfies the novel specifications. Moreover, if the relevant density functions are further assumed to be continuous, then one can construct a set of data points with nonzero Lebesgue measure satisfying the inequality in Proposition 3 (or Assumption 2), since it will hold throughout an open neighborhood of the point $\tilde{\mathbf{x}}$ (Royden & Fitzpatrick, 2018).

**Conservative solutions for non-identifiable distributions.** It is beneficial to provide a way to circumvent the inherent non-identifiability issue with $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ described in Proposition 2. Let us consider a "conservative" case in which $p(S = 0 \mid Z_1 = 1)$ and $p(S = 0)$ are very small, meaning that in the original data generating process, the chance of having novel scenarios is low:

**Assumption 3.** *We have $P(S = 0 \mid Z_i = 1) \approx 0$, $i = 1, 2$, and $P(S = 0) \approx 0$.*

Under this assumption, the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ becomes approximately identifiable, in the sense that it is approximately equal to a distribution constructed from certain distribution components implied by the training data, formally stated below.

**Proposition 4.** *Under Assumptions 1 and 3, the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is approximately identifiable from the given data.*

The proof, provided in Appendix C.4, is constructive and provides a way to find the distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ under Assumption 3. We call this solution a *conservative solution* since it does not enforce assumptions on the distribution of the (potential) data that were not in the given dataset, corresponding to $S = 0$.

### 3.3 SEDGE WITH MULTIPLE SPECIFICATIONS

We now extend the results to show how structural properties can be exploited for extrapolative data generation in the setting with more than two specifications. Due to page limits, we provide a summary of the theoretical extension here, and readers may refer to the details in Appendix C.5.

We first consider the multi-specification setting where specifications can be partitioned into disjoint subsets without sharing features. Under the extended basic coverage conditions (Assumption 4 in Appendix C.4) of Assumption 1 in the two-specification setting, we know that every specification subset appears with positive probability under selection. The novel distribution is therefore identifiable by recovering each component separately and recombining them; see Theorem 1 in Appendix C.5. When specifications share features, $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1})$ is generally not identifiable. Analogous to the two-specification setting, extrapolated data points can still be constructed. Leveraging conditional independence between specification subsets given the shared features, together with a positivity condition on the shared features, one can combine observed samples to form features satisfying the novel specification. The existence of valid novel data points is therefore guaranteed; see Theorem 2 in Appendix C.5. Finally, analogous to the conservative recovery result in the two-specification case, non-identifiability can be addressed, similarly to Proposition 4. Assuming that the probability of unobserved scenarios is low, the novel distribution $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1})$ becomes approximately identifiable and can be estimated from the given data, as seen from Theorem 3 in Appendix C.5.

**Summary of the theory.** We have demonstrated that extrapolative data generation for novel specification combinations is generally impossible without suitable assumptions. We then established conditions under which this task becomes feasible. In particular, the specification variables $Z_i$ are assumed to be conditionally independent given the features $\mathbf{X}$, allowing joint specification constraints to be decomposed into individual ones. This condition is also necessary in the sense that, without it, extrapolation cannot be guaranteed in general. For example, if there is an edge between $Z_1$ and $Z_2$ in Fig. 2(b), i.e., they are always conditionally dependent, then the factorization used to construct the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$, must involve at least one term involving both $Z_1$ and $Z_2$, which cannot be identified from training data. Furthermore, Assumption 3, surprisingly, renders the inherently non-identifiable novel data distribution approximately identifiable–it is needed only to construct this distribution, not to generate feasible extrapolated data points.

# 4 Practical Algorithms for Extrapolative Generation

The preceding section establishes conditions and a approximation of the feature distribution under novel combinations of specifications. We now develop two practical SEDGE approaches: likelihood-weighted resampling (LWR) and likelihood-guided diffusion (LGD). We first assume a known feature-specification structure, and then discuss structure estimation and representation learning. For clarity, we use the three-feature, two-specification setting in Fig. 1(b), which also forms the basis of our synthetic experiments.

## 4.1 Structure-Informed Sampling

Following Proposition 4 and Eq. 17 in Appendix C.4, we construct the approximation of the feature distribution under novel specification combination $z_1^\star, z_2^\star$:

$$p(x_1, x_2, x_3 \mid z_1^\star, z_2^\star) \propto a_\theta(x_1, x_2 \mid z_1^\star) q_\phi(z_2^\star \mid x_2) b_\eta(x_3 \mid x_2, z_2^\star),$$

where $a_\theta$, $q_\phi$, and $b_\eta$ approximate the anchor $p^{\mathcal{D}}(X_1, X_2 \mid Z_1)$, specification likelihood $p^{\mathcal{D}}(Z_2 \mid X_2)$, and remaining-feature conditional $p^{\mathcal{D}}(X_3 \mid Z_2, X_2)$, respectively. We parameterize $a_\theta$ and $b_\eta$ using conditional diffusion models and fit $q_\phi$ by conditional maximum likelihood. Both approaches share these components and differ only in how they incorporate $q_\phi$ into anchor sampling. After generating $(x_1, x_2)$, they both draw $x_3 \sim b_\eta(\cdot \mid x_2, z_2^\star)$ to obtain the complete sample $(x_1, x_2, x_3)$.

**Likelihood-weighted resampling (LWR).** We draw $K$ candidates $(x_1^{(k)}, x_2^{(k)}) \sim a_\theta(\cdot \mid z_1^\star)$ and assign normalized importance weights $w_k = \frac{q_\phi(z_2^\star \mid x_2^{(k)})}{\sum_{\ell=1}^K q_\phi(z_2^\star \mid x_2^{(\ell)})}$. We then resample candidate pairs according to these weights (Algorithm 1). Resampling preserves each pair, propagating the specification's influence on $X_2$ to the associated $X_1$. Finite-sample accuracy depends on whether the anchor candidates adequately cover regions favored by the likelihood.

**Likelihood-guided diffusion (LGD).** Alternatively, we propose to incorporate the likelihood during anchor denoising, avoiding reliance on a fixed pool of anchor candidates. Let $A_t$ denote the forward-diffused state of $A_0 := (X_1, X_2)$. We learn a noise-aware density $r_\psi(Z_2 \mid A_t, Z_1, t)$ by maximum likelihood estimation on synthetic examples: draw $A_0$ from the frozen anchor, draw $Z_2$ from $q_\phi(\cdot \mid X_{2,0})$, and forward-diffuse $A_0$. During sampling, we apply the score correction according to $s_{\text{guided}}(a_t, t) = \nabla_{a_t} \log a_{\theta,t}(a_t \mid z_1^\star) + \nabla_{a_t} \log r_\psi(z_2^\star \mid a_t, z_1^\star, t)$. This extends classifier guidance (Dhariwal & Nichol, 2021) to handle continuous specifications. The correction can move both $X_1$ and $X_2$ through their learned dependence, while keeping the anchor parameters fixed. Algorithm 2 describes the procedure, and Appendix E.2 contains the full training objectives.

## 4.2 Structure Estimation and Canonical Representation Learning

When $\mathbf{X}$ and $\mathbf{Z}$ are observed and their known structure satisfies the required conditional independences, no structure estimation is needed. For unknown structure, we fit conditional likelihoods $q_{i,M}(Z_i \mid \mathbf{X}_M)$ under candidate input masks and select each mask by validation negative log-likelihood plus a sparsity penalty. Alternatively, one can use conventional causal discovery methods to select the minimal parent set for each $Z_i$ to make it conditional independent from other specification variables. Comparesions are provided and discussed in Section.5.

When the original representation is incompatible with the required conditional independences, or when $\mathbf{X}$ and $\mathbf{Z}$ are not directly observed, we further learn a canonical representation (Appendix D.1), aiming

Table 1: Validity (%), joint MMD, and coverage (%) across five synthetic settings. † denotes adapted baselines.

| Method | Weighted Discrete | | | Linear | | | Cubic | | | Smoothstep | | | 2D Bridge | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ |
| *Conditional diffusion (CD, generation direction $Z \to X$)* | | | | | | | | | | | | | | | |
| CD | 13.55 | 0.1467 | 40.00 | 83.22 | 0.0025 | 78.52 | 78.12 | 0.0062 | 74.35 | 85.55 | 0.0034 | 70.49 | 80.65 | 0.0072 | 73.08 |
| Compositional CD | 75.50 | 0.1065 | 84.78 | 38.54 | 0.0139 | 31.25 | 31.74 | 0.0368 | 27.67 | 21.21 | 0.0497 | 16.06 | 20.93 | 0.1054 | 20.04 |
| *Classifier-guided diffusion (joint-X prior; likelihood $X_t \to Z$)* | | | | | | | | | | | | | | | |
| CG: sparse factorized | 76.13 | 0.0530 | 61.22 | 24.82 | 0.0161 | 7.42 | 35.74 | 0.0183 | 13.85 | 19.48 | 0.0383 | 4.87 | 36.04 | 0.0758 | 30.99 |
| CG: dense factorized | **82.85** | 0.0633 | 56.77 | 29.61 | 0.0116 | 9.83 | 26.16 | 0.0194 | 2.67 | 17.01 | 0.0423 | 1.14 | 5.70 | 0.0752 | 4.64 |
| CG: dense joint | 2.03 | 0.4807 | 5.93 | 18.20 | 0.0111 | 9.18 | 34.49 | 0.0143 | 13.77 | 25.72 | 0.0241 | 6.35 | 14.19 | 0.0394 | 9.52 |
| *Extrapolation-inspired baselines* | | | | | | | | | | | | | | | |
| Additive mean† | 0.00 | 0.1174 | 0.00 | **100.00** | 0.0088 | 0.55 | 0.00 | 0.1711 | 0.00 | 0.00 | 0.0716 | 0.00 | 0.00 | 0.3416 | 0.00 |
| Additive + residual† | 0.18 | 0.2322 | 0.65 | 7.81 | 0.1407 | 6.17 | 0.00 | 0.2037 | 0.00 | 1.35 | 0.1523 | 0.86 | 0.18 | 0.3266 | 0.10 |
| Fixed composition + residual† | 0.13 | 0.2353 | 0.50 | 0.03 | 0.2327 | 0.00 | 0.02 | 0.2014 | 0.00 | 0.07 | 0.2548 | 0.00 | 0.00 | 0.3798 | 0.00 |
| Rep4Ex-inspired + residual† | 0.85 | 0.2714 | 2.87 | 3.68 | 0.2490 | 3.06 | 0.23 | 0.2732 | 0.02 | 0.39 | 0.2621 | 0.02 | 0.03 | 0.3648 | 0.03 |
| *SEDGE (ours)* | | | | | | | | | | | | | | | |
| SEDGE: no weighting | 12.00 | 0.2198 | 29.95 | 8.98 | 0.4006 | 20.82 | 8.97 | 0.3650 | 17.61 | 8.92 | 0.4248 | 17.12 | 2.46 | 0.6294 | 4.10 |
| SEDGE-LGD | 56.80 | 0.0996 | 81.08 | 78.24 | 0.0104 | 79.17 | 71.89 | 0.0302 | 50.91 | 79.57 | 0.0147 | 70.05 | 76.25 | 0.0516 | 56.77 |
| SEDGE-LWR | 77.55 | **0.0225** | **89.97** | 91.98 | **0.0018** | **87.37** | **91.26** | **0.0024** | **84.29** | **91.55** | **0.0030** | **82.23** | **87.39** | **0.0016** | **78.86** |

to satisfy the conditional independence condition. A special case arises when $\mathbf{X}$ and $\mathbf{Z}$ are observed and an invertible transformation $\widehat{\mathbf{Z}} = T(\mathbf{Z})$ suffices to satisfy the conditions, leaving $\mathbf{X}$ unchanged. The general estimation objective and identifiability conditions are provided in Appendix D.

**Scope and practical extensions.** More complex structures require corresponding factorizations and sampling procedures, whose estimation errors may compound. These practical challenges are distinct from the theoretical conditions established earlier. For molecular design, we use a blockwise approximation suited to the limited set of target properties considered in our experiments (Algorithm 3). For text-to-image generation, we use DPS-style guidance that combines specification likelihoods in a single sampling objective, avoiding a separately trained noise-aware predictor (Algorithm 4). Blockwise grouping need not preserve all structural conditions, and DPS approximates posterior sampling under a learned joint prior. These practical experiments examine whether incorporating structural information can improve extrapolative generation, even with coarse blockwise structure or approximate DPS guidance. Developing scalable methods that more faithfully implement the general structural decomposition remains an important direction for future work.

## 5 Experiments

We evaluate the feasibility of the proposed SEDGE approach. Synthetic experiments compare our two sampling algorithms with baselines that drop certain properties. We further evaluate property-conditioned molecule design and text-to-image generation, examining the behavior of SEDGE compared with baselines that do not properly exploit out structural constraints. Regarding implementation, unless otherwise specified, we implement the anchor and remaining-feature conditional using conditional DDPMs (Ho et al., 2020), with conditioning variables supplied to the denoising networks. We model $q_\phi(Z_2 \mid X_2)$ using a conditional rational-quadratic spline flow with a Gaussian base (Durkan et al., 2019). Alternative likelihood models are compared in Appendix F.1.3.

### 5.1 Synthetic Experiments

**Data.** Five synthetic settings hold out joint $(Z_1, Z_2)$ regions while retaining marginal specification coverage. Each specification individually permits a broad range of the shared bridge $X_2$, but satisfying both restricts $X_2$ to a narrow overlap. Thus, models must generate unseen specification combinations with tightly constrained shared features. Detailed data-generating mechanisms, train–test splits, and design rationale are provided in Appendix F.1.1. **Baselinse.** Direct and compositional conditional diffusion (CD), classifier guidance (CG) with sparse/factorized/joint likelihoods, and additive (Lachapelle et al., 2023), fixed-composition (Wiedemer et al., 2023), and Rep4Ex (Saengkyongam et al., 2023) adaptations. We compare SEDGE-LWR, LGD, and an unweighted ablation (no likelihood is used for the anchor). **Metrics.** We jointly assess validity (↑), joint $(X, Z)$ MMD (↓), and reference-sample coverage (↑). Validity measures constraint satisfaction, MMD measures distributional fidelity, and coverage measures diversity. They should be interpreted together: valid samples may collapse to a few solutions, while high coverage alone does not ensure validity. Results are averaged on three random seeds. See the implementation and metrics details in Appendix F.1.2.

Table 2: Property-guided molecular extrapolation to a held-out joint-property region. Results are percentages averaged over 520 target conditions, with 30 generations per condition and method; invalid or missing outputs remain in the denominator. **Target** denotes generations satisfying both $|\Delta\text{LogP}| \leq 0.2$ and $|\Delta\text{TPSA}| \leq 5$[2]. **Region** denotes generations whose measured properties satisfy $\text{LogP} \geq 5$ and $\text{TPSA} \geq 100$[2]. **Valid** denotes molecular validity.

| Method | Target ↑ | Region ↑ | Valid ↑ |
|---|---|---|---|
| *Direct property-conditioned baselines* | | | |
| STGG+ | 25.20 | 77.37 | 99.96 |
| STGG+ (logit comp.) | 21.53 | 65.69 | 99.99 |
| *Pretrained text-to-molecule baselines* | | | |
| LDMol (fine-tuned) | 0.42 | 13.21 | 62.86 |
| BioT5+ (fine-tuned) | 1.13 | 17.94 | 99.96 |
| *Proposed method and ablations* | | | |
| SEDGE (blocks only) | 28.96 | 73.51 | 99.97 |
| SEDGE | **37.49** | **78.86** | **100.00** |

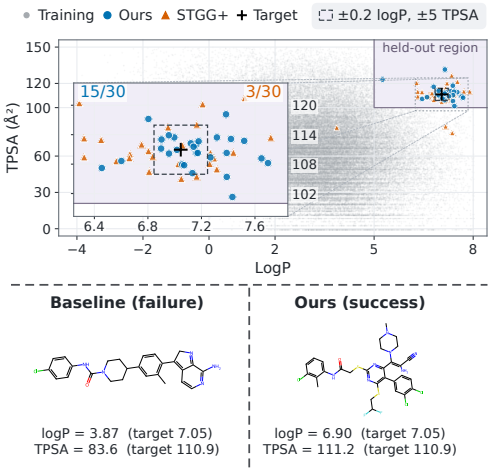



Figure 3: Target ($\text{logP} = 7.05$, $\text{TPSA} = 110.94$). **Top**: generated samples in the target space. The dashed box marks the target success tolerance. "15/30" and "3/30" are sucess ratios for our method and STGG+, respectively. **Bottom**: illustrative generated molecules for STGG+ and our method.

**Generation with observed variables and known structure.** We examine the benefit of exploiting structure under our assumptions. Table 1 evaluates held-out specification combinations requiring tightly constrained shared features. For each CG variant and SEDGE-LGD, we report the scale maximizing mean validity among $s \in \{0.5, 1, 2\}$; full scale comparisons and standard deviations appear in Table 5. CD and compositional CD use specification-conditioned models, as in Fig. 1(a), whose behavior on unseen combinations requires additional assumptions. SEDGE instead exploits the feature-to-specification structure to decompose the posterior into locally estimable factors. SEDGE instead exploits the feature-to-specification structure to decompose the posterior into locally estimable factors. SEDGE-LWR achieves the lowest joint MMD and highest coverage across the selected settings, though not always the highest validity. CG also uses feature-to-specification likelihoods, but guides a joint unconditional prior rather than composing conditional anchors and completions. Its generally lower coverage suggests that likelihood guidance alone does not provide the same benefit, although CG is not strictly confined to the prior's high-density regions. Table 7 in Appendix F.1.5 further shows that SEDGE more often locates shared features compatible with both specifications than CD—capturing the constrained "beak" in our motivating example. The extrapolation-inspired adaptations generally perform poorly, potentially reflecting restrictive functional forms and a mismatch with conditional sampling. Additive mean achieves perfect Linear validity but negligible coverage, illustrating why validity alone is insufficient. Finally, LWR outperforms LGD overall. One plausible explanation is that reweighting exploits compatible anchor proposals, whereas LGD additionally depends on noisy-state likelihood estimates and their gradients.

**Structure estimation.** We compare conditional independence (CI)-test-based PC-stable (Spirtes et al., 2001; Colombo & Maathuis, 2014) with conditional spline flows over all eight input masks, selected by validation negative log-likelihood plus a sparsity penalty. KCI and Fisher-Z recover all tested continuous graphs, while stronger sparsity penalties improve mask selection, reaching zero SHD at $\lambda = 1$. Full comparisons and sensitivity analyses appear in Appendix F.1.6. *Due to space limitations, we defer experiments on learned features and specifications, robustness to assumption violations, and parameter sensitivity to Appendix F.1.*

### 5.2 Extrapolative Property-Guided Molecule Design

We regard the desired LogP and TPSA as target specifications $Z$ and the learned molecular representation as features $X$. We construct a dataset that explicitly removes the joint region $\text{LogP} \geq 5$ and $\text{TPSA} \geq 100$[2] from training while retaining marginal coverage of each property. Leveraging the idea of the blockwise construction in Algorithm 3, we use STGG+ (Jolicoeur-Martineau et al., 2025) as the generative backbone and partition the representation into a shared block $C$ and two property-specific blocks $U_{\text{LogP}}$ and $U_{\text{TPSA}}$. Further details are provided in Appendix F.2.




Figure 4: Comparison of image generation results under different compositional prompts. w/o Spar and w/o CI denote SEDGE without sparsity loss and conditional independence structure, respectively.

We compare SEDGE against STGG+, LDMol (Chang & Ye, 2025), and BioT5+ (Pei et al., 2024). For the pretrained models, each numerical target pair is converted into a fixed natural-language template. We report target match, defined by $|\Delta\text{LogP}| \leq 0.2$ and $|\Delta\text{TPSA}| \leq 5$ [2], together with validity and the proportion of all generations entering the held-out region. SEDGE achieves higher target-match and held-out-region hit rates than the baselines. Moreover, SEDGE also outperforms STGG+ (logit comp.), which adds the LogP-only and TPSA-only logit shifts to an unconditional prediction at each decoding step, and SEDGE (blocks only), which retains the partition without either LWR and LGD, supporting the importance of our method for extrapolative generation.

### 5.3 Extrapolative Text-guided Image Generation

In our final experiment, we return to the running example: can we generate "a peacock eating ice cream"? To apply SEDGE to high-dimensional image–text data, we use the concept representations of Aligner (Xie et al., 2025b), taking image concepts as features and text concepts as specifications, and learn a sparse likelihood model of text concepts given image concepts. For a target prompt, we sample image concepts with Algorithm 4 and use SANA (Xie et al., 2024) as the image generator.

We compare our model against several state-of-the-art baselines, including the original SANA, Aligner, Stable Diffusion 3.5-Large (Esser et al., 2024), QwenImage (Wu et al., 2025), Z-Image (Cai et al., 2025), and GPT5.2 (Fig. 4). These baselines often suffer from concept leakage or the generation of spurious components, such as a peacock merely touching ice cream instead of eating it, or suitcases accompanied by redundant wings or birds. In contrast, SEDGE produces semantically coherent images that better preserve the required compositional attributes. We further conduct an ablation study by removing the sparsity loss and conditional independence structure, which results in more entangled compositions (e.g., generating a full bird instead of bird wings) and poorer adherence to the prompt (e.g., the ice cream is missing). Notably, our 1.6B-parameter model achieves more faithful generation than the 20B QwenImage model and performs comparably to GPT5.2, demonstrating its extrapolation ability and efficiency under limited training data. More details can be found Appendix F.3.

## 6 Conclusion and Discussions

For generative AI to reliably produce novel content, the ability to extrapolate beyond the training data is essential. Unlike traditional approaches that often rely on assumptions about the functional forms of the underlying relationships, we investigate this problem from a structural perspective, by employing a graphical model to represent how features and property specifications are related, as well as how the training data are limited. This framework enables us to characterize which components in the decomposition of the joint distribution of the training data can be leveraged to inform extrapolative data generation, and how this can be achieved in practice. We further provide conditions under which novel data can be generated reliably, as well as conditions under which the distribution of the novel data can be approximately constructed. As future work, we plan to explore how these ideas can address important problems such as drug discovery and novel protein structure prediction.

## AI USE STATEMENT

We used LLM-based tools to refine hypotheses and experimental designs, develop synthetic data-generation procedures, implement methods and evaluations, and interpret results. We also used these tools for literature searches, preparing figures and tables, and drafting and editing text. The authors take responsibility for all content, claims, code, and results in this submission.

## ETHICS STATEMENT

Property-guided molecule design has potential dual-use risks. Generated candidates require independent safety assessment before practical use.

## REPRODUCIBILITY STATEMENT

The appendix provides details on data generation, evaluation protocols, hyperparameters, and the anonymous code repository to support reproducibility.

## REFERENCES

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. *OpenAI Technical Report*, 2023. URL `https://cdn.openai.com/papers/dall-e-3.pdf`.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Jinho Chang and Jong Chul Ye. LDMol: A text-to-molecule diffusion model with structurally informative latent space surpasses AR models. In *Forty-second International Conference on Machine Learning*, 2025.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-$\alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-{\delta}: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. *arXiv preprint arXiv:2503.09641*, 2025a.

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, et al. Sana-video: Efficient video generation with block linear diffusion transformer. *arXiv preprint arXiv:2509.24695*, 2025b.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

Yusuf Dalva and Pinar Yanardag. Noiseclr: A contrastive learning approach for unsupervised discovery of interpretable directions in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24209–24218, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Kefan Dong and Tengyu Ma. First steps toward understanding the extrapolation of nonlinear models to unseen domains. *arXiv preprint arXiv:2211.11719*, 2022.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

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.

Rongyao Fang, Aldrich Yu, Chengqi Duan, Linjiang Huang, Shuai Bai, Yuxuan Cai, Kun Wang, Si Liu, Xihui Liu, and Hongsheng Li. Flux-reason-6m & prism-bench: A million-scale text-to-image reasoning dataset and comprehensive benchmark. *arXiv preprint arXiv:2509.09680*, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152. Curran Associates, Inc., 2023. doi: 10.52202/075280-2270. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Paper-Datasets_and_Benchmarks.pdf.

Tejas Gokhale, Hamid Palangi, Besmira Nushi, Vibhav Vineet, Eric Horvitz, Ece Kamar, Chitta Baral, and Yezhou Yang. Benchmarking spatial relationships in text-to-image generation. *arXiv preprint arXiv:2212.10015*, 2022.

James J. Heckman. Sample selection bias as a specification error. *Applied Econometrics*, 31:129–137, 1979. URL https://api.semanticscholar.org/CorpusID:30028243.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Imagen Team, Google. Imagen 3, 2024. URL https://arxiv.org/abs/2408.07009.

Alexia Jolicoeur-Martineau, Aristide Baratin, Kisoo Kwon, Boris Knyazev, and Yan Zhang. Any-property-conditional molecule generation with self-criticism using spanning trees. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=QGZd5Bfb1L.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

Lingjing Kong, Guangyi Chen, Petar Stojanov, Haoxuan Li, Eric Xing, and Kun Zhang. Towards understanding extrapolation: a causal lens. *Advances in Neural Information Processing Systems*, 37:123534–123562, 2024.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. *Advances in Neural Information Processing Systems*, 36:25112–25150, 2023.

Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph diffusion transformers for multi-conditional molecular generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 8065–8092. Curran Associates, Inc., 2024. doi: 10.52202/079017-0260. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0f6931a9e339a012a9909306d7c758b4-Paper-Conference.pdf.

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*, pp. 423–439. Springer, 2022.

Nan Liu, Yilun Du, Shuang Li, Joshua B Tenenbaum, and Antonio Torralba. Unsupervised compositional concepts discovery with text-to-image generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2095, 2023.

Yanchen Luo, Junfeng Fang, Sihang Li, Zhiyuan Liu, Jiancan Wu, An Zhang, Wenjie Du, and Xiang Wang. Text-guided small molecule generation via diffusion model. *iScience*, 27(11), 2024.

Judea Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of Cognitive Science Society (CSS-7)*, 1985.

Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. BioT5+: Towards generalized biological understanding with IUPAC integration and multi-task tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1216–1240, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.71. URL https://aclanthology.org/2024.findings-acl.71/.

Anjie Qiao, Zhen Wang, Chuan Chen, De-Fu Lian, and Enhong Chen. Composable score-based graph diffusion model for multi-conditional molecular generation. *ArXiv*, abs/2509.09451, 2025. URL https://api.semanticscholar.org/CorpusID:281252251.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021. URL https://api.semanticscholar.org/CorpusID:245335280.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

H. L. Royden and Patrick Fitzpatrick. *Real Analysis*. Pearson modern classic. Pearson, New York, NY, fourth edition edition, 2018. ISBN 9780134689494.

Sorawit Saengkyongam, Elan Rosenfeld, Pradeep Ravikumar, Niklas Pfister, and Jonas Peters. Identifying representations for intervention extrapolation. *arXiv preprint arXiv:2310.04295*, 2023.

Xinwei Shen and Nicolai Meinshausen. Engression: extrapolation through the lens of distributional regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(3): 653–677, 2025.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019. URL https://api.semanticscholar.org/CorpusID:195811894.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2001.

Mayank Vatsa, Aparna Bharati, and Richa Singh. Right looks, wrong reasons: Compositional fidelity in text-to-image generation. *arXiv preprint arXiv:2511.10136*, 2025.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. *Advances in Neural Information Processing Systems*, 36:6941–6960, 2023.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1900–1910, 2023.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Chengyue Wu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*, 2025a.

Shaoan Xie, Lingjing Kong, Yujia Zheng, Zeyu Tang, Eric P Xing, Guangyi Chen, and Kun Zhang. Learning vision and language concepts for controllable image generation. In *Forty-second International Conference on Machine Learning*, 2025b.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. *arXiv preprint arXiv:2310.08580*, 2023.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:11127–11150, 2023.

# APPENDICES FOR *"Structural Extrapolative Data Generation: Hardness, Feasibility, and Formal Conditions"*

## CONTENTS


**A Related Work** **15**

**B Limitations and Impact** **16**

**C Main Proofs for Extrapolative Generation** **16**

C.1 Proof of Proposition 1 . . . . . 16

C.2 Proof of Proposition 2 . . . . . 16

C.3 Proof of Proposition 3 . . . . . 17

C.4 Proof of Proposition 4 . . . . . 17

C.5 SEDGE with Multiple Specifications . . . . . 18

C.5.1 Identifiable Case: Specifications Do Not Share Common Features . . . . . 18

C.5.2 Non-Identifiable Case: Specifications Share Common Features . . . . . 18

C.6 Proof of Theorem 1 . . . . . 20

C.7 Proof of Theorem 2 . . . . . 20

C.8 Proof of Theorem 3 . . . . . 21

**D Identifiability of Features and Specifications** **22**

D.1 Canonical Representation of the Data Generating Process . . . . . 22

D.2 Component-wise Identifiability of Specifications $\mathbf{Z}$ via Conditional Independence . 23

D.3 Subspace Identifiability of Features $\mathbf{X}$ via Sparse Structure . . . . . 25

**E Appendix of Structured Extrapolative Data Generation Algorithms and Other Details** **29**

E.1 Likelihood-weighted Resampling and Likelihood-guided Diffusion Algorithms . . 29

E.2 Details on Learning Noise-aware Likelihood Guidance . . . . . 32

**F Appendix on Experiment Details and Additional Results** **32**

F.1 Synthetic Experiments . . . . . 32

F.1.1 Data: Synthetic Data Generation Processes . . . . . 32

F.1.2 Implementation & Metrics: Details . . . . . 35

F.1.3 Method: Likelihood Estimation Comparisons . . . . . 36

F.1.4 Results & Sensitivity Study: Full Table 1 with Standard Deviations and LGD Scale Sensitivity . . . . . 36

F.1.5 Estimation Reliability & Extrapolated Shared Feature Compatibility . . . . 36

F.1.6 Results & Sensitivity Study: Structure Estimation and Sparsity Penalties . . 38

F.1.7 Results: Feature or Specification not Given . . . . . 38

F.2 Extrapolative Property-to-Molecule Design . . . . . 39

F.2.1 Experiment setup . . . . . 39

F.2.2 More experiment results . . . . . 40
F.3 Extrapolative Text-to-Image Generation . . . . . 41
F.3.1 Experiment Setup . . . . . 41
F.3.2 Qualitative Evaluation Results . . . . . 42
F.3.3 Quantitative Evaluation Results . . . . . 42

# A Related Work

**Extrapolation.** Existing approaches for extrapolation largely fall into two categories, distinguished by where the assumption are imposed. The first category make assumptions on the data distribution, typically assuming the target distribution shares the same support as the source. It includes most work on the topic of out-of-distribution generalization, which focuses on invariance or reweighting rather than extrapolation on novel specifications (Yang et al., 2024). There is also work in regression and statistical inference studying prediction or uncertainty guarantees outside the observed support (Dong & Ma, 2022; Shen & Meinshausen, 2025) without addressing novel combinations of specifications.

The second category introduce functional assumptions on the generation process to regularize the behavior ouside training support. Special restrictions are introduced to data generating process including linear, additive, slot-wise, compositional generators, or other specific parametric relations (Liu et al., 2022; Lachapelle et al., 2023; Wiedemer et al., 2023; Saengkyongam et al., 2023). Other work derives extrapolation guarantees from regularities on distribution shifts, for instance, smoothness (Kong et al., 2024). These assumptions are hard to be verified from observed data and do not provide a general framework for extrapolative data generation on novel specifications. In contrast, we provide a general structural view to the problem that does not assume a particular functional form of the generating mechanism or smoothness of shifts.

**Text-to-image generation.** In recent years, diffusion-based approaches have become the dominant paradigm for text-to-image generation due to their strong performance (Rombach et al., 2022; Esser et al., 2024; Cai et al., 2025; Wu et al., 2025; Labs, 2024). One line of research focuses on improving the efficiency of training and sampling (Chen et al., 2023; 2024; Xie et al., 2024; 2025a; Chen et al., 2025a;b). Another line of work aims to enhance controllability by incorporating additional conditioning signals, such as structural or semantic guidance (Zhang et al., 2023; Ye et al., 2023; Zhao et al., 2023; Xie et al., 2023). A further line of research studies disentanglement in text-to-image generation, seeking to separate and manipulate semantic factors within the generative process (Wu et al., 2023; Liu et al., 2023; Xie et al., 2025b; Dalva & Yanardag, 2024). By synthesizing images for combinations of familiar concepts absent from training data, we illustrate how our method can address real-world requests with novel visual specifications.

**Property-guided molecule design.** Property-guided molecular design aims to produce molecules satisfying specified property constraints. Graph DiT incorporates numerical and categorical conditions into graph diffusion for multi-property generation (Liu et al., 2024). STGG+ combines autoregressive spanning-tree generation with flexible property conditioning and self-criticism, and evaluates generation at extreme property values (Jolicoeur-Martineau et al., 2025). Text-conditioned models, including BioT5+ and LDMol, provide an alternative interface for generating molecules from natural-language descriptions (Pei et al., 2024; Chang & Ye, 2025). More directly related to composition of different conditions, CSGD combines property-specific scores during sampling to enable flexible multi-property control (Qiao et al., 2025). However, supporting multiple conditions or exploring novel chemical regions does not by itself establish generalization to unseen combinations of property values. We use property-guided molecular design as a concrete, practically motivated example of extrapolative generation: the model must generate molecules satisfying joint property specifications absent from task-specific training, even when each property value has been observed separately. This application illustrates how our method can address real-world design problems involving novel specifications.

# B Limitations and Impact

**Limitations.** This paper relies on particular structural assumptions to make extrapolative data generation feasible and reliable. When these assumptions are violated, the performance of the proposed framework may degrade. In addition, the finite-sample properties of the method are not yet well understood, although this limitation is common in deep learning. Finally, further work is needed to develop more robust and efficient practical algorithms that can fully exploit the theoretical results under more complex graph structures.

**Impact.** This work aims to advance machine learning toward more reliable and trustworthy extrapolative data generation. We hope this line of research can improve the ability of generative AI to support reliable creativity and applications such as scientific discovery, drug discovery, and materials design. At the same time, improved extrapolative generation may also make it easier to produce novel but fake or misleading content, and such risks should be carefully considered.

# C Main Proofs for Extrapolative Generation

## C.1 Proof of Proposition 1

**Proposition 1.** *Suppose that Assumption 1 holds and the data generating process follows Fig. 2(a). Then, the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is identifiable from the given data.*

*Proof.* By Bayes Rule, we have

$$\begin{aligned}
&p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1) \\
&= p(X_1, X_2 \mid Z_1 = 1, Z_2 = 1) \\
&\quad \cdot p(X_3, X_4 \mid X_1, X_2, Z_1 = 1, Z_2 = 1) \\
&= p(X_1, X_2 \mid Z_1 = 1) p(X_3, X_4 \mid Z_2 = 1) \qquad (2) \\
&= p^{\mathcal{D}}(X_1, X_2 \mid Z_1 = 1) p^{\mathcal{D}}(X_3, X_4, \mid Z_2 = 1), \qquad (3)
\end{aligned}$$

where (2) follows from $\{X_1, X_2\} \perp\!\!\!\perp Z_2 \mid Z_1$ and $\{X_3, X_4\} \perp\!\!\!\perp \{X_1, X_2, Z_1\} \mid Z_2$. Eq. (3) follows from $\{X_1, X_2\} \perp\!\!\!\perp S \mid Z_1$, $\{X_3, X_4\} \perp\!\!\!\perp S \mid Z_2$, and Assumption 1. □

## C.2 Proof of Proposition 2

**Proposition 2.** *Suppose that Assumption 1 holds and the data generating process follows Fig. 2(b). Then, the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is not identifiable from the given data.*

*Proof.* We begin with deriving the data distribution following the novel specification, $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$.

$$\begin{aligned}
&p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1) && (4) \\
=&p(X_1, X_2 \mid Z_1 = 1, Z_2 = 1) p(X_3 \mid X_1, X_2, Z_1 = 1, Z_2 = 1) && (5) \\
=&p(X_2 \mid Z_1 = 1, Z_2 = 1) p(X_1 \mid X_2, Z_1 = 1, Z_2 = 1) p(X_3 \mid X_1, X_2, Z_1 = 1, Z_2 = 1) && (6) \\
=&p(X_2 \mid Z_1 = 1, Z_2 = 1) p(X_1 \mid X_2, Z_1 = 1, ) p(X_3 \mid X_2, Z_2 = 1) && (7) \\
=&p(X_2 \mid Z_1 = 1, Z_2 = 1) p(X_1 \mid X_2, Z_1 = 1, S = 1) p(X_3 \mid X_2, Z_2 = 1, S = 1) && (8) \\
=&p(X_2 \mid Z_1 = 1, Z_2 = 1) p^{\mathcal{D}}(X_1 \mid X_2, Z_1 = 1) p^{\mathcal{D}}(X_3 \mid X_2, Z_2 = 1), && (9)
\end{aligned}$$

Eq. (7) holds because $X_1 \perp\!\!\!\perp Z_2 \mid \{X_2, Z_1\}$ and $X_3 \perp\!\!\!\perp \{Z_1, X_1\} \mid \{X_2, Z_2\}$. Eq. (9) is the case because $X_1 \perp\!\!\!\perp S \mid \{X_2, Z_1\}$ and $X_3 \perp\!\!\!\perp S \mid \{X_2, Z_2\}$. Conditioning on $S = 1$ means that they are distributions implies by the given data. Note that although $X_2 \perp\!\!\!\perp S \mid \{Z_1, Z_2\}$ implies

$$p(X_2 \mid Z_1 = 1, Z_2 = 1) = p(X_2 \mid Z_1 = 1, Z_2 = 1, S = 1),$$

the quantity on the right-hand side is not defined since for the conditioning variables, $P(Z_1 = 1, Z_2 = 1, S = 1) = P(Z_1 = 1, Z_2 = 1 \mid S = 1) P(S = 1) = 0$, as the given data do not satisfy $Z_1 = 1, Z_2 = 1$.

Deriving $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ then reduces to the estimation of $p(X_2 \mid Z_1 = 1, Z_2 = 1)$.

$$p(X_2 \mid Z_1 = 1, Z_2 = 1) \tag{10}$$

$$= P(Z_1 = 1, Z_2 = 1 \mid X_2) \cdot \frac{p(X_2)}{P(Z_1 = 1, Z_2 = 1)} \tag{11}$$

$$\propto P(Z_1 = 1, Z_2 = 1 \mid X_2) p(X_2) \tag{12}$$

$$= P(Z_1 = 1 \mid X_2) p(Z_2 = 1 \mid X_2) p(X_2) \tag{13}$$

$$\propto p(X_2 \mid Z_1 = 1) p(X_2 \mid Z_2 = 1) \cdot \frac{1}{p(X_2)}. \tag{14}$$

The problem is that none of the three involved quantities, $p(X_2 \mid Z_1 = 1)$, $p(X_2 \mid Z_1 = 1)$, and $p(X_2)$ is known since we only have access to the distribution corresponding to the given data, for which $S = 1$. In fact,

$$p(X_2 \mid Z_1 = 1) = p(X_2 \mid Z_1 = 1, S = 1) P(S = 1 \mid Z_1 = 1) + p(X_2 \mid Z_1 = 1, S = 0) P(S = 0 \mid Z_1 = 1). \tag{15}$$

We do not have information of the data with $S = 0$, which are not in the given dataset. As a consequence, neither of $p(X_2 \mid Z_1 = 1, S = 0)$ and $P(S = 0 \mid Z_1 = 1)$ is known. Hence, $p(X_2 \mid Z_1 = 1)$ is not identifiable from the given data, and it also happens to $p(X_2 \mid Z_1 = 1)$ and $p(X_2)$. Therefore, $p(X_2 \mid Z_1 = 1, Z_2 = 1)$ is *not identifiable*, and hence according to (9), $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is not identifiable given the distribution of the observed data.

□

### C.3 Proof of Proposition 3

**Proposition 3.** *Suppose Assumptions 1 and 2 hold, and that the data generating process follows Fig. 2(b). Then, there exists a value $\tilde{\mathbf{x}}$ of $\mathbf{X}$ that can be constructed from the given data such that*

$$p(\mathbf{X} = \tilde{\mathbf{x}} \mid Z_1 = 1, Z_2 = 1) > 0.$$

*Proof.* First, note that $p(S = 1 \mid Z_1 = 1) > 0$ as for the given data, $S = 1$, and bear in mind that

$$p(X_2 = \tilde{x}_2 \mid Z_1 = 1, S = 1) > 0,$$

according to Assumption 2. Then according to (15), we have

$$p(X_2 = \tilde{x}_2 \mid Z_1 = 1) > 0.$$

Similarly, we have

$$p(X_2 = \tilde{x}_2 \mid Z_2 = 1) > 0.$$

Then according to (13), $p(X_2 = \tilde{x}_2 \mid Z_1 = 1, Z_2 = 1) > 0$. We further know that there exist $\tilde{x}_1$, as values of $X_1$, such that

$$p^{\mathcal{D}}(X_1 = \tilde{x}_1 \mid X_2 = \tilde{x}_2, Z_1 = 1) > 0;$$

this is because otherwise, $p^{\mathcal{D}}(X_1 = \tilde{x}_1 \mid X_2 = \tilde{x}_2, Z_1 = 1) = 0$ implies $p^{\mathcal{D}}(X_1 = \tilde{x}_1, X_2 = \tilde{x}_2 \mid Z_1 = 1) = 0$ and hence $p^{\mathcal{D}}(X_2 = \tilde{x}_2 \mid Z_1 = 1) = 0$, contradicting Assumption 2. Similarly, there exist values of $X_3$, denoted by $\tilde{x}_3$, such that

$$p^{\mathcal{D}}(X_3 = \tilde{x}_3 \mid X_2 = \tilde{x}_2, Z_2 = 1) > 0.$$

Finally, according to (9), we obtain

$$p(X_2 = \tilde{x}_2, X_1 = \tilde{x}_1, X_3 = \tilde{x}_3 \mid Z_1 = 1, Z_2 = 1) > 0.$$

□

### C.4 Proof of Proposition 4

**Proposition 4.** *Under Assumptions 1 and 3, the novel data distribution $p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1)$ is approximately identifiable from the given data.*

*Proof.* By Assumption 3, one can show

$$p(X_2 \mid Z_i = 1) \approx p^{\mathcal{D}}(X_2 \mid Z_i = 1), \quad i = 1, 2,$$
$$p(X_2) \approx p^{\mathcal{D}}(X_2).$$

Plugging these into (9) and (14) yields

$$\begin{aligned} & p(\mathbf{X} \mid Z_1 = 1, Z_2 = 1) \\ \approx & \frac{p^{\mathcal{D}}(X_2 \mid Z_1 = 1) p^{\mathcal{D}}(X_2 \mid Z_2 = 1)}{p^{\mathcal{D}}(X_2)} \frac{P(Z_1 = 1) P(Z_2 = 1)}{P(Z_1 = 1, Z_2 = 1)} \\ & \cdot p^{\mathcal{D}}(X_1 \mid X_2, Z_1 = 1) p^{\mathcal{D}}(X_3 \mid X_2, Z_2 = 1) \qquad (16) \\ \propto & p^{\mathcal{D}}(X_1, X_2 \mid Z_1 = 1) P^{\mathcal{D}}(Z_2 = 1 \mid X_2) \cdot p^{\mathcal{D}}(X_3 \mid X_2, Z_2 = 1). \qquad (17) \end{aligned}$$

All the three terms are implied by the given data. □

### C.5 SEDGE WITH MULTIPLE SPECIFICATIONS

We now extend the results from the previous subsection to show how structural properties can be exploited to enable extrapolative data generation in the general setting with multiple specifications. Analogous to the two-specification setting in Assumption 1, we aim to extrapolate to the novel combination of specifications $\mathbf{Z} = 1$ that is unseen in the given data due to selection, i.e., $p(\mathbf{Z} = 1 \mid S = 1) = 0$. We introduce the following basic conditions.

**Assumption 4** (Basic conditions for specifications and selection)**.** *We have*

$$P(S = 1) > 0, \qquad (18)$$
$$P(\mathbf{Z} = \mathbf{1}) > 0, \qquad (19)$$
$$P(Z_i = 1 \mid S = 1) > 0, \quad i \in [k]. \qquad (20)$$

Specifically, (18) ensures that the given dataset is nonempty. Eq. (19) guarantees that the target specification combination is feasible in the underlying population, while (20) ensures that each individual specification value appears in the given/training data in some context, providing the necessary marginal coverage to potentially extrapolate to the novel combinations.

#### C.5.1 IDENTIFIABLE CASE: SPECIFICATIONS DO NOT SHARE COMMON FEATURES

Recall that Fig. 2(a) and Proposition 4 consider the setting in which specifications do not share common features, by making use of the independence between $Z_1$ and $Z_2$, i.e., $Z_1 \perp\!\!\!\perp Z_2$. Building on this intuition, we now extend it to multiple specifications. We present the following result, which shows that when specifications do not share common features, the distribution corresponding to novel specifications, $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1})$, is identifiable from the observed data.

**Theorem 1.** *Suppose that there exists a partition of the specifications $\mathbf{Z}$ into disjoint specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ such that no feature is shared between any two different subsets and that features across different subsets are not adjacent. Suppose further that Assumption 4 holds and*

$$P(\mathbf{Z}_i = 1 \mid S = 1) > 0, \quad i \in [k].$$

*Then, the novel data distribution $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1})$ is identifiable from the given data.*

The proof is given in C.6. It is worth noting that, unlike the setting in Fig. 2(a) that requires all specifications not to share any feature, the result above further allows the specifications to be partitioned into subset of specifications. That is, while no features are shared across different subsets, arbitrary feature sharing and dependence relations are permitted within each subset.

#### C.5.2 NON-IDENTIFIABLE CASE: SPECIFICATIONS SHARE COMMON FEATURES

Previously, Fig. 2(b) and Fig. 3 study the setting in which two specifications share common features, by leveraging the conditional independence between $Z_1$ and $Z_2$ given their common child $X_2$, i.e., $Z_1 \perp\!\!\!\perp Z_2 \mid X_2$. Analogous to this case, we now consider a more general setting with multiple specifications that share common features. As shown in Proposition 2, the distribution satisfying

novel specifications is generally not identifiable. In the following, we extend Proposition 3 from the two-specification setting to show that generating novel data samples remains possible with multiple specifications, and generalize Proposition 4 to show how the non-identifiability issue with multiple specifications can be addressed by learning conservative solutions.

**Existence of novel data sample.** The following result shows that, under a generalization of Assumption 2 which requires that the common features have a positive density under each set of specifications individually in the given data, one can construct data points satisfying the novel specifications $\mathbf{Z} = \mathbf{1}$ from the given data.

**Theorem 2.** *Suppose that there exists a partition of the specifications $\mathbf{Z}$ into disjoint specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ such that they are conditionally independent given a subset of features $\mathbf{X}_c \subset \mathbf{X}$ (The set $\mathbf{X}_c$ is selected with respect to the chosen partition). Moreover, suppose that, given $\mathbf{X}_c$, each feature $X_j \notin \mathbf{X}_c$ is conditionally independent of all other features that are not associated with the same specification subset as $X_j$.*[3] *Suppose further that Assumption 4 holds,*

$$P(\mathbf{Z}_i = 1 \,|\, S = 1) > 0, \quad i \in [k],$$

*and there exists value $\tilde{\mathbf{x}}_c$ of $\mathbf{X}_c$ such that*

$$p^{\mathcal{D}}(\mathbf{X}_c = \tilde{\mathbf{x}}_c \,|\, \mathbf{Z}_i = 1) > 0, \quad i \in [k].$$

*Then, there exists a value $\mathbf{X} = \tilde{\mathbf{x}}$ of $\mathbf{X}$ that can be constructed from the given data such that*

$$p(\mathbf{X} = \tilde{\mathbf{x}} \,|\, \mathbf{Z} = \mathbf{1}) > 0.$$

The proof is provided in C.7. The theorem is rather general and accommodates complex structures, allowing one to exploit conditional independencies implied by the structural properties to generate data points satisfying novel specifications in a broad range of settings.

**Conservative solutions for non-identifiable distributions.** Similar to Proposition 4 with two specifications, the non-identifiability issue of $p(\mathbf{X} \,|\, \mathbf{Z} = \mathbf{1})$ with multiple specifications can be resolved by assuming that the chance of having novel scenarios in the data generating process, which are not contained in the given data, is low. This is formally stated in the following result, with a proof given in C.8.

**Theorem 3.** *Suppose that there exists a partition of the specifications $\mathbf{Z}$ into disjoint specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ such that they are conditionally independent given a subset of features $\mathbf{X}_c \subset \mathbf{X}$. Moreover, suppose that, given $\mathbf{X}_c$, each feature $X_j \notin \mathbf{X}_c$ is conditionally independent of all other features that are not associated with the same specification subset as $X_j$. Suppose further that Assumption 4 holds and*

$$\begin{aligned} P(\mathbf{Z}_i = 1 \,|\, S = 1) &> 0, \quad i \in [k], \\ P(S = 0 \,|\, \mathbf{Z}_i = 1) &\approx 0, \quad i \in [k], \\ P(S = 0) &\approx 0. \end{aligned}$$

*Then, the novel data distribution $p(\mathbf{X} \,|\, \mathbf{Z} = \mathbf{1})$ is approximately identifiable from the given data.*

**Specification grouping and the choice of $\mathbf{X}_c$.** For a fixed feature representation and dependency structure, an admissible specification partition uniquely determines its minimal conditioning set $\mathbf{X}_c$. Finer partitions relax the within-group coverage requirements, $P(\mathbf{Z}_i = \mathbf{1} \mid S = 1) > 0$, by requiring smaller target combinations to be observed in the training data. However, refinement can turn dependencies within a group into dependencies across groups, potentially requiring a different or larger conditioning set $\mathbf{X}_c$ to satisfy the separation assumptions. The resulting conditional distributions must also satisfy the shared-support condition stated in the theorems. Thus, weaker within-group coverage requirements alone do not ensure that a finer partition is admissible. Coarser partitions retain more dependencies within each group, but require coverage and estimation of larger joint factors. In the extreme case of placing all specifications in a single group, the coverage condition requires the entire target combination to occur in the training data, which conflicts with the unseen-combination setting considered here.

[3] Here, a feature is said to be *associated* with a specification set if it is a parent of at least one specification variable in that set.

Grouping existing variables does not itself discard information. In practical implementations, reconstruction fidelity and the accuracy of estimated factors also depend on the representation constraints, model capacity, and available training data. Enforcing a finer decomposition through learned representations or approximate factor models therefore need not improve generation performance.

### C.6 Proof of Theorem 1

**Theorem 1.** *Suppose that there exists a partition of the specifications $\mathbf{Z}$ into disjoint specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ such that no feature is shared between any two different subsets and that features across different subsets are not adjacent. Suppose further that Assumption 4 holds and*

$$P(\mathbf{Z}_i = 1 \,|\, S = 1) > 0, \quad i \in [k].$$

*Then, the novel data distribution $p(\mathbf{X} \,|\, \mathbf{Z} = \mathbf{1})$ is identifiable from the given data.*

*Proof.* With a slight abuse of notation, let $\mathbf{V}_{\mathbf{Z}_i} \subseteq \mathbf{X}$ be the parent set of specification subset $\mathbf{Z}_i$. By assumption, no feature is shared between the specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$, and features across different subsets are not adjacent. This implies that (1) $\mathbf{V}_{\mathbf{Z}_1}, \ldots, \mathbf{V}_{\mathbf{Z}_k}$ are conditionally independent of each other given $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ and (2) $\mathbf{V}_{\mathbf{Z}_i} \perp\!\!\!\perp \mathbf{Z}_j \,|\, \mathbf{Z}_i$ for $i \neq j$.

By Bayes Rule, we have

$$\begin{aligned}
&p(\mathbf{X} \,|\, \mathbf{Z} = \mathbf{1}) \\
&= \prod_{i=1}^{k} p(\mathbf{V}_{\mathbf{Z}_i} \,|\, \mathbf{Z} = \mathbf{1}) \\
&= \prod_{i=1}^{k} p(\mathbf{V}_{\mathbf{Z}_i} \,|\, \mathbf{Z}_i = \mathbf{1}) \\
&= \prod_{i=1}^{k} p(\mathbf{V}_{\mathbf{Z}_i} \,|\, \mathbf{Z}_i = \mathbf{1}, S = 1) \\
&= \prod_{i=1}^{k} p^{\mathcal{D}}(\mathbf{V}_{\mathbf{Z}_i} \,|\, \mathbf{Z}_i = \mathbf{1}).
\end{aligned}$$

Here, the second line follows from the fact that $\mathbf{V}_{\mathbf{Z}_1}, \ldots, \mathbf{V}_{\mathbf{Z}_k}$ are conditionally independent of each other given $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$. The third line follows from $\mathbf{V}_{\mathbf{Z}_i} \perp\!\!\!\perp \mathbf{Z}_j \,|\, \mathbf{Z}_i$ for $i \neq j$ and the assumption of $P(\mathbf{Z}_i = 1 \,|\, S = 1) > 0$. The fourth line follows from $\mathbf{V}_{\mathbf{Z}_i} \perp\!\!\!\perp S \,|\, \mathbf{Z}_i$. □

### C.7 Proof of Theorem 2

**Theorem 2.** *Suppose that there exists a partition of the specifications $\mathbf{Z}$ into disjoint specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ such that they are conditionally independent given a subset of features $\mathbf{X}_c \subset \mathbf{X}$ (The set $\mathbf{X}_c$ is selected with respect to the chosen partition). Moreover, suppose that, given $\mathbf{X}_c$, each feature $X_j \notin \mathbf{X}_c$ is conditionally independent of all other features that are not associated with the same specification subset as $X_j$.[4] Suppose further that Assumption 4 holds,*

$$P(\mathbf{Z}_i = 1 \,|\, S = 1) > 0, \quad i \in [k],$$

*and there exists value $\tilde{\mathbf{x}}_c$ of $\mathbf{X}_c$ such that*

$$p^{\mathcal{D}}(\mathbf{X}_c = \tilde{\mathbf{x}}_c \,|\, \mathbf{Z}_i = 1) > 0, \quad i \in [k].$$

*Then, there exists a value $\mathbf{X} = \tilde{\mathbf{x}}$ of $\mathbf{X}$ that can be constructed from the given data such that*

$$p(\mathbf{X} = \tilde{\mathbf{x}} \,|\, \mathbf{Z} = \mathbf{1}) > 0.$$

[4]Here, a feature is said to be *associated* with a specification set if it is a parent of at least one specification variable in that set.

*Proof.* With a slight abuse of notation, let $\mathbf{V}_{\mathbf{Z}_i} \subseteq \mathbf{X}$ be the parent set of specification subset $\mathbf{Z}_i$. By assumption, the disjoint specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ are conditionally independent given $\mathbf{X}_c$. Also, each feature $X_j \notin \mathbf{X}_c$ is conditionally independent of all other features that are not associated with the same specification subset as $X_j$.

By Bayes rule, We have

$$\begin{aligned}
&p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1}) \\
=&p(\mathbf{X}_c \mid \mathbf{Z} = \mathbf{1}) p(\mathbf{X} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z} = \mathbf{1}) \\
=&p(\mathbf{X}_c \mid \mathbf{Z} = \mathbf{1}) \prod_{i=1}^{k} p(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z} = \mathbf{1}) \\
=&p(\mathbf{X}_c \mid \mathbf{Z} = \mathbf{1}) \prod_{i=1}^{k} p(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z}_i = \mathbf{1}) \\
=&p(\mathbf{X}_c \mid \mathbf{Z} = \mathbf{1}) \prod_{i=1}^{k} p(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z}_i = \mathbf{1}, S = 1) \\
=&p(\mathbf{X}_c \mid \mathbf{Z} = \mathbf{1}) \prod_{i=1}^{k} p^{\mathcal{D}}(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z}_i = \mathbf{1})
\end{aligned} \tag{21}$$

Here, the third line follows from the fact that, given $\mathbf{X}_c$, each feature $X_j \notin \mathbf{X}_c$ is conditionally independent of all other features that are not associated with the same specification subset as $X_j$. The fourth line follows from the fact that each $X_j \in \mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c$ is conditionally independent with $\mathbf{Z}_l, l \neq i$ given $\mathbf{X}_c$. The fifth line follows from the assumption of $P(\mathbf{Z}_i = 1 \mid S = 1) > 0$ and the fact that each $X_j \in \mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c$ is conditionally independent with $S$ given $\mathbf{X}_c$.

Deriving $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1})$ then reduces to the estimation of $p(\mathbf{X}_c \mid \mathbf{Z} = \mathbf{1})$. We have

$$\begin{aligned}
&p(\mathbf{X}_c \mid \mathbf{Z} = \mathbf{1}) \\
=&P(\mathbf{Z} = \mathbf{1} \mid \mathbf{X}_c) \cdot \frac{p(\mathbf{X}_c)}{P(\mathbf{Z} = \mathbf{1})} \\
\propto&P(\mathbf{Z} = \mathbf{1} \mid \mathbf{X}_c) p(\mathbf{X}_c) \\
=&p(\mathbf{X}_c) \prod_{i=1}^{k} P(\mathbf{Z}_i = \mathbf{1} \mid \mathbf{X}_c) \qquad (22) \\
\propto&\frac{1}{p(\mathbf{X}_c)^{k-1}} \prod_{i=1}^{k} p(\mathbf{X}_c \mid \mathbf{Z}_i = \mathbf{1}). \qquad (23)
\end{aligned}$$

By assumption, there exists value $\tilde{\mathbf{x}}_c$ of $\mathbf{X}_c$ such that

$$p^{\mathcal{D}}(\mathbf{X}_c = \tilde{\mathbf{x}}_c \mid \mathbf{Z}_i = 1) > 0, \quad i \in [k],$$

This implies

$$p(\mathbf{X}_c = \tilde{\mathbf{x}}_c \mid \mathbf{Z}_i = 1) > 0.$$

Then according to (23), $p(\mathbf{X}_c = \tilde{\mathbf{x}}_c \mid \mathbf{Z} = \mathbf{1}) > 0$. We further know that there exist $\tilde{\mathbf{v}}_{\mathbf{Z}_i} \setminus \tilde{\mathbf{x}}_c$, as values of $\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c$, such that

$$p^{\mathcal{D}}(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c = \tilde{\mathbf{v}}_{\mathbf{Z}_i} \setminus \tilde{\mathbf{x}}_c \mid \mathbf{X}_c = \mathbf{x}_c, \mathbf{Z}_i = \mathbf{1}) > 0;$$

Finally, according to (21), we can construct the point such that

$$p(\mathbf{X} = \tilde{x} \mid \mathbf{Z} = \mathbf{1}) > 0.$$

□

### C.8 Proof of Theorem 3

**Theorem 3.** *Suppose that there exists a partition of the specifications $\mathbf{Z}$ into disjoint specification subsets $\mathbf{Z}_1, \ldots, \mathbf{Z}_k$ such that they are conditionally independent given a subset of features $\mathbf{X}_c \subset \mathbf{X}$.*

*Moreover, suppose that, given $\mathbf{X}_c$, each feature $X_j \notin \mathbf{X}_c$ is conditionally independent of all other features that are not associated with the same specification subset as $X_j$. Suppose further that Assumption 4 holds and*

$$P(\mathbf{Z}_i = 1 \mid S = 1) > 0, \quad i \in [k],$$
$$P(S = 0 \mid \mathbf{Z}_i = 1) \approx 0, \quad i \in [k],$$
$$P(S = 0) \approx 0.$$

*Then, the novel data distribution $p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1})$ is approximately identifiable from the given data.*

*Proof.* By assumption, one can show

$$p(\mathbf{X}_c \mid \mathbf{Z}_i = \mathbf{1}) \approx p^{\mathcal{D}}(\mathbf{X}_c \mid \mathbf{Z}_i = \mathbf{1}), \quad i \in [k],$$
$$p(\mathbf{X}_c) \approx p^{\mathcal{D}}(\mathbf{X}_c).$$

Plugging these into (23) and (21) yields

$$\begin{aligned} p(\mathbf{X} \mid \mathbf{Z} = \mathbf{1}) &\propto \frac{1}{p^{\mathcal{D}}(\mathbf{X}_c)^{k-1}} \prod_{i=1}^{k} p^{\mathcal{D}}(\mathbf{X}_c \mid \mathbf{Z}_i = 1) \prod_{i=1}^{k} p^{\mathcal{D}}(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z}_i = \mathbf{1}) \\ &\propto p^{\mathcal{D}}(\mathbf{X}_c \mid \mathbf{Z}_r = \mathbf{1}) \prod_{i \neq r} P^{\mathcal{D}}(\mathbf{Z}_i = \mathbf{1} \mid \mathbf{X}_c) \prod_{i=1}^{k} p^{\mathcal{D}}(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z}_i = \mathbf{1}) \\ &= p^{\mathcal{D}}(\mathbf{V}_{\mathbf{Z}_r} \cup \mathbf{X}_c \mid \mathbf{Z}_r = \mathbf{1}) \prod_{i \neq r} \left[ P^{\mathcal{D}}(\mathbf{Z}_i = \mathbf{1} \mid \mathbf{X}_c) p^{\mathcal{D}}(\mathbf{V}_{\mathbf{Z}_i} \setminus \mathbf{X}_c \mid \mathbf{X}_c, \mathbf{Z}_i = \mathbf{1}) \right], \end{aligned}$$

where $r \in \{1, \ldots, k\}$. □

# D Identifiability of Features and Specifications

We focus on the case where the features and the specifications are neither observed nor given. Recall that in Section 3.1 we introduced the structural assumptions, including the conditional independence of the specifications given the features. Here, we want to clarify that the violation of this assumption does not forbid the application of our proposal, while it potentially hurts how precise the results would be, as the identifiability of the latent variables would be weaker.

## D.1 Canonical Representation of the Data Generating Process

**Canonical representation of the data generation process.** We first introduce a canonical representation of the data generation process, and then demonstrate its generality.

**Definition 1.** *(Canonical Representation of the Data Generation Process) The features $\mathbf{X} = \{X_1, \cdots, X_n\}$ are unconditionally independent, and $\mathbf{X} \sim p_{\mathbf{X}}$. Each specification $Z_i \in \mathbf{Z}$, for $i = 1, \cdots, d$, follows the structural equation $Z_i = h_i(\mathbf{V}_i, e_i)$, where $e_i$ is the noise term, $\mathbf{V}i := \mathrm{PA}_{Z_i}$ denotes the parent set of $Z_i$, and $\mathbf{V}_i \subseteq \mathbf{X}$. The observed modality data corresponding to $\mathbf{X}$ and $\mathbf{Z}$ are denoted by $\mathbf{Y}_{\mathbf{X}}$ and $\mathbf{Y}_{\mathbf{Z}}$, respectively. They are generated by two mixing procedures: $\mathbf{Y}_{\mathbf{X}} = g_X(\mathbf{X})$, $\mathbf{Y}_{\mathbf{Z}} = g_Z(\mathbf{Z})$, where $g_X : \mathcal{X} \mapsto \mathcal{Y}_{\mathcal{X}}$ and $g_Z : \mathcal{Z} \mapsto \mathcal{Y}_{\mathcal{Z}}$ are both invertible. The overall observed data domain is defined as $\mathcal{D} := \mathcal{Y}_{\mathcal{X}}, \mathcal{Y}_{\mathcal{Z}}$.*

This canonical representation is general, in the sense that the following cases can be reduced to it. First, consider cases where there are dependencies among $\mathbf{X}$. These are not excluded by the structural assumptions in Sec. 3.1, as they do not affect the identifiability of the posterior distribution (since component-wise identifiability of $\mathbf{X}$ is not required). Suppose there is an edge $X_1 \to X_3$ in the structure of Fig. 2(b). In this case, $X_3$ is a function of $X_1$ and $\epsilon_3$, assuming that $\epsilon_3$ is the noise term for $X_3$. Then, we can replace $X_3$ with $\epsilon_3$ to render the features unconditionally independent, while introducing an additional edge $X_1 \to Z_2$ as a consequence of the original edge $X_1 \to X_3$. More generally, if there are additional edges within $\mathbf{X}$, the same transformation can be applied, converting them into (possibly additional) edges from $\mathbf{X}$ to $\mathbf{Z}$, thereby reducing the system to the canonical representation. Second, consider cases where there are edges among $\mathbf{Z}$, which are explicitly excluded

in Sec. 3.1. This is because dependencies among $\mathbf{Z}$ would make the identifiability of the posterior distribution of features under new specification combinations significantly more difficult to analyze. Nevertheless, we illustrate how such cases can also be reduced to the canonical representation. Suppose that there is an determinitic edge $Z_1 \to Z_2$, then $Z_2$ can be seen as a function of $Z_1$ and $e_2$, where $e_2$ is the noise term for $Z_2$. If we substitute $Z_2$ with $e_2$ to satisfy the conditional independence constraint, then there will be an additional link $X_3 \to Z_1$. Similarly, in more general settings, the same transformation can be applied and transform the edges between $\mathbf{Z}$ to possibly extra edges from $\mathbf{X}$ to $\mathbf{Z}$, to reduce to the canonical representation as defined in Definition 1. The canonical representation enables a clearer analysis of identifiability for both the latent variables and the posterior distributions.

It is worth noting that, if the within-specification relations are not deterministic, deriving the conditional distribution of the features given novel specification combinations always involves a term containing both $Z_1$ and $Z_2$. This term is undefined for unseen combinations, preventing extrapolative data generation. It makes the conditional independence of the specifications given the features the only necessary condition for extrapolative generation.

**Existing related identifiability results.** Xie et al. (2025b) gives conditions for identifiability for both the features $\mathbf{X}$ and the specifications $\mathbf{Z}$. In our work, we share the part where the specifications are required to be identifiable in order to decompose the constraint of a novel combination of specifications into individual ones, and the individual constraints are already defined as seen in the data (recall that there is no novel value for any specification $Z_i$). The component-wise identifiability of $\mathbf{Z}$ is mostly built on the conditional independence of each $Z_i$ given $\mathbf{X}$ (see Condition 4.2 and Theorem 4.4 in their work). However, Condition 4.3 for deriving component-wise identifiability of $\mathbf{X}$ is rather strong in our setting, and our extrapolative data generation does not require each $X_i$ to be identifiable. Instead, to guarantee a sufficient and clean estimate of the conditional distribution $p(Z_i \mid \mathbf{V}_i)$, we need to locate the subspace of $\mathbf{V}_i$. Such a property should be facilitated by the sparsity constraint of the functions from $\mathbf{X}$ to $Z_i$. At the same time, features that connect to multiple specifications should be easier to identify.

Formally, we derive the identifiability of both specifications $\mathbf{Z}$ and subspace identifiability of the features $\mathbf{X}$.

**Logic of proving identifiability in this paper.** The high-level idea for establishing identifiability of the latent variables is as follows. Assume the observed joint distribution $p(\mathbf{Y_X}, \mathbf{Y_Z})$ is generated following the canonical representation of the data generation process in Definition 1, and we learn a model $(\hat{g}_X, \hat{g}_Z, p_{\hat{\mathbf{X}}}, p_{\hat{\mathbf{Z}}|\hat{\mathbf{X}}})$ that assumes the same process as in Definition 1 and matches the true joint distribution, i.e.,

$$p_{\mathbf{Y_X},\mathbf{Y_Z}}(\mathbf{y_x}, \mathbf{y_z}) = p_{\hat{\mathbf{Y}}_X,\hat{\mathbf{Y}}_Z}(\mathbf{y_x}, \mathbf{y_z}), \forall y_x, y_z \in \mathcal{D},$$

where the random variables $\mathbf{Y_X}, \mathbf{Y_Z}$ are generated from the true process $(g_X, g_Z, p_{\mathbf{X}}, p_{\mathbf{Z}|\mathbf{X}})$, and $\hat{\mathbf{Y}}_X, \hat{\mathbf{Y}}_Z$ are generated from the estimated process $(\hat{g}_X, \hat{g}_Z, p_{\hat{\mathbf{X}}}, p_{\hat{\mathbf{Z}}|\hat{\mathbf{X}}})$. In the following, leveraging this distributional equivalence along with additional assumptions, we establish relationships between $\hat{\mathbf{Z}}$ and $\mathbf{Z}$, and between $\hat{\mathbf{X}}$ and $\mathbf{X}$.

### D.2 Component-wise Identifiability of Specifications $\mathbf{Z}$ via Conditional Independence

The following theorem is adapted from ConceptAligner Xie et al. (2025b) (Step 1 of Theorem 4.4), where the key idea is to exploit conditional independence to identify the latent variables.

**Theorem 4.** *(Component-wise Identifiability of Specifications) If the following conditions C1-3 hold for the canonical data generation process, then the specification variables $\mathbf{Z}$ are component-wise identifiable.*

- *C1 (Third-order differentiable and positive). For every $\mathbf{x} \in \mathbb{R}^n$, the conditional density $p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z} \mid \mathbf{x})$ is strictly positive and third-order continuously differentiable with respect to $\mathbf{z} \in \mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^n$.*

- *C2 (Conditional independence). The specification variables $\mathbf{Z} = Z_1, \cdots, Z_d$ are conditionally independent given the features $\mathbf{X}$.*
- *C3 (Sufficient variability on $\mathbf{Z}$). For each fixed value of $\mathbf{Z}$, there exist $2d+1$ distinct values of $\mathbf{X}$, denoted by $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(2d)}$, such that the $2d$ vectors $l(\mathbf{z}, \mathbf{x}^{(m)}) = w(\mathbf{z}, \mathbf{x}^{(m)}) - w(\mathbf{z}, \mathbf{x}^{(0)}) \in \mathbb{R}^{2d}, m = 1, \cdots, 2d$, where*

$$w(\mathbf{z}, \mathbf{x}^{(m)}) = \left(\frac{\partial \log p(\mathbf{Z}_i \mid \mathbf{X})}{\partial Z_i}, \frac{\partial^2 \log p(\mathbf{Z}_i \mid \mathbf{X})}{\partial Z_i^2}\right)_{i \in [d]} \mid_{\mathbf{X}=\mathbf{x}^{(m)}} .$$

*Proof.* Consider an observationally equivalent estimated model, and define the separate latent coordinate maps

$$h = g_X^{-1} \circ \hat{g}_X, \qquad r = g_Z^{-1} \circ \hat{g}_Z, \qquad \mathbf{X} = h(\hat{\mathbf{X}}), \quad \mathbf{Z} = r(\hat{\mathbf{Z}}).$$

The joint map $(\hat{\mathbf{X}}, \hat{\mathbf{Z}}) \mapsto (h(\hat{\mathbf{X}}), r(\hat{\mathbf{Z}}))$ has a block-diagonal Jacobian. Observational equivalence and change of variables therefore give

$$\hat{p}(\hat{\mathbf{X}}, \hat{\mathbf{Z}}) = p(\mathbf{X}, \mathbf{Z}) |\det J_h(\hat{\mathbf{X}})| \, |\det J_r(\hat{\mathbf{Z}})|,$$
$$\hat{p}(\hat{\mathbf{x}}) = p(\mathbf{X}) |\det J_h(\hat{\mathbf{X}})|.$$

Dividing the joint density by the feature marginal cancels the feature-Jacobian factor and yields

$$\hat{p}(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}}) = \frac{\hat{p}(\hat{\mathbf{X}}, \hat{\mathbf{Z}})}{\hat{p}(\hat{\mathbf{X}})} = p(\mathbf{Z} \mid \mathbf{X}) \, |\det J_r(\hat{\mathbf{Z}})|. \tag{24}$$

Below, densities are evaluated at these matched coordinates, with $p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}})$ denoting the estimated conditional density.

Since, in the data-generating process, $\mathbf{Z} = Z_1, \cdots, Z_d$ are assumed to be conditionally independent given $\mathbf{X}$, and the estimated model is assumed the same way, we have, $p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}}) = \prod_{i \in [d]} p(\hat{Z}_i \mid \hat{\mathbf{X}})$, and the second cross derivative of the log conditional density is zero, i.e,

$$\frac{\partial^2 \log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}})}{\partial \hat{Z}_k \partial \hat{Z}_l} = \frac{\partial^2 \sum_{i \in [d]} \log p(\hat{\mathbf{Z}}_i \mid \hat{\mathbf{X}})}{\partial \hat{Z}_k \partial \hat{Z}_l} \equiv 0, \forall k \neq l, k, l \in [d]. \tag{25}$$

Substituting (24) into the second-order cross derivative of any pair $k \neq l, k, l \in [d]$, we have,

$$0 = \frac{\partial^2 \log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}})}{\partial \hat{Z}_k \partial \hat{Z}_l} \tag{26}$$

$$= \sum_{i \in [d]} \frac{\partial \log p(Z_i \mid \mathbf{X})}{\partial Z_i} \frac{\partial^2 Z_i}{\partial \hat{Z}_k \partial \hat{Z}_l} + \sum_{i \in [d]} \frac{\partial^2 \log p(Z_i \mid \mathbf{X})}{\partial Z_i^2} \frac{\partial Z_i}{\partial \hat{Z}_l} \frac{\partial Z_i}{\partial \hat{Z}_k} + \tag{27}$$

$$\sum_{i,j \in [d], i \neq j} \frac{\partial^2 \log p(Z_i \mid \mathbf{X})}{\partial Z_i Z_j} \left(\frac{\partial Z_i}{\partial \hat{Z}_l} \frac{\partial Z_i}{\partial \hat{Z}_k} + \frac{\partial Z_j}{\partial \hat{Z}_l} \frac{\partial Z_i}{\partial \hat{Z}_k}\right) + \frac{\partial^2 |\det J_{\hat{g}_Z^{-1} \circ g_Z}|}{\partial \hat{Z}_k \partial \hat{Z}_l}, \tag{28}$$

where $\frac{\partial^2 \log p(\mathbf{Z}_i | \mathbf{X})}{\partial Z_i Z_j} = 0$ due to conditional independence. Now, for any fixed values of $Z_i, i \in [d]$, suppose we have $2d+1$ different values of $\mathbf{X}$, i.e., $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(m)}, \cdots, \mathbf{x}^{(2d)}$, so that we can construct $2d$ linearly independent vectors:

$$l(\mathbf{z}, \mathbf{x}^{(m)}) = w(\mathbf{z}, \mathbf{x}^{(m)}) - w(\mathbf{z}, \mathbf{x}^{(0)}) \in \mathbb{R}^{2d},$$

where

$$w(\mathbf{z}, \mathbf{x}^{(m)}) = \left(\frac{\partial \log p(\mathbf{Z}_i \mid \mathbf{X})}{\partial Z_i}, \frac{\partial^2 \log p(\mathbf{Z}_i \mid \mathbf{X})}{\partial Z_i^2}\right)_{i \in [d]} \mid_{\mathbf{X}=\mathbf{x}^{(m)}} .$$

By condition C3, we obtain $2d$ linearly independent vectors $l(\mathbf{z}, \mathbf{x}^{(m)})$. This allows us to solve the linear system in (28), by subtracting $w(\mathbf{z}, \mathbf{x}^{(0)})$ from each of the $w(\mathbf{z}, \mathbf{x}^{(m)}); m = 1, \cdots, 2d$ rows. We conclude that, for all $Z_i$, $\frac{\partial^2 Z_i}{\partial \hat{Z}_k \partial \hat{Z}_l} = 0$ and $\frac{\partial Z_i}{\partial \hat{Z}_k} \frac{\partial Z_i}{\partial \hat{Z}_l} = 0$ for any pair $k \neq l$. Therefore, each $Z_i$ can depend on at most one component of $\hat{\mathbf{Z}}$. Therefore, there exists a permutation of $[d]$, i.e., $\pi$, so that for each $Z_i, i = 1, \cdots, d$, it is a function solely of $\hat{Z}_{\pi(i)}$. This establishes that $\mathbf{Z}$ is component-wise identifiable. □

### D.3 Subspace Identifiability of Features $\mathbf{X}$ via Sparse Structure

**Theorem 5.** *(Subspace Identifiability of Features) If conditions C1–C2 in Theorem 4 hold, together with the following conditions C4–C5, then the feature variables $\mathbf{X}$ are identifiable in the following sense:*

- *C4 (Sufficient variability). For each $Z_k \in \mathbf{Z}$ and each $X_j \in \mathbf{V}_{\pi(k)}$, and for every fixed value of the parent variables $\{X_j : X_j \in \mathbf{V}_{\pi(k)}\}$, there exist $|\mathbf{V}_{\pi(k)}|$ distinct values of $Z_{\pi(k)}$, denoted by $z^{(1)}, \ldots, z^{(|\mathbf{V}_{\pi(k)}|)}$, such that the vectors*

$$u(\mathbf{x}, z_{\pi(k)}) = \left( \frac{\partial \eta'_{\pi(k)}}{\partial X_j} \right)_{X_j \in \mathbf{V}_{\pi(k)}} \Big|_{Z_{\pi(k)} = z^{(m)}}$$

  *are linearly independent.*

- *C5 (Sparsity). The model $p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}})$ is estimated under a sparsity constraint on the dependency structure from $\hat{\mathbf{X}}$ to each component of $\hat{\mathbf{Z}}$. Specifically, among models matching the observed joint distribution, the estimated total number of feature–specification edges satisfies*

$$\sum_{k=1}^{d} |\hat{\mathbf{V}}_k| \leq \sum_{k=1}^{d} |\mathbf{V}_{\pi(k)}|.$$

  *This holds, in particular, for a model with minimum edge count among observationally equivalent models when the true model is feasible.*

*Under these conditions, the feature variables $\mathbf{X}$ satisfy the following identifiability properties:*

- *Case 1. If $X_j$ is not a parent of any specification (i.e., $X_j$ is an isolated feature variable), then $X_j$ can be an arbitrary function of $\hat{\mathbf{X}}$, and is therefore not identifiable.*
- *Case 2. If $X_j$ is a parent of exactly one specification $Z_{\pi(k)}$, then $X_j$ can depend only on the estimated parent set $\hat{\mathbf{V}}_k$ of $\hat{Z}_k$. Under C5, the size of this admissible subspace matches that of the true parent set, i.e., $|\hat{\mathbf{V}}_k| = |\mathbf{V}_{\pi(k)}|$.*
- *Case 3. If $X_j$ is a parent of multiple specifications, then $X_j$ can depend only on the intersection of the estimated parent sets corresponding to those specifications. This intersection is nonempty, since the estimated counterpart of $X_j$ must belong to each such parent set. In particular, if the intersection contains only one element, then $X_j$ is component-wise identifiable.*

The following lemma is purely algebraic and uses only a fixed matrix $A(\hat{\mathbf{x}}_0)$. In the proof below, this matrix is taken to be $J_h(\hat{\mathbf{x}}_0)$, after nonsingularity and the required support restriction have been established for the feature-coordinate Jacobian.

**Lemma 1** (Permuted supergraph)**.** *Let $\mathbf{V}_{\pi(k)} \subseteq \{X_1, \ldots, X_n\}$ and $\hat{\mathbf{V}}_k \subseteq \{\hat{X}_1, \ldots, \hat{X}_n\}$, for $k \in [d]$, be the true and estimated parent sets after aligning the specifications. Fix a feature value $\hat{\mathbf{x}}_0$, and let $A(\hat{\mathbf{x}}_0) \in \mathbb{R}^{n \times n}$ be a fixed nonsingular matrix. Suppose that, for every $k \in [d]$,*

$$A_{jl}(\hat{\mathbf{x}}_0) = 0 \quad \text{whenever } X_j \in \mathbf{V}_{\pi(k)} \text{ and } \hat{X}_l \notin \hat{\mathbf{V}}_k. \tag{29}$$

*Then there exists a single permutation $\sigma$ of $[n]$ such that*

$$\sigma(\mathbf{V}_{\pi(k)}) \subseteq \hat{\mathbf{V}}_k, \qquad k \in [d], \tag{30}$$

*where $\sigma(\mathbf{V}) := \{\hat{X}_{\sigma(j)} : X_j \in \mathbf{V}\}$. Equivalently, the estimated feature–specification graph contains a copy of the true graph under one permutation of the feature indices. If, in addition,*

$$\sum_{k=1}^{d} |\hat{\mathbf{V}}_k| \leq \sum_{k=1}^{d} |\mathbf{V}_{\pi(k)}|, \tag{31}$$

*then $\hat{\mathbf{V}}_k = \sigma(\mathbf{V}_{\pi(k)})$ for every $k$.*

*Proof.* Since $A(\hat{\mathbf{x}}_0)$ is nonsingular, the Leibniz formula gives

$$0 \neq \det A(\hat{\mathbf{x}}_0) = \sum_{\sigma \in \mathfrak{S}_n} \operatorname{sgn}(\sigma) \prod_{j=1}^{n} A_{j,\sigma(j)}(\hat{\mathbf{x}}_0).$$

Here $\mathfrak{S}_n$ is the set of permutations of $[n]$, and $\operatorname{sgn}(\sigma) \in \{-1, 1\}$ is the sign of the permutation. Thus at least one permutation $\sigma$ satisfies $A_{j,\sigma(j)}(\hat{\mathbf{x}}_0) \neq 0$ for every $j$. If $X_j \in \mathbf{V}_{\pi(k)}$, equation equation 29 then implies $\hat{X}_{\sigma(j)} \in \hat{\mathbf{V}}_k$. The same permutation applies to all $k$, proving equation 30. Consequently, $|\hat{\mathbf{V}}_k| \geq |\mathbf{V}_{\pi(k)}|$ for every $k$. Under equation 31, none of these inclusions can be strict, so $\hat{\mathbf{V}}_k = \sigma(\mathbf{V}_{\pi(k)})$ for every $k$. □

*Proof of Theorem 5.* We aim to characterize, for each true feature $X_j$, which estimated features in $\hat{\mathbf{X}}$ it may depend on. Using the component-wise identifiability of $\mathbf{Z}$, we first relate each estimated specification $\hat{Z}_k$ to its true counterpart $Z_{\pi(k)}$, and then use the local parent structure to restrict the possible dependence of $X_j$ on $\hat{\mathbf{X}}$. Let

$$h := g_X^{-1} \circ \hat{g}_X, \qquad r := g_Z^{-1} \circ \hat{g}_Z, \qquad A(\hat{\mathbf{x}}) := J_h(\hat{\mathbf{x}}).$$

Thus $\mathbf{X} = h(\hat{\mathbf{X}})$ and $\mathbf{Z} = r(\hat{\mathbf{Z}})$. In particular, $A_{jl}(\hat{\mathbf{x}}) = \partial h_j(\hat{\mathbf{x}})/\partial \hat{x}_l$ is the coordinate derivative $\partial X_j/\partial \hat{X}_l$. Since $h$ is a diffeomorphism,

$$J_{h^{-1}}\big(h(\hat{\mathbf{x}})\big) J_h(\hat{\mathbf{x}}) = I_n,$$

so $J_h(\hat{\mathbf{x}})$ is nonsingular at every point. By Theorem 4, $Z_{\pi(k)} = r_{\pi(k)}(\hat{Z}_k)$ with nonzero derivative. The conditional change-of-variables formula is

$$p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}}) = p(\mathbf{Z} \mid \mathbf{X}) |\det J_r(\hat{\mathbf{Z}})|.$$

Due to the structural assumptions on both the true and estimated models, we have

$$p(\mathbf{Z} \mid \mathbf{X}) = \prod_{i \in [d]} p(Z_i \mid \mathbf{V}_i), \quad p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}}) = \prod_{i \in [d]} p(\hat{Z}_i \mid \hat{\mathbf{V}}_i).$$

The derivative of $\log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}})$ with respect to any $\hat{Z}_k \in \hat{\mathbf{Z}}$ is therefore

$$\frac{\partial \log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}})}{\partial \hat{Z}_k} = \frac{\partial \log p(\hat{Z}_k \mid \hat{\mathbf{V}}_k)}{\partial \hat{Z}_k}.$$

Define

$$\hat{\eta}'_k := \frac{\partial \log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{X}})}{\partial \hat{Z}_k} = \frac{\partial \log p(\hat{Z}_k \mid \hat{\mathbf{V}}_k)}{\partial \hat{Z}_k}.$$

By making use of the component-wise identifiability of $\mathbf{Z}$, we have

$$\begin{aligned} \hat{\eta}'_k &= \frac{\partial \log p(\mathbf{Z} \mid \mathbf{X})}{\partial \hat{Z}_k} + \frac{\partial \log |\det J_r(\hat{\mathbf{Z}})|}{\partial \hat{Z}_k} \\ &= \sum_{i \in [d]} \eta'_i \frac{\partial Z_i}{\partial \hat{Z}_k} + \frac{\partial \log |\det J_r(\hat{\mathbf{Z}})|}{\partial \hat{Z}_k}. \end{aligned}$$

Because $\mathbf{Z}$ is component-wise identifiable, only when $i = \pi(k)$ do we have $\frac{\partial Z_i}{\partial \hat{Z}_k} \neq 0$. Therefore,

$$\hat{\eta}'_k = \eta'_{\pi(k)} \frac{\partial Z_{\pi(k)}}{\partial \hat{Z}_k} + \frac{\partial \log |\det J_r(\hat{\mathbf{Z}})|}{\partial \hat{Z}_k}.$$

For any $\hat{X}_l \notin \hat{\mathbf{V}}_k$, the derivative of $\hat{\eta}'_k$ with respect to $\hat{X}_l$ is zero. Since $\mathbf{Z} = r(\hat{\mathbf{Z}})$ depends only on $\hat{\mathbf{Z}}$, differentiating at fixed $\hat{\mathbf{Z}}$ gives

$$\frac{\partial Z_{\pi(k)}}{\partial \hat{X}_l} = 0, \qquad \frac{\partial^2 Z_{\pi(k)}}{\partial \hat{Z}_k \partial \hat{X}_l} = 0, \qquad \frac{\partial^2 \log |\det J_r(\hat{\mathbf{Z}})|}{\partial \hat{Z}_k \partial \hat{X}_l} = 0.$$

Therefore,

$$0 = \frac{\partial \hat{\eta}'_k}{\partial \hat{X}_l} = \sum_{j: X_j \in \mathbf{V}_{\pi(k)}} \frac{\partial \eta'_{\pi(k)}}{\partial X_j} \frac{\partial X_j}{\partial \hat{X}_l} \frac{\partial Z_{\pi(k)}}{\partial \hat{Z}_k}. \tag{32}$$

To solve this linear system and recover the relationship between $\mathbf{X}$ and $\hat{\mathbf{X}}$, we require sufficient variability in $Z_{\pi(k)}$. This is formalized in Condition C4. That is, for every $Z_k \in \mathbf{Z}$ and for every fixed value of the parent variables $\{X_j : X_j \in \mathbf{V}_{\pi(k)}\}$, there exist $|\mathbf{V}_{\pi(k)}|$ distinct values of $Z_{\pi(k)}$, i.e., $z^{(1)}, \cdots, z^{(|\mathbf{V}_{\pi(k)}|)}$, such that we obtain $|\mathbf{V}_{\pi(k)}|$ linearly independent vectors $u(\mathbf{x}, z_{\pi(k)})$, where

$$u(\mathbf{x}, z_{\pi(k)}) = \left( \frac{\partial \eta'_{\pi(k)}}{\partial X_j} \right)_{X_j \in \mathbf{V}_{\pi(k)}} \Big|_{Z_{\pi(k)} = z^{(m)}}.$$

Fix $\hat{\mathbf{x}}$, and hence $\mathbf{x} = h(\hat{\mathbf{x}})$, while varying $Z_{\pi(k)}$ over these values. The entries $\partial X_j / \partial \hat{X}_l$ remain fixed. Dividing each equation equation 32 by the nonzero factor $\partial Z_{\pi(k)} / \partial \hat{Z}_k$ gives a homogeneous linear system with these entries as its unknowns. Under Condition C4, the coefficient matrix is nonsingular, implying that

$$\frac{\partial X_j}{\partial \hat{X}_l} = 0, \qquad X_j \in \mathbf{V}_{\pi(k)}, \quad \hat{X}_l \notin \hat{\mathbf{V}}_k, \qquad \text{for every } \hat{\mathbf{x}}. \tag{33}$$

This means that for any $X_j \in \mathbf{V}_{\pi(k)}$, it cannot depend on any $\hat{X}_l \notin \hat{\mathbf{V}}_k$. Equivalently, $X_j$ can only be a function of the intersection of the parent sets of $\hat{Z}_k$ corresponding to its children. Here the Cartesian-product domain ensures that vanishing partial derivatives imply dependence only on the remaining coordinates.

Fix any feature value $\hat{\mathbf{x}}_0$ and apply Lemma 1 to the fixed matrix $A(\hat{\mathbf{x}}_0) = J_h(\hat{\mathbf{x}}_0)$. This matrix is nonsingular by the smooth invertibility of $h$, and its support restriction follows from equation equation 33 evaluated at $\hat{\mathbf{x}}_0$. Thus the existence of the matrix required by the lemma has been established, rather than assumed separately. A single feature permutation $\sigma$ therefore satisfies

$$\sigma(\mathbf{V}_{\pi(k)}) \subseteq \hat{\mathbf{V}}_k, \qquad k \in [d].$$

Condition C5 and the last part of the lemma therefore imply

$$\hat{\mathbf{V}}_k = \sigma(\mathbf{V}_{\pi(k)}), \qquad \text{and hence} \qquad |\hat{\mathbf{V}}_k| = |\mathbf{V}_{\pi(k)}|, \quad k \in [d]. \tag{34}$$

In particular, for each feature with at least one child,

$$\hat{X}_{\sigma(j)} \in \bigcap_{k: X_j \in \mathbf{V}_{\pi(k)}} \hat{\mathbf{V}}_k.$$

If this intersection is a singleton, $X_j$ depends on only that estimated coordinate. Its derivative cannot vanish, since otherwise $J_h$ would have a zero row. On the corresponding interval domain this scalar function is strictly monotone and therefore invertible onto its image.

Together, these conclusions imply the three cases stated in the theorem:

- Case 1. When $X_j$ is not a parent of any specification (i.e., $X_j$ is an isolated feature variable), the specification structure alone provides no identification guarantee for $X_j$.
- Case 2. When $X_j$ is the parent of only one specification, i.e., $Z_{\pi(k)}$, it is a function of the parent set $\hat{\mathbf{V}}_k$ of $\hat{Z}_k$. Condition C5 further restricts the possible mixing by enforcing $|\hat{\mathbf{V}}_k| = |\mathbf{V}_{\pi(k)}|$.
- Case 3. When $X_j$ is the parent of multiple specifications, it is a function of the intersection of the parent sets of the corresponding estimated specifications. This intersection contains $\hat{X}_{\sigma(j)}$ by Lemma 1. In particular, when the intersection contains only one element, $X_j$ is component-wise identifiable.

Finally, consider the example in Fig. 1(b), with parent sets $\mathbf{V}_1 = \{X_1, X_2\}$ and $\mathbf{V}_2 = \{X_2, X_3\}$. After relabeling the estimated features by the permutation $\sigma$, equations equation 33 and equation 34 give

$$X_1 = h_1(\hat{X}_1, \hat{X}_2), \qquad X_2 = h_2(\hat{X}_2), \qquad X_3 = h_3(\hat{X}_2, \hat{X}_3).$$

These statements describe how the true coordinates depend on the estimated coordinates; they do not use an inverse-Jacobian implication. The singleton intersection identifies $X_2$ component-wise: nonsingularity of $J_h$ forces $h_2'$ to be nonzero, so $h_2$ is invertible on its interval domain. The conclusions for $X_1$ and $X_3$ restrict their possible arguments, but do not imply component-wise identification of either.

The corresponding parent *blocks*, $(X_1, X_2)$ and $(X_2, X_3)$, are identifiable up to diffeomorphisms of their matched estimated blocks. To justify this reverse-direction statement, place either parent block first in the input and output coordinates. Its equal dimension and the proved zero restrictions give $J_h = \left(\begin{smallmatrix} B & 0 \\ C & D \end{smallmatrix}\right)$ with square diagonal blocks. Nonsingularity makes $B, D$ invertible, and $J_{h^{-1}}$ has the same upper-right zero block. On the true product domain, the inverse parent block therefore depends only on the matched true parent block; the global inverse identities give a parent-block diffeomorphism. This uses the whole block structure, not entrywise reciprocity of Jacobian zeros. The allowed transformations are generally nonlinear, so these are parent-coordinate blocks rather than linear spans identifying $X_1$ or $X_3$ individually. □

**Corollary 1** (Identification of the conditional laws)**.** *Under the assumptions of Theorem 5, let $\pi, \sigma$ be the permutations obtained in its proof, and let $V_i, \hat{V}_k$ denote parent index sets. Write $Z_{\pi(k)} = r_k(\hat{Z}_k)$ and $\mathbf{X}_{V_{\pi(k)}} = H_k(\hat{\mathbf{X}}_{\hat{V}_k})$ for the corresponding scalar and parent-block diffeomorphisms, with $\hat{V}_k = \sigma(V_{\pi(k)})$. Then, on the population support,*

$$\hat{p}_k(\hat{z}_k \mid \hat{\mathbf{v}}) = p_{\pi(k)}\big(r_k(\hat{z}_k) \mid H_k(\hat{\mathbf{v}})\big)|r_k'(\hat{z}_k)|. \tag{35}$$

*Thus the conditional laws are identified up to the matched coordinate changes; the maps $H_k$ agree on overlapping outputs because they are restrictions of the same feature-coordinate map.*

*Proof.* Parent-set equality and the block-invertibility argument in the preceding proof give the maps $H_k$ for every matched parent block. Applying scalar conditional change of variables to $Z_{\pi(k)} = r_k(\hat{Z}_k)$ gives the stated identity, as in equation 24. □

**Scope: exclusion of the linear Gaussian case by C3.**

**Corollary 2** (Linear Gaussian conditionals do not satisfy C3)**.** *For the linear Gaussian model*

$$Z_i = b_i^\top \mathbf{X} + E_i, \qquad E_i \sim N(0, \sigma_i^2), \qquad \sigma_i^2 > 0,$$

*with innovations independent of $\mathbf{X}$ and constant conditional variances, condition C3 cannot hold. This case is therefore outside the scope of Theorems 4 and 5.*

*Proof.* Writing $\eta_i(z_i, \mathbf{x}) = \log p(z_i \mid \mathbf{x})$ gives

$$\partial_{z_i}\eta_i = -\frac{z_i - b_i^\top \mathbf{x}}{\sigma_i^2}, \qquad \partial_{z_i}^2\eta_i = -\frac{1}{\sigma_i^2}.$$

For every fixed $\mathbf{z}$, the $d$ second-derivative entries of $w(\mathbf{z}, \mathbf{x})$ are constant in $\mathbf{x}$. They therefore vanish in every difference $w(\mathbf{z}, \mathbf{x}^{(m)}) - w(\mathbf{z}, \mathbf{x}^{(0)})$. These differences span a space of dimension at most $d$, whereas C3 requires $2d$ linearly independent vectors. Thus C3 fails. □

**General nonlinear, nonparametric setting.** The theorem does not prescribe a parametric family for the conditional laws or require linear observation maps. It applies to the general nonlinear, nonparametric model class under the stated smoothness, invertibility, conditional-factorization, variability, and sparsity assumptions C1–C5. These assumptions can hold simultaneously in nonlinear models; nonlinearity alone does not imply them. In particular, a Gaussian conditional with feature-dependent variance need not have a constant second log-density derivative as the features vary and is not excluded by the preceding argument.

# E Appendix of Structured Extrapolative Data Generation Algorithms and Other Details

## E.1 Likelihood-weighted Resampling and Likelihood-guided Diffusion Algorithms

**Sampling conventions.** All algorithms operate in model coordinates; normalization and any representation maps are applied to targets consistently and reversed on returned samples when a reconstruction map is available. For DDPM sampling, let $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, $\bar{\alpha}_0 = 1$, and $\widetilde{\beta}_t = \beta_t(1 - \bar{\alpha}_{t-1})/(1 - \bar{\alpha}_t)$. We write

$$\mu_\theta(u_t, t; c) = \frac{1}{\sqrt{\alpha_t}} \left[ u_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(u_t, t; c) \right].$$

Unconditional models omit $c$. Gaussian initialization is the usual finite-horizon approximation. The notation $\mathrm{DDPMSample}(p; c)$ denotes a complete conditional DDPM draw using the model's fitted schedule.

**Algorithm 1** Likelihood-weighted resampling (LWR)

**Input:** target $(z_1^\star, z_2^\star)$; fitted anchor $a_\theta$, likelihood $q_\phi$, and remaining-feature conditional $b_\eta$; candidate count $K$; output count $N$.
**Output:** $N$ generated feature samples.

1. For $k = 1, \ldots, K$, draw $(x_1^{(k)}, x_2^{(k)}) \leftarrow \mathrm{DDPMSample}(a_\theta; z_1^\star)$ and compute $\ell_k \leftarrow \log q_\phi(z_2^\star \mid x_2^{(k)})$.
2. Normalize the weights in log space: $w_k \leftarrow \exp\{\ell_k - \mathrm{logsumexp}_{j=1}^{K} \ell_j\}$. If all likelihoods are zero, report insufficient candidate coverage; no normalized resampling distribution is defined.
3. For $j = 1, \ldots, N$:
   (a) Draw $I_j \sim \mathrm{Categorical}(w_1, \ldots, w_K)$ with replacement and retain the entire pair $(x_1, x_2) \leftarrow (x_1^{(I_j)}, x_2^{(I_j)})$.
   (b) Draw $x_3 \leftarrow \mathrm{DDPMSample}(b_\eta; (x_2, z_2^\star))$.
   (c) Return $(x_1, x_2, x_3)$ as the $j$th sample.

**Algorithm 2** Likelihood-guided diffusion (LGD)

**Input:** target $(z_1^\star, z_2^\star)$; frozen anchor DDPM $a_\theta$; trained noise-aware density $r_\psi(Z_2 \mid A_t, Z_1, t)$; remaining-feature DDPM $b_\eta$; schedule $\{\beta_t\}_{t=1}^{T}$.
**Output:** one generated feature sample.

1. Draw $u_T \sim \mathcal{N}(0, I)$ in the joint $(X_1, X_2)$ space.
2. For $t = T, \ldots, 1$:
   (a) Compute the anchor mean $\mu_t \leftarrow \mu_\theta(u_t, t; z_1^\star)$.
   (b) With input gradients enabled, compute $g_t \leftarrow \nabla_{u_t} \log r_\psi(z_2^\star \mid u_t, z_1^\star, t)$.
   (c) Apply the unit-strength score correction: $\mu_t^{\mathrm{guided}} \leftarrow \mu_t + \beta_t g_t / \sqrt{\alpha_t}$.
   (d) Draw $\xi_t \sim \mathcal{N}(0, I)$ if $t > 1$, otherwise set $\xi_t = 0$, and update $u_{t-1} \leftarrow \mu_t^{\mathrm{guided}} + \sqrt{\widetilde{\beta}_t}\, \xi_t$.
3. Unpack $u_0 = (x_1, x_2)$; draw $x_3 \leftarrow \mathrm{DDPMSample}(b_\eta; (x_2, z_2^\star))$ and return $(x_1, x_2, x_3)$.

**Training the LGD predictor.** Freeze $a_\theta$ and $q_\phi$. Sample $z_1$ from a design distribution over the observed training range, $u_0 \sim a_\theta(\cdot \mid z_1)$, and $z_2 \sim q_\phi(\cdot \mid x_{2,0})$. Draw $t$ uniformly from $\{1, \ldots, T\}$ and $\epsilon \sim \mathcal{N}(0, I)$, and set $u_t = \sqrt{\bar{\alpha}_t} u_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$. Fit $r_\psi$ by minimizing $-\mathbb{E} \log r_\psi(z_2 \mid u_t, z_1, t)$

over these synthetic examples. No extrapolation-test observations enter this procedure. The mean shift in Algorithm 2 follows from $\epsilon_{\text{guided}} = \epsilon_\theta - \sqrt{1-\bar{\alpha}_t}\, g_t$; it is not a generic posterior-variance-times-gradient update. Multiplying $g_t$ by an optional scale changes the score correction, so unit strength is the nominal learned-composition target.

**Algorithm 3** Blockwise structural approximation with LWR or LGD

**Input:** training observations; target specifications; observed representations or a representation-learning procedure; supplied or estimated feature–specification graph; sampler $\mathcal{M} \in \{\text{LWR}, \text{LGD}\}$.
**Output:** generated feature samples, optionally reconstructed in observation space.

1. Use observed $(\mathbf{X}, \mathbf{Z})$ when compatible; otherwise learn a canonical representation under the stated assumptions. If sufficient, transform only $\mathbf{Z}$. Express the target in the resulting specification coordinates. Use the supplied compatible graph or estimate it from training data. Let $P_i \subseteq \{1, \dots, d\}$ be the parent-index set for specification $Z_i$.
2. For $n \geq 2$, partition specification indices into two nonempty disjoint groups $J_1, J_2$ with $|J_1| = \lfloor n/2 \rfloor$ and $|J_2| = \lceil n/2 \rceil$, using a fixed ordering or seeded rule. Set $\mathbf{Z}^{(1)} = \mathbf{Z}_{J_1}$ and $\mathbf{Z}^{(2)} = \mathbf{Z}_{J_2}$.
3. Form the group parent sets $P^{(1)} = \bigcup_{i \in J_1} P_i$ and $P^{(2)} = \bigcup_{i \in J_2} P_i$. Set $B_2 = P^{(1)} \cap P^{(2)}$, $B_1 = P^{(1)} \setminus P^{(2)}$, and $B_3 = P^{(2)} \setminus P^{(1)}$.
4. Let $R = \{1, \dots, d\} \setminus (P^{(1)} \cup P^{(2)})$ be the feature indices with no specification children. Partition $R$ into nearly equal disjoint sets $R_1, R_3$ using a fixed rule, and replace $B_1 \leftarrow B_1 \cup R_1$, $B_3 \leftarrow B_3 \cup R_3$. Define $\mathbf{X}^{(j)} = \mathbf{X}_{B_j}$.
5. Fit the block models on training data:

$$a_\theta(\mathbf{X}^{(1)}, \mathbf{X}^{(2)} \mid \mathbf{Z}^{(1)}), \qquad q_\phi(\mathbf{Z}^{(2)} \mid \mathbf{X}^{(2)}), \qquad b_\eta(\mathbf{X}^{(3)} \mid \mathbf{X}^{(2)}, \mathbf{Z}^{(2)}).$$

   For LGD, additionally train $r_\psi(\mathbf{Z}^{(2)} \mid A_t, \mathbf{Z}^{(1)}, t)$ using synthetic block examples, with $A_0 = (\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$.
6. Apply Algorithm 1 or 2, replacing scalar variables and targets by their blocks. Restore the original feature ordering and, when available, apply the reconstruction map.

**Scope of the blockwise approximation.** The construction partitions the feature coordinates and preserves their estimated cross-layer parent memberships, but does not establish the conditional independences needed for the block factorization. Dependencies between feature blocks, specification residuals, or features with no specification children can invalidate it; assigning such features to either private block does not remove these dependencies. The grouping is therefore a practical approximation unless the required conditions are separately established. The block likelihood is a *joint* density of $\mathbf{Z}^{(2)}$; replacing it by a product of scalar densities requires additional conditional-independence assumptions. Empty feature blocks are omitted, and a likelihood conditioned on an empty shared block is unconditional. For two target properties, each specification block contains one property. Target coordinates in a learned representation must have a defined semantic mapping; representation learning alone does not supply that mapping.

**Algorithm 4** Structure-informed DPS-style sampling with a joint prior

**Input:** target specifications (values or a specification observation); joint prior DDPM $p_\theta(\mathbf{X})$; fitted Gaussian likelihood heads and parent sets or masks; optional representation encoders and feature decoder; diffusion schedule; guidance step sizes $\rho_t \geq 0$.
**Output:** one generated feature sample. All quantities below use consistent model coordinates.

1. Use observed feature/specification coordinates when given. Otherwise apply fitted representation maps for the unavailable variables and encode the target specification observation. Use the prior trained on the resulting feature coordinates. Normalize targets consistently with likelihood training to obtain $\mathbf{z}^\star$.
2. Draw $\mathbf{x}_T \sim \mathcal{N}(0, I)$.
3. For $t = T, \ldots, 1$, enable gradients with respect to $\mathbf{x}_t$:
   (a) Predict noise $\widehat{\epsilon}_t \leftarrow \epsilon_\theta(\mathbf{x}_t, t)$ and compute the denoised estimate
   $$\widehat{\mathbf{x}}_{0|t} \leftarrow \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\widehat{\epsilon}_t}{\sqrt{\bar{\alpha}_t}}.$$
   (b) Combine all likelihood terms at the denoised estimate:
   $$\mathcal{L}_t \leftarrow -\sum_{i=1}^{n} \log \ell_i\big(z_i^\star \mid \widehat{\mathbf{v}}_{i,t}\big), \qquad g_t \leftarrow \nabla_{\mathbf{x}_t} \mathcal{L}_t.$$
   Here $\widehat{\mathbf{v}}_{i,t}$ is the parent-restricted or masked denoised estimate. Differentiate through both Gaussian heads and $\widehat{\mathbf{x}}_{0|t}$, including the denoiser; model parameters and masks remain fixed.
   (c) Compute $\mu_t \leftarrow \mu_\theta(\mathbf{x}_t, t)$. Draw $\xi_t \sim \mathcal{N}(0, I)$ if $t > 1$, otherwise set $\xi_t = 0$.
   (d) Update $\mathbf{x}_{t-1} \leftarrow \mu_t + \sqrt{\widetilde{\beta}_t}\,\xi_t - \rho_t g_t$.
4. Reverse feature normalization. Return features directly when observed; otherwise decode the generated feature representation into observation space using the fitted feature decoder.

**Representation and likelihood estimation.** When $\mathbf{X}$ and $\mathbf{Z}$ are given, we fit the structured likelihood directly in their observed coordinates. Otherwise, we learn representations for the unavailable variables from their corresponding observations using reconstruction, structured-likelihood, and sparsity objectives (Appendix D). The joint diffusion prior is trained on the resulting observed or learned feature coordinates. For latent specifications, the target observation is encoded using a fixed convention, such as the encoder mean; this plug-in procedure does not integrate encoder uncertainty. A feature decoder is needed only when generating observations from learned features. Let $\mathbf{v}_i = \mathbf{x}_{P_i}$ for a known parent set, or $\mathbf{v}_i = \mathbf{x} \odot \mathbf{M}_i$ for a learned mask. We parameterize

$$\ell_i(z_i \mid \mathbf{v}_i) = \mathcal{N}\big(z_i; h_i^\mu(\mathbf{v}_i), [h_i^\sigma(\mathbf{v}_i)]^2\big), \qquad h_i^\sigma(\mathbf{v}_i) > 0,$$

where $h_i^\sigma$ denotes a standard deviation. Its negative log-likelihood is

$$-\log \ell_i(z_i \mid \mathbf{v}_i) = \frac{(z_i - h_i^\mu(\mathbf{v}_i))^2}{2[h_i^\sigma(\mathbf{v}_i)]^2} + \log h_i^\sigma(\mathbf{v}_i) + \tfrac{1}{2}\log(2\pi).$$

We fit the heads by minimizing the summed training negative log-likelihood, adding a sparsity penalty when learning masks. All heads and masks remain fixed during generation. The product $L(\mathbf{z} \mid \mathbf{x}) = \prod_i \ell_i(z_i \mid \mathbf{v}_i)$ specifies a conditionally independent model of the specifications given features. Fitting this product and penalizing masks does not itself establish that the data satisfy the required conditional independences or that the representation is identifiable; those conclusions require the separate assumptions and results in the representation analysis.

**DPS likelihood guidance.** Algorithm 4 uses a DPS-style plug-in approximation Chung et al. (2022), replacing the noise-aware likelihood with $L(\mathbf{z}^\star \mid \widehat{\mathbf{x}}_{0|t})$. Its guidance loss is

$$\mathcal{L}_t = \sum_i \left[ \frac{(z_i^\star - h_i^\mu(\widehat{\mathbf{v}}_{i,t}))^2}{2[h_i^\sigma(\widehat{\mathbf{v}}_{i,t})]^2} + \log h_i^\sigma(\widehat{\mathbf{v}}_{i,t}) \right],$$

where $\widehat{\mathbf{v}}_{i,t}$ is the parent-restricted or masked denoised feature estimate. We differentiate through the conditional mean and standard deviation, the fixed mask, and the denoiser with respect to $\mathbf{x}_t$. The log-scale term cannot be dropped when the variance depends on features; ordinary squared-error guidance is a special case with fixed common variance. Predictions and targets must use the same normalization. Combining terms in one objective avoids a separate noise-aware predictor but does not eliminate likelihood-estimation or plug-in approximation errors.

**Limitation imposed by the joint prior.** The ideal posterior is $\pi(\mathbf{x} \mid \mathbf{z}^\star) \propto p_\theta(\mathbf{x})L(\mathbf{z}^\star \mid \mathbf{x})$. Provided its normalizer is finite and positive, it is absolutely continuous with respect to the learned joint prior: a set with zero prior probability also has zero posterior probability. Likelihood conditioning alone therefore cannot create mass in a prior-null extrapolation region. This applies whether feature coordinates are observed or learned; using the same decoder for the prior and posterior preserves this restriction in observation space. It does not imply that approximate DPS iterates remain within the empirical training region. A learned diffusion prior may assign positive probability beyond that region, and approximate guidance need not sample the ideal posterior exactly. Exact LWR/LGD composition has the analogous anchor-support restriction, but its structural anchor may cover combinations absent from a jointly fitted training-distribution prior.

### E.2 Details on Learning Noise-aware Likelihood Guidance

Let $A_0 = (X_1, X_2)$ and define the reweighted anchor $\pi_0(a_0 \mid z_1, z_2) \propto a_\theta(a_0 \mid z_1)q_\phi(z_2 \mid x_{2,0})$. Applying the same forward diffusion kernel to $\pi_0$ and the anchor gives $\pi_t(a_t \mid z_1, z_2) \propto a_{\theta,t}(a_t \mid z_1)h_t(z_2 \mid a_t, z_1)$, where

$$h_t(z_2 \mid a_t, z_1) = \mathbb{E}_{A_0 \sim a_\theta(\cdot \mid A_t = a_t, z_1)}[q_\phi(z_2 \mid X_{2,0})]\,.$$

Consequently, the exact score correction is $\nabla_{a_t} \log h_t(z_2 \mid a_t, z_1)$. We approximate $h_t$ using a conditional density predictor $r_\psi(Z_2 \mid A_t, Z_1, t)$. With the anchor and clean likelihood frozen, we draw $Z_1$ over its observed training range, $A_0 \sim a_\theta(\cdot \mid Z_1)$, $Z_2 \sim q_\phi(\cdot \mid X_{2,0})$, and $A_t$ from the forward diffusion kernel, and minimize $-\mathbb{E}[\log r_\psi(Z_2 \mid A_t, Z_1, t)]$. The population optimum equals $h_t$ wherever the training distribution provides coverage, assuming sufficient model capacity. During sampling, we add $\nabla_{a_t} \log r_\psi(z_2^\star \mid a_t, z_1^\star, t)$ to the anchor score. Although the clean likelihood depends only on $X_2$, its noise-aware counterpart generally depends on both coordinates of $A_t$, enabling joint updates of $X_1$ and $X_2$. This correction targets the learned reweighted anchor; likelihood estimation, predictor approximation, and finite-step diffusion sampling can introduce error.

## F Appendix on Experiment Details and Additional Results

All the code can be found here: https://anonymous.4open.science/status/SEDGE-25BB.

### F.1 Synthetic Experiments

#### F.1.1 Data: Synthetic Data Generation Processes

Write $C = X_2$ for the shared bridge. All root coordinates are mutually independent before selection. Binary roots are uniform on $[-1, 1]$; continuous roots are uniform on $[0, 1]$. $S = 1$ denotes retention for training. Only a joint specification event is removed; individual specification support remains present.

For continuous settings, $\mathcal{R} = I_1 \times I_2$ is the held-out rectangle and $\mathcal{S}_Z$ is the mechanism-induced joint support. Training uses $\mathcal{S}_Z \setminus \mathcal{R}$, not a uniform distribution over the marginal ranges. Each rectangle has side length $0.003$, with no additional exclusion buffer.

- **Weighted Discrete (binary multimodal).**
  - **Mechanisms:**

    $$Z_1 = \mathbf{1}\{C \in A,\ |X_1| \le 0.25\}, \qquad Z_2 = \mathbf{1}\{C \in B,\ |X_3| \le h(C)\},$$

    where $A = [-1, -0.4] \cup [0.4, 1]$, $B = [-0.5, 0.5]$, and $h(C) = 0.05$ for $C < 0$, otherwise $0.25$.

  - **Training:** $(Z_1, Z_2) \in \{(0,0),(1,0),(0,1)\}$; both marginal supports remain $\{0,1\}$.
  - **Testing:** $(Z_1, Z_2) = (1,1)$. The negative and positive bridge modes have posterior masses $1/6$ and $5/6$, respectively.
- **Linear (scalar continuous).**
  - **Mechanisms:** $Z_1 = 0.8X_1 + 0.6C$ and $Z_2 = 0.8X_3 + 0.6C$.
  - **Training:** Both marginal supports remain $[0, 1.4]$.
  - **Testing:** $I_1 = [0.3135, 0.3165]$ and $I_2 = [1.0835, 1.0865]$.
- **Cubic (scalar continuous).**
  - **Mechanisms:** $Z_1 = 0.8X_1 + 0.6C$ and $Z_2 = 0.8X_3 + 0.6C^3$.
  - **Training:** Both marginal supports remain $[0, 1.4]$.
  - **Testing:** $I_1 = [0.3135, 0.3165]$ and $I_2 = [0.862803125, 0.865803125]$.
- **Smoothstep (scalar continuous).**
  - **Mechanisms:** $Z_1 = 0.8X_1 + 0.6C$ and $Z_2 = 0.8X_3 + 0.6(3C^2 - 2C^3)$.
  - **Training:** Both marginal supports remain $[0, 1.4]$.
  - **Testing:** $I_1 = [0.3135, 0.3165]$ and $I_2 = [1.07601875, 1.07901875]$.
- **2D Bridge (vector continuous).**
  - **Mechanisms:** With $C = (u, v)$,

$$Z_1 = 0.7u + 0.3v + (0.8 + 0.15uv)X_1,$$
$$Z_2 = 1 - 0.3u - 0.7v + [0.8 + 0.15(1-u)(1-v)]X_3.$$

  - **Training:** Both marginal supports remain $[0, 1.95]$.
  - **Testing:** $I_1 = I_2 = [0.3485, 0.3515]$.

**Sample counts.** Each seed draws a 200,000-root candidate pool, then selects 20,000 retained training rows and 2,000 retained validation rows. There are three seeds. Binary testing uses 2,000 conditional reference samples at $(1, 1)$. Continuous testing uses 64 held-out target pairs, with 32 reference draws per target (2,048 outputs). Continuous targets are drawn from the population conditional on the held-out rectangle; they are not assumed uniform in that rectangle.

**Bridge compatibility and selection audit.** For a fixed target $z$, define $F_i(z_i) = \{c : \exists x_{\text{private}} \text{ in root support with } f_i(x_{\text{private}}, c) = z_i\}$. Thus compatibility means that a private root *can* satisfy the specification; it is not sufficient for an arbitrary generated private root to satisfy it.

The following bridge compatibility sets are evaluated at the test-rectangle center, or at $(Z_1, Z_2) = (1, 1)$ for binary data. The reduction ratio is the joint measure divided by the smaller individual measure.

- **Weighted Discrete.**
  - **Individual compatibility:** $Z_1$ permits $C \in [-1, -0.4] \cup [0.4, 1]$; $Z_2$ permits $C \in [-0.5, 0.5]$.
  - **Joint compatibility:** $C \in [-0.5, -0.4] \cup [0.4, 0.5]$.
  - **Measure reduction:** Individual lengths $1.2$ and $1.0$ reduce to joint length $0.2$, giving a ratio of $0.20$.
- **Linear, Cubic, and Smoothstep.** These settings share the same center overlap, isolating changes in the mechanism.
  - **Individual compatibility:** $Z_1$ permits $C \in [0, 0.525]$; $Z_2$ permits $C \in [0.475, 1]$.
  - **Joint compatibility:** $C \in [0.475, 0.525]$.
  - **Measure reduction:** Both individual lengths are $0.525$, whereas the joint length is $0.05$, giving a ratio of approximately $0.09524$.
- **2D Bridge.**
  - **Individual compatibility:**

    $$F_1 = \{(u, v) \in [0, 1]^2 : 0.7u + 0.3v \leq 0.35\}, F_2 = \{(u, v) \in [0, 1]^2 : 0.3u + 0.7v \geq 0.65\}.$$

  - **Joint compatibility:** The polygon intersection $F_1 \cap F_2$ has coordinate projections $u \in [0, 0.125]$ and $v \in [0.875, 1]$; it is **not their Cartesian product**.
  - **Measure reduction:** Both individual areas are $2/7$, whereas the joint area is $1/112$, giving a ratio of $1/32 = 0.03125$.

**Away from the center.** For the three scalar continuous mechanisms, let $g(c) = c$, $c^3$, or $3c^2 - 2c^3$. Throughout the specified test rectangle,

$$F_1(z_1) = [0, z_1/0.6], \quad F_2(z_2) = [g^{-1}((z_2 - 0.8)/0.6), 1], \quad F_1 \cap F_2 = [g^{-1}((z_2 - 0.8)/0.6), z_1/0.6].$$

The joint width ranges are $[0.045, 0.055]$ (Linear), approximately $[0.043835, 0.056223]$ (Cubic), and $[0.045829, 0.054171]$ (Smoothstep). For 2D, replace $0.35$ by $z_1$ and $0.65$ by $1 - z_2$ in the inequalities; the audited joint area varies approximately from $0.00840$ to $0.00947$.

Table 3: Exclusion rates (%). The audit requires both conditional rates to be at most $1\%$. Continuous rates condition on test bands, not individual specification values. Weighted Discrete is retained as a relaxed reference.

| Setting | $P(S{=}0 \mid E_1)$ | $P(S{=}0 \mid E_2)$ | $P(S{=}0)$ | Audit |
|---|---|---|---|---|
| Weighted Discrete | 2.50 | 5.00 | 0.375 | Fail |
| Linear | 0.0357 | 0.0357 | $7.03 \times 10^{-5}$ | Pass |
| Cubic | 0.0357 | 0.0357 | $7.03 \times 10^{-5}$ | Pass |
| Smoothstep | 0.0357 | 0.0357 | $7.03 \times 10^{-5}$ | Pass |
| 2D Bridge | 0.0116 | 0.0116 | $1.23 \times 10^{-5}$ | Pass |

These continuous conditional rates are band averages, not verified bounds on $\sup_{z_i \in I_i} P(S = 0 \mid Z_i = z_i)$. Root independence holds before selection; selection can distort the observed factors. Small exclusion and retained marginal support do not prove an undistorted likelihood or accurate learned models. Geometric compatibility above uses exact target equations; the result evaluator separately allows $|f_i(X) - z_i| \leq 0.005$ on each continuous specification and also requires root support.

### F.1.2 Implementation & Metrics: Details

**Training protocol.** Each setting uses 20,000 training samples and 2,000 validation samples, with results averaged over three seeds. Features and specifications are standardized using training-set statistics; binary likelihood targets remain binary. Diffusion models use residual MLPs with hidden width 192, three residual blocks, SiLU activations, and 64-dimensional sinusoidal time embeddings. Each diffusion component is trained for 20,000 updates using the noise-prediction MSE objective, AdamW with learning rate $3 \times 10^{-4}$, batch size 512, zero weight decay, and gradient-norm clipping at 5. We maintain an exponential moving average with decay $0.999$ and select the checkpoint with the lowest validation noise-prediction loss, evaluated every 250 updates. Training uses a fixed budget rather than early stopping.

**Conditional generation and guidance baselines.** All diffusion samplers use 100 DDPM steps with a linear variance schedule from $\beta_1 = 10^{-4}$ to $\beta_{100} = 0.02$. Direct conditional diffusion models $p(X \mid Z_1, Z_2)$. Compositional CD combines separately trained conditional and unconditional scores as $s_t(X_t \mid Z_1) + s_t(X_t \mid Z_2) - s_t(X_t)$. CG uses an unconditional diffusion prior and a learned noisy-state likelihood. We compare sparse factorized likelihoods, whose factors see their respective noisy parent coordinates; dense factorized likelihoods, whose factors see all noisy coordinates; and a dense autoregressive joint likelihood. Guidance scales $0.5$, $1$, and $2$ are evaluated, with scales $1$ and $2$ shown for CG in the main table. Guidance gradients are clipped to Euclidean norm 50 before multiplication by the guidance scale.

**SEDGE implementation.** SEDGE learns an anchor diffusion $p(X_1, C \mid Z_1)$, a likelihood $p(Z_2 \mid C)$, and a completion diffusion $p(X_3 \mid C, Z_2)$, where $C = X_2$ may be multidimensional. SEDGE-LWR draws 1,024 anchor proposals per target, normalizes their likelihood weights, resamples one proposal, and generates the private completion. The unweighted ablation uses uniform proposal weights. SEDGE-LGD instead guides anchor diffusion using $\nabla_{(X_1,C)_t} \log q_\phi(Z_2 \mid (X_1, C)_t, Z_1, t)$, then applies the same completion model. Its noisy-state likelihood is trained by corrupting training anchors and sampling fresh $Z_2$ labels from the frozen clean likelihood. Thus, LWR and LGD share the clean likelihood model, but LGD additionally requires a noisy-state approximation.

**Likelihood models.** Continuous clean likelihoods use conditional rational-quadratic spline flows with 16 bins and three width-128 SiLU hidden layers. Their bounded supports are estimated from training data using polynomial lower and upper envelopes; this is an explicit modeling bias. Noisy-state continuous likelihoods use 16-bin spline flows with a Gaussian base, learned conditional location and scale, and identity tails outside $[-4, 4]$ in the location–scale-transformed coordinate. Binary likelihoods instead use Bernoulli networks with two width-128 SiLU hidden layers. Likelihood models are trained for 10,000 updates using AdamW with learning rate $10^{-3}$, batch size 512, zero weight decay, and gradient clipping at 5. Checkpoints are selected every 100 updates using validation negative log-likelihood. For bounded flows, selection uses supported validation observations, with out-of-support counts reported separately.

**Paper-inspired architectural controls.** The additive and fixed-composition adaptations use two separate two-hidden-layer, width-64 Softplus MLPs. Their outputs are combined by addition or by the fixed operation $\sigma(a) \odot a + \sigma(-a) \odot b$, respectively. They are trained using MSE and Adam with learning rate $10^{-3}$, batch size 256, and 3,000 updates, selecting checkpoints by validation MSE every 100 updates. Stochastic variants add centered training residuals sampled independently of the target. The Rep4Ex-inspired adaptation uses a two-dimensional autoencoder with reconstruction, maximum-moment-restriction, and covariance penalties weighted by $1$, $10$, and $0.1$, followed by additive control-function regression; each stage receives 3,000 updates. Its multimodal-discrete implementation uses a $10^{-4}$ ridge term to stabilize affine fitting. These are adaptations rather than faithful reproductions, and total parameter counts and training costs are not matched across methods.

**Evaluation protocol and validity.** Binary evaluation uses 2,000 samples at the held-out specification pair. Continuous evaluation uses 64 held-out target pairs with 32 samples per target, totaling 2,048 samples. Reference samples are generated from the corresponding conditional DGP. Validity is the percentage of finite generated samples that lie within the root support and satisfy both specifications. Binary specifications must match exactly; continuous specifications require $|f_i(X) - z_i| \leq 0.005$ for both $i = 1, 2$. Generated samples are not clipped or filtered using oracle validity.

**Joint MMD.** We compare generated and reference pairs $y = (\widetilde{X}, \widetilde{Z})$ using training-standardized coordinates. The kernel averages four Gaussian kernels,

$$k(y, y') = \frac{1}{4} \sum_{a \in \{0.5,1,2,4\}} \exp\left( -\frac{\|y - y'\|^2}{2(a\sigma)^2} \right),$$

where $\sigma$ is the median pairwise distance among the first 512 standardized training pairs. We report the square root of the nonnegative, biased V-statistic estimate of MMD squared, including diagonal terms. Kernel parameters are shared across methods within each setting and seed. All finite outputs are included, not only valid samples; full-panel MMD is unavailable if any output is nonfinite.

**Reference coverage and aggregation.** Coverage is computed in training-standardized $X$ space. Its radius $r$ is the 95th percentile of reference-to-reference nearest-neighbor distances, excluding each point itself. For generated samples $\{\widehat{x}_j\}_{j=1}^m$ and reference samples $\{x_i^\star\}_{i=1}^n$, coverage is

$$\text{Coverage} = \frac{100}{n} \sum_{i=1}^{n} \mathbf{1}\left\{ \min_j \|\widetilde{x}_i^\star - \widetilde{\widehat{x}}_j\|_2 \le r \right\}.$$

The same reference panel and radius are used for all methods within each setting and seed. Table entries are arithmetic means of seed-level metrics. Validity, joint MMD, and coverage are interpreted together to distinguish constraint satisfaction, distributional fidelity, and diversity.

### F.1.3 Method: Likelihood Estimation Comparisons

**Preliminary likelihood comparison.** Before adopting the controlled tight-overlap splits, we compared GMM, learned-uniform, and bounded-flow likelihoods within SEDGE-LWR on three earlier extrapolation settings: linear, nonlinear additive, and multiplicative. The likelihood variants reuse the same anchor proposals and sampling randomness. Flow achieves the highest validity in all three settings, while GMM obtains slightly lower joint MMD on the nonlinear and multiplicative settings. These results motivate a flexible likelihood model without implying that flow is uniformly superior.

Table 4: Preliminary likelihood comparison: mean ± sample standard deviation over five seeds. Validity is reported in percent.

| | Linear | | Nonlinear additive | | Multiplicative | |
|---|---|---|---|---|---|---|
| Method | Validity ↑ | J-MMD ↓ | Validity ↑ | J-MMD ↓ | Validity ↑ | J-MMD ↓ |
| CD | $94.28 \pm 0.64$ | $0.0219 \pm 0.0036$ | $63.15 \pm 2.84$ | $0.2683 \pm 0.0171$ | $75.35 \pm 1.92$ | $0.1857 \pm 0.0098$ |
| SEDGE-LWR (GMM) | $94.63 \pm 0.64$ | $0.0197 \pm 0.0020$ | $82.39 \pm 0.69$ | $\mathbf{0.1102} \pm 0.0069$ | $85.09 \pm 1.68$ | $\mathbf{0.0706} \pm 0.0039$ |
| SEDGE-LWR (learned uniform) | $94.69 \pm 0.77$ | $0.0223 \pm 0.0034$ | $79.30 \pm 1.13$ | $0.1496 \pm 0.0060$ | $63.87 \pm 19.67$ | $0.1325 \pm 0.0809$ |
| SEDGE-LWR (flow) | $\mathbf{94.73} \pm 0.58$ | $\mathbf{0.0185} \pm 0.0015$ | $\mathbf{83.59} \pm 1.00$ | $0.1117 \pm 0.0039$ | $\mathbf{87.80} \pm 0.83$ | $0.0735 \pm 0.0046$ |

### F.1.4 Results & Sensitivity Study: Full Table 1 with Standard Deviations and LGD Scale Sensitivity

See the vertical extended Table 5 with standard deviation on three different random seeds. Different scales during sampling for both CG and SEDGE-LGD are compared.

### F.1.5 Estimation Reliability & Extrapolated Shared Feature Compatibility

We first assess the three estimated components with observed variables and known structure. The anchor is evaluated at extrapolation-relevant $Z_1$ values, while the private decoder receives ground-truth posterior bridge values to isolate its error from upstream estimation. Likelihood total variation (TV) measures discrepancy from the ground-truth conditional distribution at validation contexts; for binary specifications, it equals absolute probability error. Table 6 reports means and sample standard deviations across three seeds. High component-wise satisfaction rates and low likelihood TV support the accuracy of these estimates on the evaluated contexts, although they do not establish likelihood accuracy under extrapolation.

We next examine whether the generated shared feature $X_2$ permits the requested specifications. Compatibility with $Z_i$ means that some private-variable value within the root support can satisfy $Z_i$ at

Table 5: Generation with observed variables and known structure: mean and sample standard deviation across three seeds. Each entry shows the mean above its ± standard deviation. Validity and coverage are percentages. Bold marks the best mean within each DGP and metric.

| Method | Weighted Discrete | | | Linear | | | Cubic | | | Smoothstep | | | 2D Bridge | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ | Valid. ↑ | J-MMD ↓ | Cover. ↑ |
| *Conditional diffusion (generation direction $Z \to X$)* | | | | | | | | | | | | | | | |
| CD | 13.55 ± 2.94 | 0.1467 ±0.0713 | 40.00 ± 5.06 | 83.22 ± 4.51 | 0.0025 ±0.0019 | 78.52 ± 6.64 | 78.12 ± 8.62 | 0.0062 ±0.0036 | 74.35 ± 3.97 | 85.55 ± 3.85 | 0.0034 ±0.0018 | 70.49 ± 5.44 | 80.65 ± 7.18 | 0.0072 ±0.0038 | 73.08 ± 3.27 |
| Compositional CD | 75.50 ± 1.96 | 0.1065 ±0.0072 | 84.78 ± 3.61 | 38.54 ± 1.76 | 0.0139 ±0.0014 | 31.25 ± 4.37 | 31.74 ± 1.32 | 0.0368 ±0.0061 | 27.67 ± 4.42 | 21.21 ± 3.04 | 0.0497 ±0.0037 | 16.06 ± 5.35 | 20.93 ± 2.47 | 0.1054 ±0.0010 | 20.04 ± 3.64 |
| *Classifier-guided diffusion (joint-$X$ prior; likelihood $X_t \to Z$)* | | | | | | | | | | | | | | | |
| CG: sparse factorized ($s = 1$) | 46.93 ± 7.34 | 0.1027 ±0.0106 | 74.85 ± 3.03 | 5.40 ± 2.81 | 0.0776 ±0.0026 | 1.84 ± 1.49 | 8.92 ± 2.17 | 0.1280 ±0.0034 | 4.07 ± 2.65 | 3.43 ± 1.08 | 0.1096 ±0.0114 | 0.75 ± 0.20 | 4.07 ± 0.63 | 0.2045 ±0.0129 | 3.52 ± 0.81 |
| CG: sparse factorized ($s = 2$) | 76.13 ± 5.91 | 0.0530 ±0.0196 | 61.22 ± 1.04 | 24.82 ± 11.92 | 0.0161 ±0.0007 | 7.42 ± 7.76 | 35.74 ± 8.97 | 0.0183 ±0.0024 | 13.85 ± 10.90 | 19.48 ± 10.72 | 0.0383 ±0.0099 | 4.87 ± 2.69 | 36.04 ± 4.75 | 0.0758 ±0.0052 | 30.99 ± 2.84 |
| CG: dense factorized ($s = 1$) | 55.43 ± 6.43 | 0.1073 ±0.0057 | 75.58 ± 2.85 | 7.67 ± 4.64 | 0.0692 ±0.0026 | 2.86 ± 2.14 | 3.45 ± 0.70 | 0.1266 ±0.0039 | 0.65 ± 0.46 | 2.86 ± 1.90 | 0.1073 ±0.0058 | 0.80 ± 0.33 | 1.69 ± 1.21 | 0.2084 ±0.0197 | 1.79 ± 1.71 |
| CG: dense factorized ($s = 2$) | **82.85** ± 5.68 | 0.0633 ±0.0116 | 56.77 ± 3.14 | 29.61 ± 16.03 | 0.0116 ±0.0045 | 9.83 ± 7.13 | 26.16 ± 11.93 | 0.0194 ±0.0003 | 2.67 ± 1.30 | 17.01 ± 14.70 | 0.0423 ±0.0048 | 1.14 ± 1.15 | 5.70 ± 4.56 | 0.0752 ±0.0110 | 4.64 ± 4.15 |
| CG: joint ($s = 1$) | 1.47 ± 1.22 | 0.5295 ±0.0298 | 4.27 ± 3.54 | 15.35 ± 11.25 | 0.0359 ±0.0074 | 6.74 ± 6.44 | 13.22 ± 8.30 | 0.0745 ±0.0064 | 5.49 ± 4.16 | 14.75 ± 11.00 | 0.0526 ±0.0078 | 4.57 ± 4.76 | 6.18 ± 6.07 | 0.1360 ±0.0138 | 4.82 ± 4.66 |
| CG: dense joint ($s = 2$) | 0.45 ± 0.78 | 0.5508 ±0.0336 | 1.42 ± 2.41 | 18.20 ± 26.55 | 0.0111 ±0.0080 | 9.18 ± 15.52 | 34.49 ± 25.69 | 0.0143 ±0.0037 | 13.77 ± 12.87 | 25.72 ± 20.61 | 0.0241 ±0.0018 | 6.35 ± 6.83 | 14.19 ± 19.63 | 0.0394 ±0.0117 | 9.52 ± 12.63 |
| *Extrapolation-inspired baselines* | | | | | | | | | | | | | | | |
| Additive mean† | 0.00 ± 0.00 | 0.1174 ±0.0093 | 0.00 ± 0.00 | **100.00** ± 0.00 | 0.0088 ±0.0026 | 0.55 ± 0.96 | 0.00 ± 0.00 | 0.1711 ±0.0024 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.0716 ±0.0104 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.3416 ±0.0106 | 0.00 ± 0.00 |
| Additive + residual† | 0.18 ± 0.06 | 0.2322 ±0.0041 | 0.65 ± 0.39 | 7.81 ± 1.53 | 0.1407 ±0.0076 | 6.17 ± 6.23 | 0.00 ± 0.00 | 0.2037 ±0.0008 | 0.00 ± 0.00 | 1.35 ± 0.50 | 0.1523 ±0.0089 | 0.86 ± 0.69 | 0.18 ± 0.20 | 0.3266 ±0.0053 | 0.10 ± 0.10 |
| Fixed composition + residual† | 0.13 ± 0.03 | 0.2353 ±0.0032 | 0.50 ± 0.20 | 0.03 ± 0.06 | 0.2327 ±0.0060 | 0.00 ± 0.00 | 0.02 ± 0.03 | 0.2014 ±0.0141 | 0.00 ± 0.00 | 0.07 ± 0.07 | 0.2548 ±0.0161 | 0.00 ± 0.00 | 0.00 ± 0.00 | 0.3798 ±0.0096 | 0.00 ± 0.00 |
| Rep4Ex-inspired + residual† | 0.85 ± 0.74 | 0.2714 ±0.0658 | 2.87 ± 2.45 | 3.68 ± 3.83 | 0.2490 ±0.1286 | 3.06 ± 3.89 | 0.23 ± 0.35 | 0.2732 ±0.1037 | 0.02 ± 0.03 | 0.39 ± 0.32 | 0.2621 ±0.1050 | 0.02 ± 0.03 | 0.03 ± 0.06 | 0.3648 ±0.1766 | 0.03 ± 0.06 |
| *SEDGE (ours)* | | | | | | | | | | | | | | | |
| SEDGE: no weighting | 12.00 ± 0.15 | 0.2198 ±0.0133 | 29.95 ± 3.03 | 8.98 ± 1.36 | 0.4006 ±0.0066 | 20.82 ± 5.01 | 8.97 ± 1.34 | 0.3650 ±0.0061 | 17.61 ± 3.34 | 8.92 ± 1.34 | 0.4248 ±0.0069 | 17.12 ± 2.88 | 2.46 ± 0.49 | 0.6294 ±0.0069 | 4.10 ± 0.91 |
| SEDGE-LGD ($s = 0.5$) | 38.07 ± 2.28 | 0.1525 ±0.0028 | 70.00 ± 2.51 | 44.11 ± 5.68 | 0.1343 ±0.0121 | 64.81 ± 6.63 | 31.92 ± 2.40 | 0.2048 ±0.0153 | 49.61 ± 5.06 | 47.90 ± 7.27 | 0.1267 ±0.0251 | 60.01 ± 10.11 | 23.67 ± 2.13 | 0.3420 ±0.0225 | 35.40 ± 1.17 |
| SEDGE-LGD ($s = 1$) | 56.80 ± 4.00 | 0.0996 ±0.0107 | 81.08 ± 5.86 | 78.24 ± 2.88 | 0.0104 ±0.0019 | 79.17 ± 1.27 | 60.17 ± 6.17 | 0.0711 ±0.0199 | 64.49 ± 5.27 | 79.57 ± 4.48 | 0.0147 ±0.0063 | 70.05 ± 4.07 | 66.37 ± 2.01 | 0.1153 ±0.0295 | 66.60 ± 1.44 |
| SEDGE-LGD ($s = 2$) | 39.98 ± 19.14 | 0.0272 ±0.0095 | 57.12 ± 18.38 | 77.07 ± 7.78 | 0.0302 ±0.0064 | 62.42 ± 7.84 | 71.89 ± 4.58 | 0.0302 ±0.0024 | 50.91 ± 3.46 | 79.46 ± 4.99 | 0.0328 ±0.0075 | 55.27 ± 9.85 | 76.25 ± 3.09 | 0.0516 ±0.0229 | 56.77 ± 3.07 |
| SEDGE-LWR | 77.55 ± 3.96 | **0.0225** ±0.0183 | **89.97** ± 1.11 | 91.98 ± 1.33 | **0.0018** ±0.0013 | **87.37** ± 3.93 | **91.26** ± 1.27 | **0.0024** ±0.0021 | **84.29** ± 2.46 | **91.55** ± 1.08 | **0.0030** ±0.0007 | **82.23** ± 3.58 | **87.39** ± 3.02 | **0.0016** ±0.0004 | **78.86** ± 1.11 |

Table 6: Compunent reliability of SEDGE.

| Setting | Anchor $Z_1$ satisfaction ↑ | Likelihood TV ↓ | Private decoder $Z_2$ satisfaction ↑ |
|---|---|---|---|
| Weighted Discrete | $94.29 \pm 0.34$ | $0.0142 \pm 0.0017$ | $95.78 \pm 1.44$ |
| Linear | $98.65 \pm 0.37$ | $0.0250 \pm 0.0102$ | $98.94 \pm 0.84$ |
| Cubic | $98.63 \pm 0.39$ | $0.0212 \pm 0.0046$ | $97.67 \pm 1.98$ |
| Smoothstep | $98.65 \pm 0.37$ | $0.0248 \pm 0.0098$ | $97.98 \pm 1.80$ |
| 2D Bridge | $97.09 \pm 0.23$ | $0.0250 \pm 0.0055$ | $98.84 \pm 0.44$ |

the generated $X_2$; joint compatibility requires the same $X_2$ to permit both specifications. This isolates bridge feasibility from errors in generating the private variables. Out-of-support or nonfinite bridges count as incompatible. Table 7 retains the settings, methods, and selected scales of Table 1 without reselection. SEDGE-LWR consistently improves joint compatibility over CD across the five settings, supporting its ability to locate shared features compatible with novel specification combinations. Compatibility alone, however, does not establish distributional fidelity or coverage, as illustrated by the additive-mean control in the Linear setting.

Table 7: Bridge compatibility rates (percent).

| Method | Weighted Discrete | | | Linear | | | Cubic | | | Smoothstep | | | 2D Bridge | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $Z_1$ ↑ | $Z_2$ ↑ | Both↑ | $Z_1$ ↑ | $Z_2$ ↑ | Both↑ | $Z_1$ ↑ | $Z_2$ ↑ | Both↑ | $Z_1$ ↑ | $Z_2$ ↑ | Both↑ | $Z_1$ ↑ | $Z_2$ ↑ | Both↑ |
| *Conditional diffusion* | | | | | | | | | | | | | | | |
| CD | 23.25 | 95.35 | 18.60 | 91.99 | 91.29 | 83.28 | 93.38 | 83.94 | 77.31 | 91.55 | 94.08 | 85.63 | 88.93 | 88.12 | 82.36 |
| Compositional CD | 86.40 | 95.60 | 82.00 | 71.21 | 71.24 | 42.45 | 63.40 | 69.71 | 33.11 | 56.02 | 80.05 | 36.07 | 46.42 | 52.64 | 34.65 |
| *Classifier-guided diffusion* | | | | | | | | | | | | | | | |
| CG: sparse factorized | 90.20 | 92.43 | 82.63 | 74.87 | 71.48 | 46.35 | 77.29 | 66.57 | 43.88 | 64.47 | 87.63 | 52.10 | 60.07 | 70.80 | 47.28 |
| CG: dense factorized | 97.08 | 90.05 | **87.13** | 75.60 | 72.15 | 47.75 | 75.47 | 66.42 | 41.89 | 62.03 | 86.83 | 48.86 | 67.50 | 62.78 | 47.33 |
| CG: joint | 86.07 | 31.63 | 17.78 | 76.38 | 83.14 | 59.52 | 70.08 | 81.53 | 51.63 | 71.32 | 91.99 | 63.31 | 71.21 | 79.28 | 59.93 |
| *Extrapolation-inspired baselines* | | | | | | | | | | | | | | | |
| Additive mean† | 0.00 | **100.00** | 0.00 | **100.00** | **100.00** | **100.00** | **100.00** | 0.00 | 0.00 | 0.00 | **100.00** | 0.00 | 0.00 | 0.00 | 0.00 |
| Additive + residual† | 49.15 | 52.40 | 10.33 | 55.05 | 54.07 | 9.23 | 64.16 | 43.77 | 9.36 | 49.17 | 60.92 | 10.35 | 21.57 | 22.98 | 2.08 |
| Fixed composition + residual† | 49.88 | 52.33 | 9.87 | 36.70 | 70.46 | 9.41 | 54.10 | 55.94 | 10.29 | 32.80 | 74.38 | 9.54 | 17.48 | 26.45 | 1.99 |
| Rep4Ex-inspired + residual† | 39.53 | 66.00 | 9.30 | 53.99 | 40.79 | 7.18 | 62.08 | 32.57 | 5.79 | 51.42 | 50.28 | 11.98 | 24.53 | 28.61 | 3.37 |
| *SEDGE (ours)* | | | | | | | | | | | | | | | |
| SEDGE: no weighting | **97.35** | 14.50 | 13.13 | 98.73 | 9.70 | 9.02 | 98.70 | 9.73 | 9.05 | **98.73** | 9.65 | 8.97 | **97.41** | 2.62 | 2.46 |
| SEDGE-LGD | 78.95 | 81.93 | 60.88 | 89.27 | 88.36 | 77.65 | 73.73 | 98.16 | 71.89 | 88.66 | 90.53 | 79.18 | 84.68 | 84.96 | 77.85 |
| SEDGE-LWR | 89.35 | 93.82 | 83.17 | 92.58 | **100.00** | 92.58 | 93.00 | **100.00** | **93.00** | 92.15 | **100.00** | **92.15** | 89.47 | **94.03** | **89.21** |

### F.1.6 Results & Sensitivity Study: Structure Estimation and Sparsity Penalties

We evaluate structure recovery on 30 continuous settings spanning linear, cubic, and smoothstep mechanisms, with three random seeds per setting. Structural Hamming distance (SHD) counts missing and extra feature-to-specification edges; lower is better.

We compare PC-stable Colombo & Maathuis (2014) with KCI, RCIT, RCoT, and Fisher-Z conditional-independence tests, using the known bipartite direction and significance level $\alpha = 0.05$. For likelihood-based estimation, we fit a conditional spline flow for each specification under each of the eight possible input masks, then select the mask minimizing validation negative log-likelihood plus $\lambda$ times its number of parents. The primary penalty is $\lambda = 0.01$; we additionally sweep $\lambda \in \{0, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1\}$.

Fig. 5 shows that KCI and Fisher-Z recover every tested graph, while RCIT and RCoT make occasional errors. Stronger sparsity penalties substantially improve likelihood-based selection: mean SHD decreases from $0.967$ without a penalty to $0.922$, $0.111$, and $0$ at $\lambda = 0.01$, $0.3$, and $1$, respectively. Thus, sparsity helps remove unnecessary parents without losing true edges at the strongest tested penalty.

### F.1.7 Results: Feature or Specification not Given

We evaluate feature and specification recovery under linear mixtures across the Linear, Cubic, and Smoothstep data setting families. Fig. 6 reports mean correlation coefficients (MCCs) after coordinate matching, providing an empirical assessment of the recoverability discussed in our theoretical analysis. Recovery is generally stronger when only specifications are unknown than when both features and

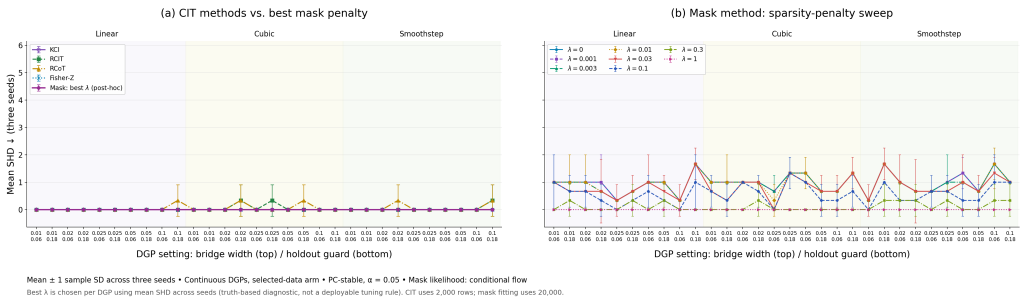



Figure 5: Structure recovery on continuous benchmarks after held-out-region selection. Left: CI-test-based PC versus the best mask penalty per setting. Right: sensitivity to the sparsity penalty. Points and error bars show means and one sample standard deviation across three seeds. Background colors distinguish mechanism families; horizontal-axis labels specify bridge width and holdout guard.

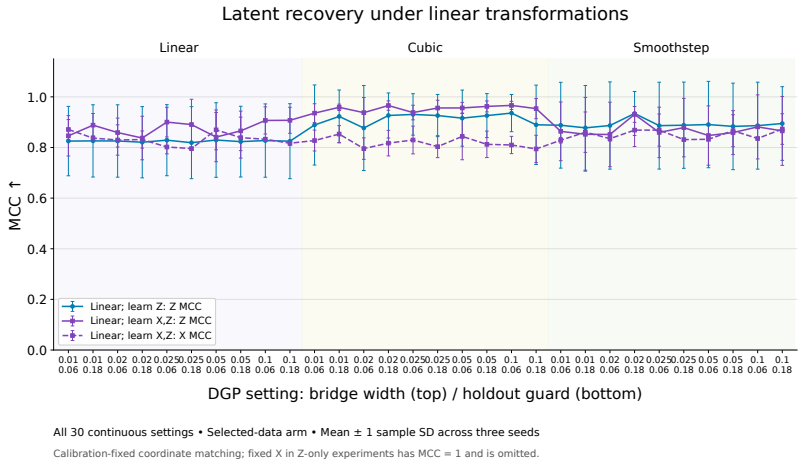



Figure 6: Feature and specification recovery under linear observation mixtures across 30 continuous settings. Curves show mean MCCs with one-standard-deviation error bars across three seeds. Coordinate matching is fixed using a separate calibration set. When only $Z$ is learned, $X$ is observed and its MCC of one is omitted.

specifications must be learned. Once representations and their structure are estimated, the same SEDGE extrapolation procedure applies as in the observed-variable setting. We focus here on representation recovery and omit downstream generation results.

## F.2 Extrapolative Property-to-Molecule Design

### F.2.1 Experiment setup

**Dataset.** We construct v0-2 from 100,000 unique molecules, using continuous RDKit LogP and TPSA values as conditioning variables. We hold out the region $\mathrm{LogP} \geq 5$ and $\mathrm{TPSA} \geq 100\,\text{Å}^2$, containing 1,042 molecules (1.042%), and divide it into validation and test sets of 516 and 520 distinct target pairs, respectively. The remaining molecules form the training (89,005) and in-distribution validation (9,953) sets. Splitting is performed by exact (LogP, TPSA) pairs to prevent identical conditions from appearing across splits. All targets lie within the training range of each individual property, while their joint region is entirely excluded from training.

**Evaluation.** Target match requires a valid molecule satisfying both $|\Delta\mathrm{LogP}| \leq 0.2$ and $|\Delta\mathrm{TPSA}| \leq 5\,\text{Å}^2$; region hit instead requires its measured properties to lie in the held-out region. Invalid, missing, and repeated outputs remain in the attempt denominator. Molecular validity and properties are evaluated using the common RDKit-based evaluation rules.

**Baselines.** **STGG+** predicts molecular construction actions conditioned directly on the numerical property pair. Its training objective combines next-action cross-entropy with the auxiliary squared-error property loss. The model learns unconditional, single-property, and joint-property conditioning. At inference, joint-condition classifier-free guidance (CFG) with scale 1.5 and temperature 1 are used. We retain STGG+ on our dataset from scratch. **STGG+ (logit comp.)** tests whether simply combining single-property guidance at inference time is sufficient for joint-property extrapolation. It reuses the trained STGG+. At step $t$, let $\ell_{00,t}$, $\ell_{10,t}$, $\ell_{01,t}$, and $\ell_{11,t}$ denote action logits computed on the same molecular prefix with neither property, LogP only, TPSA only, and both properties observed, respectively. A separate missingness indicator distinguishes an unobserved property from an observed zero-valued target. During generation, it uses $\ell_{\mathrm{comp},t} = \ell_{00,t} + 1.5(\ell_{10,t} - \ell_{00,t}) + 1.5(\ell_{01,t} - \ell_{00,t})$ instead of original $\ell_{\mathrm{joint}} = \ell_{00} + 1.5(\ell_{11} - \ell_{00})$. **LDMol** and **BioT5+** receive the fixed template *The molecule has a logP of {LogP} and a topological polar surface area (TPSA) of {TPSA} square angstroms.* Starting from released checkpoints, LDMol updates only its diffusion Transformer for 10,000 steps (batch 128, learning rate $10^{-5}$), keeping the autoencoder and text encoder frozen. Sampling uses 100 diffusion steps and CFG 2.5. BioT5+ updates all parameters for three epochs (batch 64, learning rate $3 \times 10^{-5}$), then samples SELFIES with top-$p = 0.95$ and temperature 1. We admit that they might have seen molecules in the held-out region during their pre-training.

**Ours and ablation model.** **SEDGE** furhter splits the hidden state of STGG+ into $C$ (512 dimensions) and $H_1, H_2$ (256 each). Two local modules update $H_i$ using the conditioned $C$ and property $z_i$; their outputs are concatenated with $C$ and processed by the final Transformer layer. We model each property likelihood $p_{\theta_i}(z_i \mid C^0, H_i)$ with a Gaussian head. Training combines action cross-entropy with mean Gaussian NLL weighted by 0.1. All targets are observed, and generation uses no CFG. **SEDGE (Blocks Only)** retains the partition and local modules but exposes both local modules to all observed properties. Native STGG+ property MSE replaces the property likelihood objective. Random property masking and CFG 1.5 in STGG+ are restored.

### F.2.2 More experiment results

**Quantitative results and ablations.** Table 8 reports performance on both In-distribution (ID) and Held-out region property targets. SEDGE achieves the highest ID joint-success rate (JSR), 58.05%, together with the lowest LogP and TPSA MAEs. On Held-out region targets, SEDGE achieves a JSR of 33.82%, exceeding STGG+ by 11.28 percentage points, with a reported validity of 100.00%. Fine-tuning improves both pretrained text-to-molecule baselines, but their Held-out region JSRs remain below 1%. Naive logit composition reduces the Held-out region JSR of STGG+ from 22.54% to 18.51%, while the blocks-only ablation reaches 25.11%, compared with 33.82% for full SEDGE. These comparisons suggest that the gain from the full model extends beyond introducing latent blocks or combining property-conditioned logits.

**Complete sample pools for Figure 3.** Figure 7 shows generated samples per method for the same target used in the main-text example: $\mathrm{LogP} = 7.0456$ and $\mathrm{TPSA} = 110.94\,\text{Å}^2$. SEDGE produces 15 joint target hits, compared with 3 for STGG+, under tolerances of $\pm 0.2$ for LogP and $\pm 5\,\text{Å}^2$ for TPSA. For this target, the entire tolerance window lies inside the held-out region, so each joint target hit also counts as an Held-out region success. The complete pools include both successful and unsuccessful generations.

**Details of Figure 3.** For each target condition we draw 30 samples per method and evaluate the measured logP and TPSA of each sample; a sample counts as a joint hit when both properties fall within the tolerance ($\pm 0.2$ logP, $\pm 5$ TPSA) of the target. For the condition (target logP $= 7.0456$, TPSA $= 110.94$), SEDGE achieves 15/30 joint hits versus 3/30 for STGG+ (two-sided Fisher exact $p = 0.0015$). Across all 520 conditions, Wilcoxon signed-rank $p = 9.4 \times 10^{-43}$). All training and fine-tuning exclude the held-out joint region shown in the figure. The two example molecules are a representative failure of STGG+ and a joint hit of SEDGE under this condition.

Table 8: Additional results for property-guided molecular generation on In-ditribution and Held-out region targets. JSR measures the percentage of generation attempts yielding a valid molecule within both target tolerances: $\pm 0.2$ for LogP and $\pm 5$ [2] for TPSA. Held-out region JSR additionally requires measured $\mathrm{LogP} \geq 5$ and $\mathrm{TPSA} \geq 100$ [2]. U-N yield counts unique successful molecules absent from the training set, divided by all generation attempts. Validity measures the percentage passing molecular validity checks. MAEs are computed over outputs with valid measured properties; TPSA MAE is in [2]. Invalid and missing outputs remain in the denominators of generation rates.

| Method | In-distribution region | | | Held-out region | | |
|---|---|---|---|---|---|---|
| | JSR (%) ↑ | LogP MAE ↓ | TPSA MAE ↓ | JSR (%) ↑ | U-N yield (%) ↑ | Validity (%) ↑ |
| *Direct property-conditioned baselines* | | | | | | |
| STGG+ | 47.81 | 0.248 | 2.95 | 22.54 | 22.54 | 99.96 |
| STGG+ (logit comp.) | 31.48 | 0.414 | 5.56 | 18.51 | 18.51 | 99.99 |
| *Pretrained text-to-molecule baselines* | | | | | | |
| LDMol (zero-shot) | 0.00 | 6.770 | 75.79 | 0.00 | 0.00 | 45.60 |
| LDMol (fine-tuned) | 1.17 | 1.204 | 26.25 | 0.35 | 0.35 | 62.86 |
| BioT5+ (zero-shot) | 0.03 | 3.797 | 172.32 | 0.03 | 0.03 | 99.01 |
| BioT5+ (fine-tuned) | 4.74 | 0.764 | 13.21 | 0.95 | 0.95 | 99.96 |
| *Proposed method and ablations* | | | | | | |
| SEDGE (blocks only) | 45.63 | 0.247 | 3.19 | 25.11 | 25.11 | 99.97 |
| SEDGE | **58.05** | **0.228** | **2.48** | **33.82** | **33.82** | **100.00** |

### F.3 Extrapolative Text-to-Image Generation

In this section, we describe the experimental setup for applying SEDGE to extrapolative text-to-image generation, including implementation details and datasets. We then present qualitative and quantitative comparisons with SANA and Aligner.

#### F.3.1 Experiment Setup

We implement SEDGE in the concept space learned by Aligner (Xie et al., 2025b). Specifically, we use image concepts $\mathbf{z}^I$ as feature variables and text concepts $\mathbf{z}^T$ as specification variables. These learned representations are treated as given variables, following the feature-and-specification setting considered in Scenario 1 of the synthetic experiments. We denote the numbers of image and text concepts by $N_I$ and $N_T$, respectively.

The dependency structure required by our likelihood model is learned from data. While Aligner models visual concepts conditioned on textual concepts, our likelihood model estimates the conditional distribution of textual concepts given visual concepts. We therefore jointly learn a concept-level likelihood predictor and a sparse mask that parameterizes its cross-modal dependencies.

To generate image concepts satisfying the textual specifications, we use Algorithm 4 in the image-concept space. Our implementation consists of a likelihood predictor, a learnable mask, and a diffusion prior. The sampled image concepts subsequently condition a SANA-based image generator.

**Likelihood Predictor.** We use concept-wise conditional normalizing flows to model the conditional distributions of text concepts given image concepts. For each text concept $z_i^T$, a learnable soft mask $\mathbf{M}_i \in [0,1]^{N_I}$ weights the image concepts used as conditioning information. An MLP encodes the masked representation into a context vector: $c_i = \mathrm{MLP}_i\left(\mathbf{z}^I \odot \mathbf{M}_i\right)$, where each mask weight is applied to the corresponding image-concept vector. Conditioned on $c_i$, an invertible transformation $f_i$ maps $z_i^T$ to a standard Gaussian base distribution. The conditional log-likelihood is

$$\log p_{\theta_i}(z_i^T \mid \mathbf{z}^I) = \log p_u\big(f_i(z_i^T; c_i)\big) + \log \left| \det \frac{\partial f_i(z_i^T; c_i)}{\partial z_i^T} \right|,$$

where $p_u = \mathcal{N}(0, \mathbf{I})$. The likelihood loss is the sum of the negative log-likelihoods across all $N_T$ text concepts, averaged over the training samples.

**Causal Mask.** We parameterize the cross-modal dependency structure using a learnable mask $\mathbf{M} \in [0,1]^{N_T \times N_I}$, with entries $M_{ij} = \sigma(w_{ij})$, where $w_{ij}$ is a learnable parameter and $\sigma$ is the

sigmoid function. Each entry controls the contribution of image concept $z_j^I$ to the prediction of text concept $z_i^T$. Alongside the likelihood loss, we apply the sparsity regularizer $\mathcal{L}_s = \sum_{i=1}^{N_T} \sum_{j=1}^{N_I} |M_{ij}|$ to encourage sparse connections between image and text concepts.

**Prior Model.** We train an unconditional Diffusion Transformer (DiT) to model the image-concept prior $p^{\mathcal{D}}(\mathbf{z}^I)$. The image-concept representation is organized as $\mathbf{z}^I \in \mathbb{R}^{N_I \times D_I}$, where $D_I$ is the dimensionality of each image concept. The DiT processes this representation as a sequence of $N_I$ concept tokens and provides the diffusion prior used during DPS.

**Image Generation Backbone.** Following Aligner (Xie et al., 2025b), we use a SANA-based generator (Xie et al., 2024) conditioned on image concepts to produce the final images. At inference time, a text prompt is mapped to text concepts, which serve as the specifications for SEDGE. DPS then samples image concepts under these specifications, and the sampled concepts are passed to the image generator.

**Training Data.** We use image–text pairs from FLUX-Reason-6M (Fang et al., 2025) for training. The dataset contains approximately six million images synthesized by FLUX.1-dev, accompanied by textual descriptions covering diverse semantic and compositional characteristics, including object identities, spatial relations, and visual styles.

### F.3.2 Qualitative Evaluation Results

Figure 8 presents additional qualitative comparisons of SEDGE with SANA and Aligner on prompts involving object interactions and unusual attribute combinations. The crocodile example highlights the importance of preserving the target object's identity: SEDGE retains a crocodile snout while adding a lion's mane, whereas the baseline outputs exhibit more lion-like facial features. For the shopping-cart prompt, SANA produces a cart with conventional wheels, and Aligner introduces an unrequested tank turret. SEDGE incorporates tank treads into the cart's base while preserving its recognizable structure. These examples illustrate selective composition of the requested attributes.

Figure 9 examines generation consistency and sensitivity to the guidance parameter $\eta$ using the prompt "a camera with insect antennae." We compare Aligner with SEDGE across eight random seeds and six guidance values, $\eta \in \{0.1, 0.3, 0.5, 0.7, 1.0, 1.5\}$. Here, consistency concerns whether the generated images satisfy the requested camera–antennae combination across different seeds and guidance settings. Aligner frequently introduces additional insect anatomy, including legs and wings. For many of the displayed seeds, larger guidance values reduce these unwanted features while retaining a recognizable camera with antennae.

### F.3.3 Quantitative Evaluation Results

We constructed a new 75-prompt benchmark for extrapolative T2I for quantitative evaluation. For each generated image, we obtained three GPT-5.6-Luna ratings on a 0–10 scale, obtained through separate evaluation calls For the CLIP score (Hessel et al., 2021), we used the OpenAI ViT-L-14 model (Radford et al., 2021) as the scorer. For each prompt, we used three different seeds to generate three images and reported the mean score. For GenEval (Ghosh et al., 2023), we used the official GenEval scoring pipeline and its canonical metadata for all 553 prompts. We computed accuracy separately for each of the six GenEval categories, and the final score was their unweighted mean.

Table 9: Quantitative comparison on our 75-prompt benchmark and GenEval. The last three columns report GenEval results. Overall is the unweighted mean accuracy across all six GenEval categories. Best results are shown in bold.

| Model | GPT-5.6-Luna ↑ | CLIP score ↑ | Position ↑ | Color Attribution ↑ | Overall ↑ |
|---|---|---|---|---|---|
| SANA | 7.696 | 29.481 | 25.75% | 42.75% | 63.20% |
| Aligner | 7.691 | 29.611 | 24.50% | 48.75% | 66.32% |
| SEDGE (Ours) | **8.058** | **29.724** | **41.00%** | **49.00%** | **66.49%** |

We compared SEDGE with SANA and Aligner, where all three methods are evaluated under the same 1.6B-scale backbone setting, providing a controlled comparison that avoids conflating structural

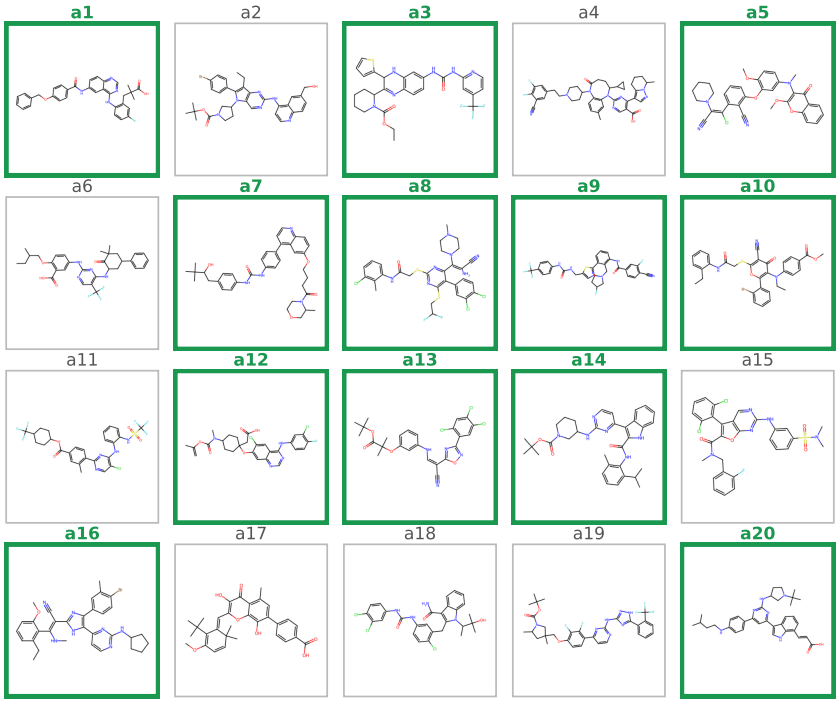
improvements with model scale. Within this matched setting, SEDGE achieves the highest reported overall scores on GPT-5.6-Luna, CLIP, and GenEval, showing that its extrapolation-oriented design does not compromise standard image–text alignment or compositional generation quality. Notably, the Position and Color Attribute categories in GenEval test spatial and attribute recombination using prompts involving relatively uncommon combinations, such as "a pink banana" and "a wine glass above a kite," and may therefore better reflect a model's extrapolation ability. SEDGE performs best in both categories, with a substantial gain in Position accuracy (41.00% vs. 25.75%).

These evaluations serve complementary purposes: GPT-5.6-Luna and CLIP measure image–text alignment, whereas GenEval measures general compositional correctness on its standard benchmark without establishing novelty relative to our training distribution. Thus, GenEval complements rather than replaces our original extrapolation-focused evaluation.

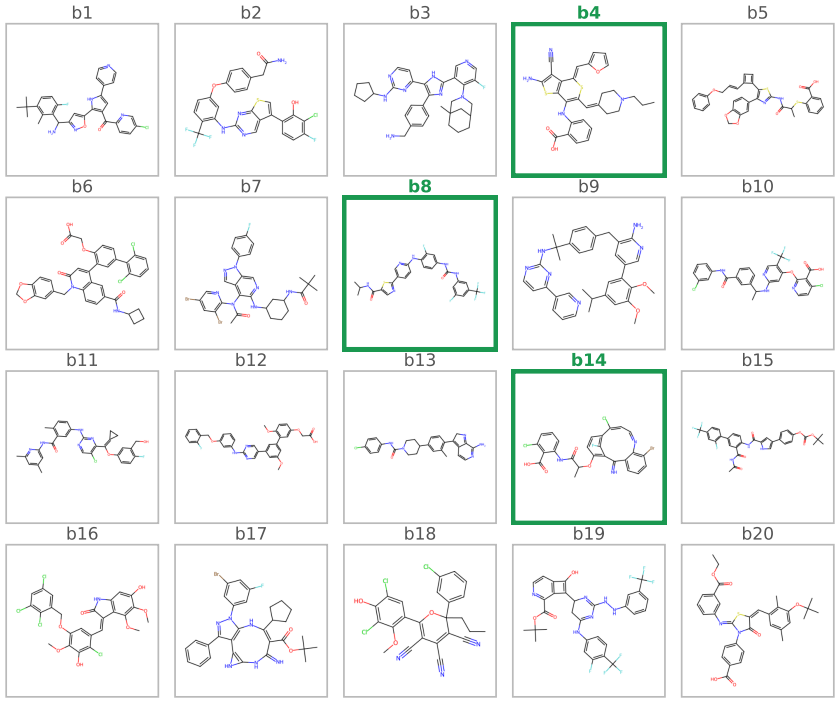
Table 10: Mean differences in scores between SEDGE and the baselines, with 95% confidence intervals obtained by paired bootstrap resampling over prompts. Positive values favor SEDGE.

| **Comparison** | **Mean difference ↑** | **95% CI** |
|---|---|---|
| SEDGE − SANA | $+0.362$ | $[0.020, 0.787]$ |
| SEDGE − Aligner | $+0.367$ | $[0.082, 0.711]$ |

We also examined uncertainty in the GPT-based evaluation. We compared SEDGE with Aligner and SANA using paired prompt-level differences. We perform BCa bootstrap confidence interval estimation over paired prompt-level differences (1,000,000 bootstrap resamples). Positive confidence intervals indicate that the improvement is statistically distinguishable from zero under the paired bootstrap analysis. As shown in the Table 10, the paired confidence intervals remain above zero, indicating that the observed improvement is not driven by evaluator stochasticity or a small subset of evaluation prompts.




**(a) Ours**




**(b) STGG+**

Figure 7: Generated molecules for the held-out joint-property target (logP $= 7.05$, TPSA $= 110.9$ Å$^2$) shown in Figure 3. (a) Twenty samples from our method (SEDGE); (b) twenty samples from the baseline (STGG+). Green boxes mark molecules that satisfy both the Target and Region constraints.




Figure 8: Qualitative comparisons on text-to-image generation. Each column corresponds to the prompt shown above, and rows show SANA, Aligner, and SEDGE from top to bottom.

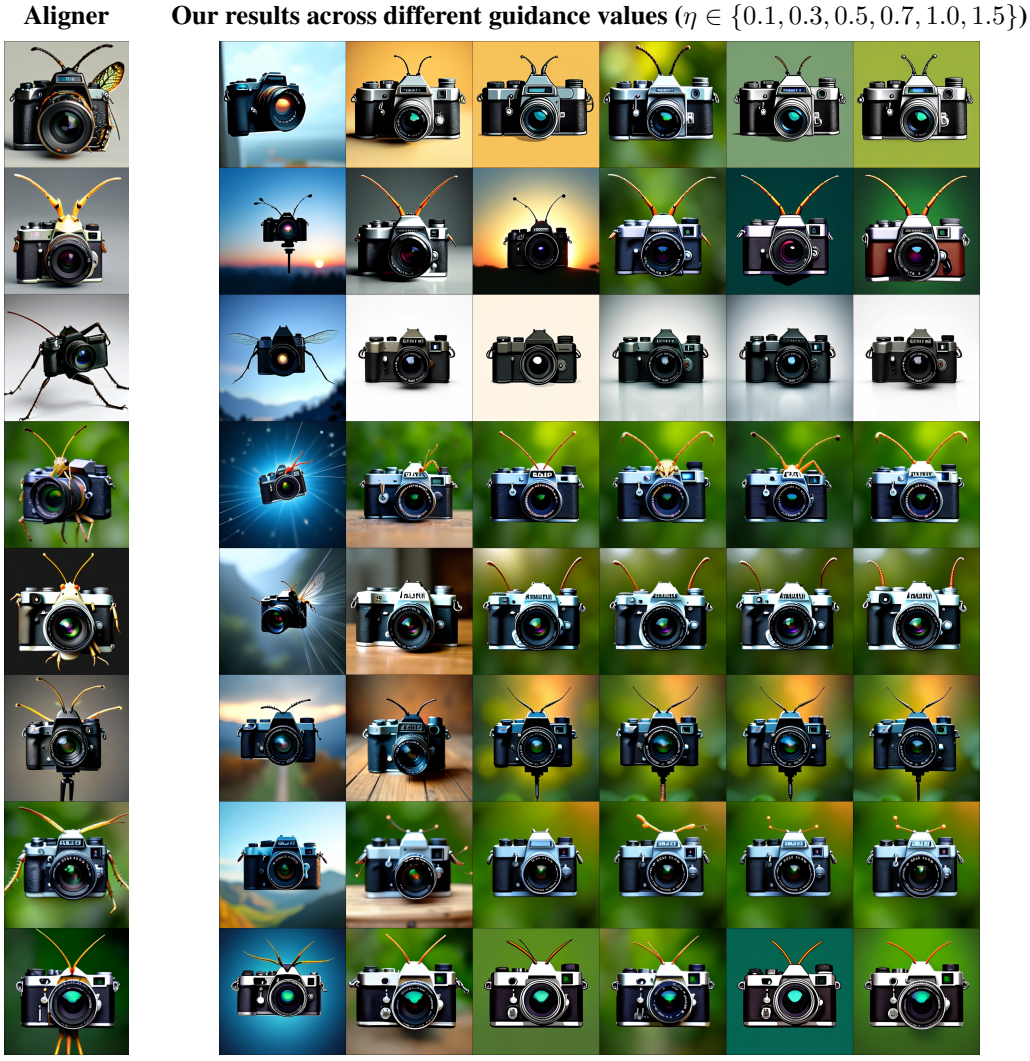



Figure 9: Consistency of our generation. Given prompt `a camera with insect atennae`, we generated figures using both ours and concept aligner models acorss 8 different seeds. As we can see in the figure, the generation of concept aligner has mixed up concepts like legs and bodies. Our generation results (especially with larger guidance values) makes consistent generation.