DIFFUSION MODELS WITH DOUBLE GUIDANCE

Yanfeng Yang
Graduate University of Advanced Studies & The Institute of Statistical Mathematics
Tokyo, Japan
yanfengyang0316@gmail.com

Kenji Fukumizu
The Institute of Statistical Mathematics
Tokyo, Japan
fukumizu@ism.ac.jp

ABSTRACT

Creating large-scale datasets for training high-performance generative models is often prohibitively expensive, especially when associated attributes or annotations must be provided. As a result, merging existing datasets has become a common strategy. However, the sets of attributes across datasets are often inconsistent, and their naive concatenation typically leads to block-wise missing conditions. This presents a significant challenge for conditional generative modeling when the multiple attributes are used jointly as conditions, thereby limiting the model's controllability and applicability. To address this issue, we propose a novel generative approach, Diffusion Model with Double Guidance, which enables precise conditional generation even when no training samples contain all conditions simultaneously. Our method maintains rigorous control over multiple conditions without requiring joint annotations. We demonstrate its effectiveness in simulation and a molecular task, where it outperforms existing baselines both in alignment with target conditional distributions and in controllability under missing condition settings.

1 INTRODUCTION

Diffusion-based generative models have achieved remarkable success across a range of data modalities, including images, videos, molecules, and text (Ho et al., 2020; Song et al., 2021; Rombach et al., 2021; Ho et al., 2022; Hoogeboom et al., 2022; Xu et al., 2025). In many real-world applications, the goal is not simple generation from the unconditional distribution P_{X_0} of data $X_0 \in \mathbb{R}^d$, but rather from the conditional distribution $P_{X_0|C}$ given some auxiliary conditions $C \in \mathbb{R}^p$ such as labels and properties. Conditional diffusion models (CDMs) enable such generation under conditioning information, allowing fine-grained control over content. In computer vision, CDMs generate images conditioned on object categories, spatial layouts, or styles (Song et al., 2023; Zhang et al., 2023; Zhao et al., 2025b; Chung et al., 2023; Ho & Salimans, 2022; Zhao et al., 2025a; Dhariwal & Nichol, 2021). In medical imaging, CDMs have been used to rectify tilted CT images (Kawata et al., 2025). In chemistry, CDMs enable the generation of molecules with the desired physicochemical properties, facilitating drug discovery and material design (Gebauer et al., 2022; Shen et al., 2025; Ninniri et al., 2024). In time series, CDMs provide effective approaches to probabilistic forecasting (Ye et al., 2025; Yang et al., 2025a). Beyond these domains, CDMs have also been applied to Bayesian inference and causal discovery (Gloeckler et al., 2025; Yang et al., 2025b; Sanchez & Tsaftaris, 2022).

Despite the strong expressive power of CDMs, training a high-quality CDM critically depends on the sample size of the training data (Chen et al., 2023; Fu et al., 2024). A common strategy to increase the amount of training data is to merge existing datasets into an aggregated one (Kaufman et al., 2024a; Saharia et al., 2022; Li et al., 2020; Fu et al., 2024). However, datasets collected from different sources often contain distinct types of conditioning information. In the simplified setting illustrated in Table 1, dataset $D^{(1)}$ contains no samples labeled with condition C_2, while dataset $D^{(2)}$

lacks samples associated with condition C_1. This naturally gives rise to a fundamental challenge: how can a CDM capture the conditional distribution with joint conditions $P_{X_0|C_1,C_2}$ with aggregated datasets containing block-wise missing conditions? Addressing this challenge would substantially enhance the controllability of diffusion models and broaden the range of tasks they can support.

As a motivating example, molecular generation tasks in chemistry exhibit particularly pronounced challenges arising from data aggregation. In drug discovery, conditional generative models have been widely used to design molecular candidates that exhibit the desired physicochemical properties (Li et al., 2024; Shen et al., 2025; Hoogeboom et al., 2022; Gebauer et al., 2022). However, constructing large-scale molecular datasets annotated with additional properties remains a formidable challenge in terms of both cost and time. Laboratory measurements are often prohibitively expensive (Erhirhie et al., 2018; Shen et al., 2025), while the accurate computation of molecular properties is notoriously time-consuming (Fernández et al., 2024; Jarrold, 2022).

Conventional approaches to accessing the combined conditions have been considered. A direct approach is to recover missing conditional information via manual experiments or imputation approaches. As discussed above, the former is often impractical; the latter, imputation, typically relies on strong correlations between variables (Yoon et al., 2018; Jolicoeur-Martineau et al., 2024). However, in our case, the datasets may lack samples drawn from the joint distribution P_{C_1,C_2}, which severely compromises the performance of standard imputation methods, leading to inaccurate results.

Recent generative approaches to approximating $P_{X_0|C_1,C_2}$ from block-wise missing datasets include composing different score functions (Liu et al., 2022; Gaudi et al., 2025). In parallel, other strategies leverage ControlNets (Zhang et al., 2023) to encode additional conditions into a pre-trained conditional generative model, with further fine-tuning via reinforcement learning (RL) (Zhao et al., 2025b). Although compositional methods offer a flexible way to combine multiple conditions, their theoretical foundation may not have theoretical rigorousness (Bradley et al., 2025). On the other hand, the RL-based method relies on a strong pre-trained diffusion model (Zhao et al., 2025b), and the reward function neglects finer-grained control over the intermediate steps of the reverse process (Hu et al., 2025).

In light of these limitations, we argue that an approach should be both explainable and training-free. Moreover, it should leverage the dependence structure among variables, without relying on additional experiments or imputations.

In this work, we assume access to a pre-trained diffusion model that has already captured the distribution P_{X_0} and $P_{X_0|C_1}$. Building on this foundation, we develop methods to generate samples from the joint conditional distribution $P_{X_0|C_1,C_2}$, even in the absence of joint samples from P_{C_1,C_2} being available. Our approach leverages the assumption that C_1 and C_2 are conditionally independent given X_0 and exploits this structure to introduce an additional guidance term into the generation process.

Our main contributions are summarized as follows.

- **Plug-and-play conditional generation on block-wise missing datasets:** We propose two efficient methods for generating samples approximately from $P_{X_0|C_1,C_2}$ when we only have access to block-wise missing datasets shown in Table 1.

- **Extensive experiments across multiple data types:** We evaluate our methods on synthetic datasets and molecular datasets (Axelrod & Gómez-Bombarelli, 2022; Sterling & Irwin, 2015), demonstrating superior recovery of conditional distributions and controllability over the target conditions.

The remainder of this paper is organized as follows. Section 2 introduces the problem setting and background. Section 3 describes our proposed methods. Section 4 presents a thorough simulation study. Section 5 includes a practical applications to molecular datasets. Section 6 summarizes our results and provides a discussion of the future directions.

Table 1: An illustration of aggregated datasets. Concatenating them makes a block-wise missing dataset. Missing values are represented as $\varnothing$.

Dataset	Target variable	Condition 1	Condition 2
$D^{(1)}$	$X_0^{(1)}$	$C_1^{(1)}$	$\varnothing$
$D^{(2)}$	$X_0^{(2)}$	$\varnothing$	$C_2^{(2)}$

2 Problem Setting and Preliminaries

2.1 Problem setting

Our algorithms are designed to operate on function-mapping datasets, which consist of samples (X_0, C). The condition C can be decomposed into two distinct parts (C_1, C_2) by dimension, where $C_1 \in \mathbb{R}^k$ and $C_2 \in \mathbb{R}^{p-k}$. Both C_1 and C_2 can be expressed by two deterministic functions $f_1 : \mathbb{R}^d \to \mathbb{R}^k$ and $f_2 : \mathbb{R}^d \to \mathbb{R}^{p-k}$ with additive noise:

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} f_1(X_0) \\ f_2(X_0) \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix}, \epsilon_1 \sim N(0, \sigma_1^2 I_k), \epsilon_2 \sim N(0, \sigma_2^2 I_{p-k}), \epsilon_1 \perp\!\!\!\perp \epsilon_2, \sigma_1, \sigma_2 > 0. \quad (1)$$

An important consequence of this assumption is the **conditional independence** (CI), denoted by $C_1 \perp\!\!\!\perp C_2 | X_0$. Note that this is different from (unconditional) independence $C_1 \perp\!\!\!\perp C_2$; even if they are correlated, the conditional independence given X_0 may hold.

Function-mapping datasets are ubiquitous and naturally arise from a variety of domains of science. Molecular datasets, for example, are typical function-mapping datasets (Axelrod & Gómez-Bombarelli, 2022; Sterling & Irwin, 2015; Ramakrishnan et al., 2014), where X_0 represents the structure of a molecule and C_1, C_2 correspond to different properties of the molecule. Note that in this case, two properties C_1 and C_2 can be correlated; while given a molecule X_0 they can be deterministic and therefore conditionally independent.

Although different datasets consistently include samples of X_0, the available conditions can vary due to technical constraints or limitations of the data acquisition devices (Table 1). For ease of exposition, we mainly discuss the case of two datasets with partially observed conditions: $D^{(1)} = \{X_0^{(1)}, C_1^{(1)}, \varnothing\}$ and $D^{(2)} = \{X_0^{(2)}, \varnothing, C_2^{(2)}\}$. More general cases involving more than two datasets and missing conditions can be found in Appendix B. Additionally, in practical situations, the empirical distributions of $X_0^{(1)}$ and $X_0^{(2)}$ may not be exactly the same but only be similar.

2.2 EDM framework

Diffusion models consist of a forward and a reverse process. In the forward process, Gaussian noises are gradually added to the original data, and the score function of the noisy data is learned. The reverse process then employs stochastic differential equations (SDE) or ordinary differential equations (ODE) with the learned score function to convert a Gaussian noise into a sample that closely approximates the original data distribution (Ho et al., 2020; Song et al., 2021). We employ the framework of Elucidated Diffusion Model (EDM, Karras et al., 2022), which represents a streamlined diffusion model using the simple and intuitive forward process:

$$dX_t = \sqrt{2t}\, dB_t, \quad t \in [0, t_{\max}], \quad (2)$$

where B_t is a Wiener process. By the property of Ornstein-Uhlenbeck process, we have $X_t \overset{d}{=} X_0 + t\epsilon, \epsilon \sim N(0, I_d)$ (Yang et al., 2025b). Analytically, the ODEs used for unconditional and conditional sampling in EDM's reverse process are given by, respectively,:

$$dX_t = -t \cdot \nabla \log p_t(X_t) dt, \quad \text{and} \quad dX_t = -t \cdot \nabla \log p_t(X_t | C_1) dt, \quad t \in [t_{\min}, t_{\max}], \quad (3)$$

where $p_t(X_t)$ is the density of X_t and $p_t(X_t|C_1)$ is the conditional density of X_t given C_1. In this paper, unless otherwise specified, ∇ denotes the gradient with respect to X_t. To avoid numerical explosions, the reverse process uses an early-stopping at $t_{\min}$ close to but larger than zero. Since the score functions $\nabla \log p_t(X_t)$ and $\nabla \log p_t(X_t|C_1)$ lack explicit analytical expressions, we train

a neural network nn_θ, parameterized by θ, to approximate them simultaneously by minimizing the following loss:

$$\mathcal{L} = \begin{cases} \mathbb{E}_{X_0,t,\epsilon} \| X_0 - nn_\theta\left(X_0 + t\epsilon, t, \varnothing\right) \|_2^2, & p_{\text{non}} \\ \mathbb{E}_{X_0,C_1,t,\epsilon} \| X_0 - nn_\theta\left(X_0 + t\epsilon, t, C_1\right) \|_2^2, & 1 - p_{\text{non}} \end{cases} \tag{4}$$

where $\log t \sim N(-1.2, 1.2^2)$, $X_0 \sim P_{X_0}$, $(X_0, C_1) \sim P_{X_0,C_1}$ and $p_{\text{non}} \in [0,1]$ is a predefined parameter controlling the probability of masking the condition C_1. The unconditional and conditional score functions are estimated as $s_\theta(X_t, t, \varnothing) := [nn_\theta(X_t, t, \varnothing) - X_t]/t^2$ and $s_\theta(X_t, t, C_1) := [nn_\theta(X_t, t, C_1) - X_t]/t^2$, respectively. These estimators enable the reverse process to generate samples that approximately follow P_{X_0} and $P_{X_0|C_1}$. By employing an ODE solver, EDM requires substantially fewer sampling steps than SDE-based methods (Ho et al., 2020; Song et al., 2021), enabling faster generation.

2.3 Classifier Guidance (CG) and Diffusion Posterior Sampling (DPS)

Dhariwal & Nichol (2021) proposed classifier-guided diffusion, a training-free method to add conditional control to an unconditional diffusion model. Using Bayes' rule, the conditional score function admits the decomposition:

$$\nabla \log p_t(X_t|C_1) = \nabla \log p_t(X_t) + \nabla \log p(C_1|X_t), \tag{5}$$

where $p(C_1|X_t)$ is the density of C_1 given noisy X_t. In this paper, we mainly discuss the case in which conditions C_1 and C_2 are continuous. Let $\widetilde{f}_1(X_t)$ approximate $\mathbb{E}[C_1|X_t]$. The density $p(C_1|X_t)$ is approximated by a Gaussian $N(\widetilde{f}_1(X_t), (2\lambda_1)^{-1} I_k)$, which yields the tractable form $\nabla \log p(C_1|X_t) \approx -\lambda_1 \nabla \|C_1 - \widetilde{f}_1(X_t)\|_2^2$, known as classifier guidance (CG). Incorporating CG into (5) yields:

$$\nabla \log p_t(X_t|C_1) \approx s_\theta(X_t, t, \varnothing) - \lambda_1 \nabla \|C_1 - \widetilde{f}_1(X_t)\|_2^2. \tag{6}$$

Here, λ_1, referred to as the guidance scale, is a hyperparameter that controls the strength of condition during generation. Its choice depends on the specific task and reflects a trade-off between conditional fidelity and sample quality: larger values of λ_1 enforce stronger adherence to the condition C_1 but may degrade sample quality.

When X_0 and C_1 satisfy the function-mapping relationship in (1), Chung et al. (2023) proposed Diffusion Posterior Sampling (DPS). In DPS, the mapping function f_1 is assumed to be known and operates on the posterior mean $\mathbb{E}[X_0|X_t]$ obtained via Tweedie projection:

$$\mathbb{E}[X_0|X_t] = X_t + t^2 \cdot \nabla \log p_t(X_t). \tag{7}$$

By replacing the intractable score function in (7) with its estimator $s_\theta(X_t, t, \varnothing)$, the posterior mean $\mathbb{E}[X_0|X_t]$ is estimated as $X_{0|t} := nn_\theta(X_t, t, \varnothing)$. Accordingly, the gradient $\nabla \log p(C_1|X_t)$ is approximated by $-\lambda_1 \nabla \|C_1 - f_1(X_{0|t})\|_2^2$. Even when f_1 is unknown, it can be approximated by a regressor $\widehat{f}_1(X_0)$; the estimator $-\lambda_1 \nabla \|C_1 - \widehat{f}_1(X_{0|t})\|_2^2$ is typically more accurate than $-\lambda_1 \nabla \|C_1 - \widetilde{f}_1(X_t)\|_2^2$. As a result, conditional samples generated by DPS often exhibit higher quality than those obtained via CG (6).

2.4 Classifier-Free Guidance (CFG)

Although inserting CG into an unconditional score allows conditional sampling with adjustable guidance scale, it can lead to a distorted conditional distribution with excessively complex C_1 Ho & Salimans (2022) or improper guidance scale λ_1 Wu et al. (2024). To address this issue, Ho & Salimans (2022) proposed classifier-free guidance (CFG), which achieves high-quality conditional sampling while retaining control over the conditioning strength. CFG replaces the second gradient term in (5) with $\lambda_1\big(\nabla \log p_t(X_t|C_1) - \nabla \log p_t(X_t)\big)$, resulting in:

$$\nabla \log p_t(X_t|C_1) \approx (1 - \lambda_1) s_\theta(X_t, t, \varnothing) + \lambda_1 s_\theta(X_t, t, C_1).$$

Compared to CG, CFG is capable of handling more intricate conditions during the reverse process and typically generates higher-quality samples, particularly in image generation.

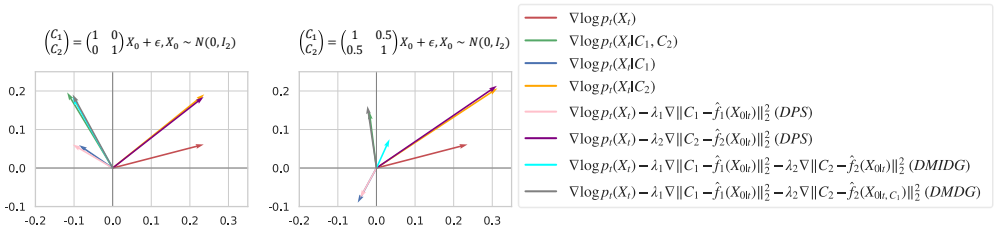

Figure 1: Comparison of score functions when $t = 1$. In the left panel, $C_1 \perp\!\!\!\perp C_2$, and both DMDG and DMIDG can accurately approximate the conditional score function $\nabla \log p_t(X_t|C_1, C_2)$. In the right panel, $C_1 \not\perp\!\!\!\perp C_2$, and DMIDG can no longer approximate the correct score function, whereas DMDG remains effective. The error of DMIDG comes from the wrong assumption $C_1 \perp\!\!\!\perp C_2|X_t$.

3 Methods of Double Guidance

This section first review the previous double guidance method and its theoretical flaw, then explains our proposed methods: Diffusion model with double guidance (DMDG) and Diffusion model with hybrid guidance (DMHG).

3.1 Existing methods

Generating samples from $P_{X_0|C_1,C_2}$ via CG, DPS, or CFG requires either modeling the conditional densities $p(C_1, C_2|X_t)$, $p(C_1, C_2|\mathbb{E}[X_0|X_t])$ or the conditional score $\nabla \log p_t(X_t|C_1, C_2)$, respectively. However, as shown in Table 1, the lack of joint samples of (X_0, C_1, C_2) renders CG, DPS and CFG infeasible.

Prior to our work, Ye et al. (2024, Appendix D.4) proposed to approximate $p(C_1, C_2|X_t)$ by $p(C_1|X_t)p(C_2|X_t)$ which does not require joint samples of (X_0, C_1, C_2). However, this approximation assumes $C_1 \perp\!\!\!\perp C_2|X_t$, an assumption that does not hold in general. Even if $C_1 \perp\!\!\!\perp C_2|X_0$ hold in general, the diffused X_t may not give the conditional independence. As illustrated in Figure 1, 3 and Appendix D.1, such an assumption can cause error in the estimated score function. We refer to this approach as Diffusion Model with *Independent* Double Guidance (DMIDG), while the original authors term it *combined guidance*. Detailed descriptions can be found in Appendix D.1.

3.2 Diffusion model with double guidance (DMDG)

Motivated by the limitations of previous methods, we develop a novel guidance method to decompose the conditional score, avoiding the use of the joint samples of (X_0, C_1, C_2). Inspired by (6) and DPS, we decompose the conditional score function as:

$$\begin{aligned} \nabla \log p_t(X_t|C_1, C_2) &= \nabla \log p_t(X_t) + \nabla \log p(C_1|X_t) + \nabla \log p(C_2|X_t, C_1) \\ &\approx \nabla \log p_t(X_t) + \nabla \log p(C_1|\mathbb{E}[X_0|X_t]) \\ &\quad + \nabla \log p(C_2|\mathbb{E}[X_0|X_t, C_1]), \end{aligned} \tag{8}$$

where $p(C_1|\mathbb{E}[X_0|X_t])$ and $p(C_2|\mathbb{E}[X_0|X_t, C_1])$ are the densities of $N(f_1(\mathbb{E}[X_0|X_t]), \sigma_1^2 I_k)$ and $N(f_2(\mathbb{E}[X_0|X_t, C_1]), \sigma_2^2 I_{p-k})$, respectively. The posterior means $\mathbb{E}[X_0|X_t]$ and $\mathbb{E}[X_0|X_t, C_1]$, obtained via Tweedie projection (7), are estimated by $X_{0|t} = nn_\theta(X_t, t, \varnothing)$ and $X_{0|t,C_1} := nn_\theta(X_t, t, C_1)$, respectively. During the reverse process, we need to compute $f_1, f_2, \nabla f_1$ and ∇f_2 multiple times. When these functions are unknown, non-differentiable, or costly, we instead train neural regressors (or classifiers) $\widehat{f}_1$ and $\widehat{f}_2$ to approximate f_1 and f_2. Subsequently, similarly to (6), $p(C_1|\mathbb{E}[X_0|X_t])$ and $p(C_2|\mathbb{E}[X_0|X_t, C_1])$ are approximated by $p(C_1|X_{0|t})$ and $p(C_2|X_{0|t,C_1})$, respectively. Equivalently, they are modeled by the following normal distributions:

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \sim N\left(\begin{bmatrix} \widehat{f}_1(X_{0|t}) \\ \widehat{f}_2(X_{0|t,C_1}) \end{bmatrix}, \begin{bmatrix} \frac{1}{2\lambda_1} I_k & 0 \\ 0 & \frac{1}{2\lambda_2} I_{p-k} \end{bmatrix} \right), \tag{9}$$

where λ_1, λ_2 are positive guidance scales used to adjust the strength of guidance. Finally, replacing $\nabla \log p_t(X_t)$ in (8) with $s_\theta(X_t, t, \varnothing)$, we get Diffusion Model with Double Guidance (DMDG):

$$\nabla \log p_t(X_t|C_1, C_2) \approx s_\theta(X_t, t, \varnothing) - \lambda_1 \nabla \|C_1 - \widehat{f}_1(X_{0|t})\|_2^2 - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1})\|_2^2. \quad (10)$$

Compared to DPS, our DMDG introduces an additional guidance term related to C_2, $-\lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1})\|_2^2$. This term enables explicit control over C_2, while implicitly preserving the correlation between C_1 and C_2 through $\widehat{f}_2(X_{0|t,C_1})$. In addition, DMDG does not require joint samples (X_0, C_1, C_2) because the regressors (or classifiers) $\widehat{f}_1$ and $\widehat{f}_2$ can be trained on distinct datasets separately. Thus, DMDG circumvents the challenges posed by block-wise missing structure in aggregated datasets. Substituting $\widehat{f}_2(X_{0|t,C_1})$ in (10) with $\widehat{f}_2(X_{0|t})$ results in DMIDG. The detailed derivation of (8) is presented in Appendix A, which does not violate $C_1 \not\perp\!\!\!\perp C_2|X_t$.

3.3 Diffusion Model with Hybrid Guidance (DMHG)

Similarly to CFG, we can replace $\nabla \log p(C_1|X_t)$ in (8) by $\lambda_1(\nabla \log p_t(X_t|C_1) - \nabla \log p_t(X_t))$. Following the first equality in (8) and the Gaussian modeling in (9), we obtain:

$$\nabla \log p_t(X_t|C_1, C_2) \approx (1-\lambda_1) s_\theta(X_t, t, \varnothing) + \lambda_1 s_\theta(X_t, t, C_1) - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1})\|_2^2. \quad (11)$$

We term this approach the Diffusion Model with Hybrid Guidance (DMHG), as it incorporates both CG and CFG. Similarly to CFG, DMHG exhibits improved fidelity to the target conditional distribution $P_{X_0|C_1,C_2}$ when condition C_1 is highly complex. Similarly to DMIDG, replacing $\widehat{f}_2(X_{0|t,C_1})$ in (11) with $\widehat{f}_2(X_{0|t})$ results in Diffusion Model with Independent Hybrid Guidance (DMIHG). The complete sampling algorithm for both DMDG and DMHG is provided in Algorithm 1. Extensions of DMDG and DMHG to settings involving more than two aggregated datasets and multiple conditions are presented in Appendix B.

4 Synthetic Simulation: A Gaussian Mixture Example

To enable an analytic evaluation of the generation performance, we conduct simulations on a Gaussian mixture model. The target variable is $X_0 \in \mathbb{R}^5$, $X_0 \sim \sum_{k=1}^K w_k N(m_k, \Sigma_k)$ ($K = 10$) and the function mapping is a linear transformation:

$$Y := (Y_1, Y_2, Y_3, Y_4, Y_5)^\top = AX_0 + \sigma\eta, \eta \sim N(0, I_5),$$

where $A \in \mathbb{R}^{5\times 5}$ is a fixed matrix and $\sigma > 0$ is the standard deviation. Given $Y = y$, the posterior distribution of X_0 can be analytically computed:

$$p_{X_0|Y=y} \propto \sum_{k=1}^K \widetilde{w}_k N(\widetilde{m}_k, \widetilde{\Sigma}_k), \quad (12)$$

where $\widetilde{\Sigma}_k = (A^\top A/\sigma^2 + \Sigma_k^{-1})^{-1}$, $\widetilde{m}_k = \widetilde{\Sigma}_k(A^\top y/\sigma^2 + \Sigma_k^{-1} m_k)$, and $\widetilde{w}_k = w_k \cdot \exp[0.5(\widetilde{m}_k^\top \widetilde{\Sigma}_k^{-1} \widetilde{m}_k - m_k^\top \Sigma_k^{-1} m_k)]/\sqrt{|\Sigma_k|}$ (Hagemann et al., 2022). We set $C_1 = (Y_1, Y_2)^\top$, $C_2 = (Y_3, Y_4, Y_5)^\top$, and each dataset $D^{(1)}$ and $D^{(2)}$ contains 10000 samples. With (12), we can accurately evaluate the distance between the generated samples and the true conditional distribution.

We compare DMDG and DMHG with a diverse set of baselines, including the CG-based DMIDG (Ye et al., 2024) and the CFG-based DMIHG. We also consider methods based on compositional score functions, such as Composition (Liu et al., 2022) and COIND (Gaudi et al., 2025). In addition, we include CTRL (Zhao et al., 2025b), which leverages RL and ControlNet to fine-tune a pretrained conditional diffusion model. We further evaluate four imputation-based approaches, which first recover missing conditions in the aggregated datasets and then train a CDM: regressor imputation, GAIN (Yoon et al., 2018), Forest Diffusion (Jolicoeur-Martineau et al., 2024), and KNN imputation. We consider the following two settings:

Setting I ($C_1 \perp\!\!\!\perp C_2$): $\Sigma_k = 10^{-2} I$ and $m_k \sim \mathcal{U}([-1, 1]^5)$ for all k, and $\sigma = 0.1$. $A = (a_{ij})$ is diagonal with $a_{ii} = 0.1/i$.

Setting II ($C_1 \not\perp\!\!\!\perp C_2$): $\Sigma_k = I$ and $m_k \sim \mathcal{U}([-5, 5]^5)$ for all k, and $\sigma = 1$. $A = (a_{ij})$ with $a_{ii} = 0.5$ and $a_{ij} = 0.25$ for $i \neq j$.

Table 2: Means and standard deviations of 2-Wasserstein (W_2) distances of each methods (over 50 repeats) under Setting I and II with 1000 samples. All relevant parameters are tuned to minimize the W_2 distance. The best results are highlighted in bold, the second-best results are underlined, and the third-best results are dash-underlined.

Method	**Setting I**		**Setting II**	
	W_2 ($\times 10^2$) ($\downarrow$)	Guidance scales	W_2 ($\downarrow$)	Guidance scales
DMDG (10)	3.032(1.568)	$\lambda_1 = \lambda_2 = 180$	3.275(3.411)	$\lambda_1 = \lambda_2 = 1.5$
DMIDG	3.095(1.646)	$\lambda_1 = \lambda_2 = 180$	4.688(6.447)	$\lambda_1 = \lambda_2 = 2$
DMHG (11)	**2.097**(0.830)	$\lambda_1 = 1, \lambda_2 = 250$	**2.386**(2.923)	$\lambda_1 = \lambda_2 = 1.5$
DMIHG	2.162(1.106)	$\lambda_1 = 1, \lambda_2 = 250$	4.919(6.415)	$\lambda_1 = \lambda_2 = 1.5$
Composition	23.452(16.471)	$\lambda_1 = \lambda_2 = 1$	22.432(10.369)	$\lambda_1 = \lambda_2 = 1$
COIND*	2.839(1.049)	N/A	2.746(2.993)	N/A
CTRL	4.057(2.949)	N/A	5.620(6.045)	N/A
Regressor	157.645(94.224)	N/A	6.800(3.818)	N/A
GAIN	26.312(14.977)	N/A	6.169(2.138)	N/A
Forest Diffusion	17.639(18.648)	N/A	8.210(6.747)	N/A
KNN	5.591(4.790)	N/A	5.469(4.655)	N/A

N/A: Guidance scales are not available; *: COIND is trained on full datasets without missing.

The results are summarized in Table 2. In both Settings I and II, DMHG and DMIHG exhibit clear advantages, indicating that the CFG-based methods are more effective at matching the target conditional distribution (except COIND, which is trained on full datasets). The CG-based methods, DMDG and DMIDG, perform slightly worse than their CFG-based counterparts, which is consistent with findings reported in prior work (Ho & Salimans, 2022; Wu et al., 2024). In addition, CG-based methods consistently outperform imputation-based and RL-based approaches. In Setting I, due to the independence of C_1 and C_2, DMDG and DMHG perform similarly to their counterparts DMIDG and DMIHG, although our methods exhibit lower variances. In Setting II, the structure of matrix A induces dependency between C_1 and C_2, leading to a substantial performance gap: both DMDG and DMHG significantly outperform DMIDG and DMIHG. This observation is supported by the phenomenon shown in right panel of Figure 1 that when $C_1 \not\perp\!\!\!\perp C_2$, DMIDG introduces errors in the score estimation. The performances of Composition are not ideal due to the violation of conditional independence, as we mentioned in Appendix D.

5 Molecule Generation Task

5.1 The aggregated datasets

We aggregate two datasets for molecular generation: GEOM-DRUG and ZINC250k. GEOM-DRUG contains more than 210,000 drug-like molecular conformers annotated with their energetic properties (Axelrod & Gómez-Bombarelli, 2022). We focus on lowest energy, ensemble energy, and ensemble entropy, which characterize molecular stability and flexibility required for receptor binding. An ideal drug candidate typically exhibits low values of the first two and moderately low values of the third (Fogolari et al., 2018). The latter dataset, ZINC250k consists of approximately 250,000 molecules annotated with drug-likeness properties (Sterling & Irwin, 2015). We selected logP, QED, and SAS as representative properties, as promising drug candidates are generally expected to meet desirable thresholds of these metrics (Ertl & Schuffenhauer, 2009). We aggregate GEOM-DRUG and ZINC250k to train the diffusion models, and separately train $\widehat{f}_1$ and $\widehat{f}_2$ using their respective datasets.

5.2 De novo drug design using DMDG

De novo drug design refers to the generation of entirely new molecular structures. The aim is to generate molecules that exhibit both energetic stability and high drug-likeness. We use the proposed diffusion models with two datasets; we set $D^{(1)}$ as GEOM-DRUG, with $C_1 = (\text{lowest energy}, \text{ensemble energy}, \text{ensemble entropy})$, and $D^{(2)}$ as ZINC250k, with $C_2 =$

(logP, QED, SAS). We consider two tasks with the following conditions:

$$\begin{aligned} &\text{Task 1: Lowest energy, Ensemble energy} \in [q_{0.1}, q_{0.4}],\ \text{Ensemble entropy} \in [q_{0.25}, q_{0.5}]; \\ &\text{Task 2: logP} \in [0, 4],\ \text{QED} \in [0.7, 1],\ \text{SAS} \in [0, 5], \end{aligned} \tag{13}$$

where each q_α in Task 1 denotes the α-quantiles of the corresponding property.

The baselines are the same with these in Section 4, except from COIND, which requires a dataset without missing values. All diffusion models operate in the latent space learned by COATI (Kaufman et al., 2024b), a powerful variational autoencoder (VAE). For each method, we generate 5,000 molecules, with their properties guided toward the midpoints of the specified intervals. A generation is considered successful if all the six molecular properties *simultaneously* satisfy the constraints of Tasks 1 and 2. In addition, we evaluate the W_2 distances between the generated molecules and the target conditional distributions. Unless otherwise specified, all diffusion models use 18 sampling steps, following Karras et al. (2022). Further details of the implementation and compared methods are provided in Appendix D.

The results are reported in Figure 2. CG-based methods, DMDG and DMIDG, achieve relatively high success rates. DMDG outperforms DMIDG by a substantial margin (76.0% vs. 58.7%), highlighting the importance of accounting for the dependence between C_1 and C_2 given X_t. At the same success rate, DMDG also attains a significantly smaller W_2 distance from the GEOM-DRUG dataset. In contrast, CFG-based methods exhibit limited improvement in task success, with all success rates remaining below 25%, consistent with the observations reported by Kaufman et al. (2024a). Compositional and imputation-based methods similarly demonstrate limited effectiveness, with success rates also generally below 25%. CTRL, the RL-based approach fine-tuned on ZINC250k, achieves a relatively low W_2 distance from ZINC250k; however, its success rate remains below 20%.

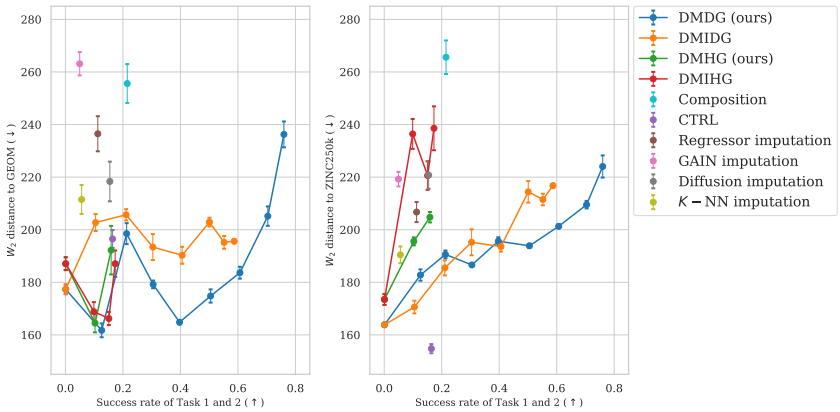

Figure 2: Success rate of Tasks 1 & 2 (13) and W_2 distances to GEOM-DRUG and ZINC250k. The results of DMDG, DMIDG, DMHG, and DMIHG are shown as curves, since they involve tunable parameters λ_1 and λ_2. Error bars represent one standard deviation across 5000 generated molecules.

6 Conclusion

In this paper, we propose a double guidance framework and two specific instantiations, DMDG and DMHG, to enable joint conditional generation on aggregated datasets with block-wise missing. The proposed DMDG achieves superior joint-condition controllability in simulation and the molecular generation task. Combining our methods with more advanced techniques of guided diffusion, self-adaptive guidance scales, and deploying them on larger and more aggregated datasets constitute our future directions.

References

Simon Axelrod and Rafael Gómez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Arwen Bradley, Preetum Nakkiran, David Berthelot, James Thornton, and Joshua M. Susskind. Mechanisms of projective composition of diffusion models. In *International Conference on Machine Learning*, 2025.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023.

Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Earnest Oghenesuvwe Erhirhie, Chibueze Peter Ihekwereme, and Emmanuel Emeka Ilodigwe. Advances in acute toxicity testing: Strengths, weaknesses and regulatory acceptance. *Interdisciplinary Toxicology*, 11(1):5–12, 2018.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, 2009.

Francisco J Fernández, Javier Querol-García, Sergio Navas-Yuste, Fabrizio Martino, and M Cristina Vega. X-ray crystallography for macromolecular complexes. *Advances in Experimental Medicine and Biology*, 3234:125–140, 2024.

Federico Fogolari, Alessandra Corazza, and Gennaro Esposito. Free energy, enthalpy and entropy from implicit solvent end-point simulations. *Frontiers in Molecular Biosciences*, 5:11, 2018.

Hengyu Fu, Zhuoran Yang, Mengdi Wang, and Minshuo Chen. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv:2403.11968*, 2024.

Sachit Gaudi, Gautam Sreekumar, and Vishnu Boddeti. Coind: Enabling logical compositions in diffusion models. In *International Conference on Learning Representations*, 2025.

Niklas W. A. Gebauer, Michael Gastegger, Stefaan S. P. Hessmann, Klaus-Robert Müller, and Kristof T. Schütt. Inverse design of 3d molecular structures with conditional generative neural networks. *Nature Communications*, 13(1):973, 2022. ISSN 2041-1723.

Manuel Gloeckler, Shoji Toyota, Kenji Fukumizu, and Jakob H. Macke. Compositional simulation-based inference for time series. In *International Conference on Learning Representations*, 2025.

Paul Hagemann, Johannes Hertrich, and Gabriele Steidl. Stochastic normalizing flows for inverse problems: A markov chains viewpoint. *SIAM/ASA Journal on Uncertainty Quantification*, 10(3): 1162–1190, 2022.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, et al. Video diffusion models. In *Neural Information Processing Systems*, volume 35, 2022.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, volume 162, pp. 8867–8887, 2022.

Zijing Hu, Fengda Zhang, Long Chen, Kun Kuang, Jiahui Li, Kaifeng Gao, Jun Xiao, Xin Wang, and Wenwu Zhu. Towards better alignment: Training diffusion models with reinforcement learning against sparse rewards. In *Computer Vision and Pattern Recognition*, 2025.

Martin F. Jarrold. Applications of charge detection mass spectrometry in molecular biology and biotechnology. *Chemical Reviews*, 122(8):7415–7441, 2022. ISSN 0009-2665.

Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees. In *International Conference on Artificial Intelligence and Statistics*, volume 238, pp. 1288–1296, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Benjamin Kaufman, Edward C. Williams, Ryan Pederson, Carl Underkoffler, Zahid Panjwani, Miles Wang-Henderson, Narbe Mardirossian, Matthew H. Katcher, Zack Strater, Jean-Marc Grandjean, Bryan Lee, and John Parkhill. Latent diffusion for conditional generation of molecules. *bioRxiv*, 2024a.

Benjamin Kaufman, Edward C. Williams, Carl Underkoffler, Ryan Pederson, Narbe Mardirossian, Ian Watson, and John Parkhill. COATI: Multimodal contrastive pretraining for representing and traversing chemical space. *Journal of Chemical Information and Modeling*, 64(4):1145–1157, 2024b. ISSN 1549-9596.

Ryotaro Kawata, Kazusato Oko, Atsushi Nitanda, and Taiji Suzuki. Direct distributional optimization for provable alignment of diffusion models. In *International Conference on Learning Representations*, 2025.

Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12861–12872, 2020.

Haote Li, Yu Shee, Brandon Allen, Federica Maschietto, Anton Morgunov, and Victor Batista. Kernel-elastic autoencoder for molecular design. *PNAS Nexus*, 3(4):168, 2024.

Shuai Li, Ziqi Chen, Hongtu Zhu, Christina Dan Wang, and Wang Wen. Nearest-neighbor sampling based conditional independence testing. *AAAI Conference on Artificial Intelligence*, 37(7):8631–8639, 2023a.

Shuai Li, Yingjie Zhang, Hongtu Zhu, Christina Wang, Hai Shu, Ziqi Chen, Zhuoran Sun, and Yanfeng Yang. K-nearest-neighbor local sampling based conditional independence testing. In *Neural Information Processing Systems*, volume 36, pp. 23321–23344, 2023b.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Computer Vision Association*, pp. 423–439, 2022.

Matteo Ninniri, Marco Podda, and Davide Bacciu. Classifier-free graph diffusion for molecular property targeting. In *Machine Learning and Knowledge Discovery in Databases*, pp. 318–335, 2024.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv:2112.10752*, 2021.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Lit, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Neural Information Processing Systems*, 2022.

Pedro Sanchez and Sotirios A. Tsaftaris. Diffusion causal models for counterfactual estimation. In *Conference on Causal Learning and Reasoning*, volume 177, pp. 647–668, 2022.

Yuchen Shen, Chenhao Zhang, Sijie Fu, Chenghui Zhou, Newell Washburn, and Barnabas Poczos. Chemistry-inspired diffusion with non-differentiable guidance. In *International Conference on Learning Representations*, 2025.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, volume 202, pp. 32483–32498, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Teague Sterling and John J. Irwin. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015.

Yuchen Wu, Minshuo Chen, Zihao Li, Mengdi Wang, and Yuting Wei. Theoretical insights for diffusion guidance: A case study for gaussian mixture models. In *International Conference on Machine Learning*, 2024.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *International Conference on Learning Representations*, 2025.

Yanfeng Yang, Siwei Chen, Pingping Hu, Zhaotong Shen, Yingjie Zhang, Zhuoran Sun, Shuai Li, Ziqi Chen, and Kenji Fukumizu. Conditionally whitened generative models for probabilistic time series forecasting. *arXiv:2509.20928*, 2025a.

Yanfeng Yang, Shuai Li, Yingjie Zhang, Zhuoran Sun, Hai Shu, Ziqi Chen, and Renming Zhang. Conditional diffusion models based conditional independence testing. In *AAAI Conference on Artificial Intelligence*, 2025b.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *Neural Information Processing Systems*, 2024.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *International Conference on Machine Learning*, 2025.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, volume 80, pp. 5689–5698, 2018.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *International Conference on Computer Vision*, pp. 3813–3824, 2023.

Jiankun Zhao, Bowen Song, and Liyue Shen. CoSIGN: Few-step guidance of consistency model to solve general inverse problems. In *European Conference on Computer Vision*, pp. 108–126, 2025a.

Yulai Zhao, Masatoshi Uehara, Gabriele Scalia, Sunyuan Kung, Tommaso Biancalani, Sergey Levine, and Ehsan Hajiramezanali. Adding conditional control to diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2025b.

A Derivation and Algorithm of DMDG and DMHG

First, we show how to approximate the conditional score $\nabla \log p_t(X_t|C_1, C_2)$ in (8):

$$
\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2) \\
&= \nabla \log p_t(X_t) + \nabla \log p(C_1, C_2|X_t) \\
&= \nabla \log p_t(X_t) + \nabla \log p(C_2|X_t, C_1) + \nabla \log p(C_1|X_t) \\
&= \nabla \log p_t(X_t) + \nabla \log \int p(C_2, X_0|X_t, C_1) dX_0 + \nabla \log \int p(C_1, X_0|X_t) dX_0 \\
&= \nabla \log p_t(X_t) + \nabla \log \int p(C_2|X_0, X_t, C_1) p(X_0|X_t, C_1) dX_0 \\
&\quad + \nabla \log \int p(C_1|X_0, X_t) p(X_0|X_t) dX_0 \\
&\overset{(*)}{=} \nabla \log p_t(X_t) + \nabla \log \int p(C_2|X_0) p(X_0|X_t, C_1) dX_0 + \nabla \log \int p(C_1|X_0) p(X_0|X_t) dX_0 \\
&\overset{(**)}{\approx} \nabla \log p_t(X_t) + \nabla \log p(C_2|\mathbb{E}\left[X_0|X_t, C_1\right]) + \nabla \log p(C_1|\mathbb{E}\left[X_0|X_t\right]). \qquad (14)
\end{aligned}
$$

In $(*)$, we use the property that $C_2 \perp\!\!\!\perp (X_t, C_1)|X_0$, as shown in Figure 3. For $(**)$, following Chung et al. (2023), we approximate $\int p(C_1|X_0)p(X_0|X_t)dX_0$ and $\int p(C_2|X_0)p(X_0|X_t, C_1)dX_0$ by:

$$
\begin{aligned}
&\int p(C_1|X_0)p(X_0|X_t)dX_0 = \mathbb{E}_{X_0 \sim P_{X_0|X_t}}\left[p(C_1|X_0)\right] \approx p(C_1|\mathbb{E}\left[X_0|X_t\right]), \\
&\int p(C_2|X_0)p(X_0|X_t, C_1)dX_0 = \mathbb{E}_{X_0 \sim P_{X_0|X_t, C_1}}\left[p(C_2|X_0)\right] \approx p(C_2|\mathbb{E}\left[X_0|X_t, C_1\right]).
\end{aligned}
$$

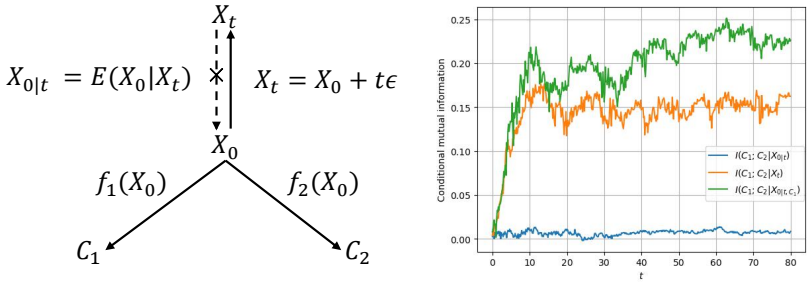

Figure 3: Left panel: Conditional independence relationships among X_0, X_t, C_1 and C_2: X_t, C_1, C_2 are mutually independent given X_0, and $C_1 \perp\!\!\!\perp C_2|X_t$ and $C_1 \perp\!\!\!\perp C_2|X_{0|t}$ are not true. In addition, DMDG and DMHG do not assume either $C_1 \perp\!\!\!\perp C_2$ or $C_1 \not\perp\!\!\!\perp C_2$, as our proposed methods are applicable to both cases. Right panel: Under the setting II in Section 4, the behavior of conditional mutual informations (CMI) over the scale of t in forward process (2).

The algorithm of DMDG and DMHG can be found in Algorithm 1.

B More General Missing Type and Diffusion Model with Triple Guidance (DMTG)

Our double guidance methods can be extended to process more aggregated datasets and missing conditions. Table 3 presents the two additional missing types: **missing type I**, where overlapping conditions exist, and **missing type II**, which has three aggregated datasets with non-overlapping conditions.

Algorithm 1 Conditional sampling by DMDG or DMHG

Input: Target conditions C_1, C_2, a trained score network nn_θ, regressors $\widehat{f}_1, \widehat{f}_2$, guidance scale λ_1, λ_2, a time schedule $t_0, t_2, \ldots, t_N$ with $t_0 = t_{\min}, t_N = t_{\max}$, sampling method: DMDG or DMHG.
Output: A sample approximately follows $P_{X_0|C_1,C_2}$.

Draw $X_{t_N} \sim N(0, t_N^2 I_d)$
for i in $N-1, N-2, \ldots, 0$ **do**
  Let $X_{0|t_{i+1}} = nn_\theta(X_{t_{i+1}}, t_{i+1}, \varnothing)$, $X_{0|t_{i+1},C_1} = nn_\theta(X_{t_{i+1}}, t_{i+1}, C_1)$
  Let $s_{i+1,\varnothing} = \dfrac{X_{0|t_{i+1}} - X_{t_{i+1}}}{t_{i+1}^2}$, $s_{i+1,C_1} = \dfrac{X_{0|t_{i+1},C_1} - X_{t_{i+1}}}{t_{i+1}^2}$
  if sampling method = DMDG **then**
    Let $d_{i+1} = s_{i+1,\varnothing} - \lambda_1 \nabla_{X_{t_{i+1}}} \|C_1 - \widehat{f}_1(X_{0|t_{i+1}})\|_2^2 - \lambda_2 \nabla_{X_{t_{i+1}}} \|C_2 - \widehat{f}_2(X_{0|t_{i+1},C_1})\|_2^2$
  else if sampling method = DMHG **then**
    Let $d_{i+1} = (1-\lambda_1) s_{i+1,\varnothing} + \lambda_1 s_{i+1,C_1} - \lambda_2 \nabla_{X_{t_{i+1}}} \|C_2 - \widehat{f}_2(X_{0|t_{i+1},C_1})\|_2^2$
  end if
  $X_{t_i} = X_{t_{i+1}} - (t_i - t_{i+1}) \cdot t_{i+1} \cdot d_{i+1}$
end for
return X_{t_0}

Table 3: Illustration of more general block-wise missing patterns.

(a) Missing Type I (overlapping conditions)

Dataset	Target	Condition 1	Condition 2	Condition 3
$D^{(1)}$	$X_0^{(1)}$	$C_1^{(1)}$	$\varnothing$	$C_3^{(1)}$
$D^{(2)}$	$X_0^{(2)}$	$\varnothing$	$C_2^{(2)}$	$C_3^{(2)}$

(b) Missing Type II (three datasets with non-overlapping conditions)

Dataset	Target	Condition 1	Condition 2	Condition 3
$D^{(1)}$	$X_0^{(1)}$	$C_1^{(1)}$	$\varnothing$	$\varnothing$
$D^{(2)}$	$X_0^{(2)}$	$\varnothing$	$C_2^{(2)}$	$\varnothing$
$D^{(3)}$	$X_0^{(3)}$	$\varnothing$	$\varnothing$	$C_3^{(3)}$

For missing type I, we can approximate the true conditional score $\nabla \log p_t(X_t|C_1, C_2, C_3)$ by:

$$\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2, C_3) \\
&= \nabla \log p_t(X_t|C_3) + \nabla \log p(C_1|X_t, C_3) + \nabla \log p(C_2|X_t, C_1, C_3) \\
&\approx s_\theta(X_t, t, \varnothing, C_3) - \lambda_1 \nabla \|C_1 - \widehat{f}_1(X_{0|t,C_3})\|_2^2 - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1,C_3})\|_2^2, \qquad (15)
\end{aligned}$$

or

$$\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2, C_3) \\
&\approx (1-\lambda_1) s_\theta(X_t, t, \varnothing, C_3) + \lambda_1 s_\theta(X_t, t, C_1, C_3) - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1,C_3})\|_2^2, \qquad (16)
\end{aligned}$$

where the neural network nn_θ now have four inputs: X_t, t, C_1 and C_3. The latter two can be masked if unavailable. Accordingly, we define $X_{0|t,C_3} := nn_\theta(X_t, t, \varnothing, C_3)$, $X_{0|t,C_1,C_3} := nn_\theta(X_t, t, C_1, C_3)$, $s_\theta(X_t, t, \varnothing, C_3) := [nn_\theta(X_t, t, \varnothing, C_3) - X_t]/t^2$ and $s_\theta(X_t, t, C_1, C_3) := [nn_\theta(X_t, t, C_1, C_3) - X_t]/t^2$. The derivations of (15) and (16) are similar to (14). Since the joint samples of (X_0, C_1, C_3) exist, $X_{0|t,C_3}$ and $X_{0|t,C_1,C_3}$ are learnable. Conceptually, they are equivalent to DMDG and DMHG, with the only difference being the inclusion of C_3 in the network input. Therefore, we refer to (15) and (16) as **DMDG-I** and **DMHG-I**, respectively. By replacing $X_{0|t,C_1,C_3}$ with $X_{0|t,C_3}$, we obtain DMIDG-I and DMIHG-I.

We can also apply our method to more extreme block-wise missing datasets. For the dataset with missing type II illustrated in Table 3 **(b)**, the conditional score function can be decomposed as:

$$\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2, C_3) \\
&= \nabla \log p(X_t|C_1, C_2) + \nabla \log \int p(C_3|X_0)p(X_0|X_t, C_1, C_2)dX_0 \\
&\approx \nabla \log p(X_t|C_1, C_2) + \nabla \log p(C_3|\mathbb{E}\left[X_0|X_t, C_1, C_2\right]) \qquad (17)
\end{aligned}$$

In (17), DMDG (10) or DMHG (11) can be directly used to approximate $\nabla \log p(X_t|C_1, C_2)$. The challenge of this missing type comes from how to estimate the posterior mean $\mathbb{E}\left[X_0|X_t, C_1, C_2\right]$. Our solution originates from the Tweedie projection (7):

$$\mathbb{E}[X_0|X_t, C_1, C_2] = X_t + t^2 \cdot \nabla \log p_t(X_t|C_1, C_2), \qquad (18)$$

where $\nabla \log p_t(X_t|C_1, C_2)$ can be directly estimated using DMDG (10) or DMHG (11). This yields the following estimations of $\mathbb{E}[X_0|X_t, C_1, C_2]$:

$$\begin{aligned}
X^{\text{DMDG}}_{0|t,C_1,C_2} &:= X_{0|t} - \lambda_1 t^2 \nabla \|C_1 - \widehat{f}_1(X_{0|t})\|_2^2 - \lambda_2 t^2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1})\|_2^2, \\
X^{\text{DMHG}}_{0|t,C_1,C_2} &:= (1-\lambda_1)X_{0|t} + \lambda_1 X_{0|t,C_1} - \lambda_2 t^2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1})\|_2^2. \qquad (19)
\end{aligned}$$

These two estimates can also be interpreted as performing two gradient descent from $X_{0|t}$, and this estimation strategy can potentially be extended to more aggregated datasets. Then, (17) can be estimated by:

$$\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2, C_3) \\
&\approx s_\theta(X_t, t, \varnothing) - \lambda_1 \nabla \|C_1 - \widehat{f}_1(X_{0|t})\|_2^2 - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1})\|_2^2 \\
&\quad - \lambda_3 \nabla \|C_3 - \widehat{f}_3(X^{\text{DMDG}}_{0|t,C_1,C_2})\|_2^2, \qquad (20)
\end{aligned}$$

or

$$\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2, C_3) \\
&\approx (1-\lambda_1)s_\theta(X_t, t, \varnothing) + \lambda_1 s_\theta(X_t, t, C_1) - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t,C_1})\|_2^2 \\
&\quad - \lambda_3 \nabla \|C_3 - \widehat{f}_3(X^{\text{DMHG}}_{0|t,C_1,C_2})\|_2^2. \qquad (21)
\end{aligned}$$

We refer to (20) as **Diffusion Model with Triple Guidance (DMTG)**, and (21) as **Diffusion Model with Triple Hybrid Guidance (DMTHG)**. Replacing $X_{0|t,C_1}$, $X^{\text{DMDG}}_{0|t,C_1,C_2}$ and $X^{\text{DMHG}}_{0|t,C_1,C_2}$ with $X_{0|t}$ results in Diffusion Model with Independent Triple Guidance (DMITG) and Diffusion Model with Independent Triple Hybrid Guidance (DMITHG). However, the estimations of $\mathbb{E}[X_0|X_t, C_1, C_2]$ (19) introduce a serious issue: when the scale of t is large, these estimations can lead to numerical explosion. To address this, we apply the guidance of C_3 only when $t \leq 1$, which effectively prevents numerical instability while still enabling control over C_3. Consequently, in our simulations of DMTG and DMTHG, the guidance scale λ_3 is larger than λ_1 and λ_2.

We conduct simulations based on the data generation mechanism from Setting II in Section 4 to evaluate the performance of DMDG-I, DMHG-I, DMTG and DMTHG. For missing type I, the sample sizes of $D^{(1)}$ and $D^{(2)}$ are all set to 10,000. For missing type II, sample sizes of $D^{(1)}$, $D^{(2)}$ and $D^{(3)}$ are 6,667. We define the condition variables as follows: $C_1 = (Y_1, Y_2)$, $C_2 = (Y_3, Y_4)$, and $C_3 = Y_5$. We compare guidance-based methods and imputation-based methods. CTRL can only handle two conditions so it is no longer suitable for triple conditions.

The results can be found in Table 4 and it should be interpreted in conjunction with Table 2. We observe that the datasets under missing type I (Table 3, (a)) exhibit the lowest level of missingness, while the datasets in Table 1 present a moderate degree, and the missing type II (Table 3, (b)) shows the highest level of missingness. Correspondingly, our proposed methods (DMDG, DMHG, DMDG-I, DMHG-I, DMTG, DMTHG), as well as imputation-based baselines, achieve lower W_2 distances under missing type I, moderate distances on the datasets in Table 1, and the highest W_2 distances under missing type II. However, methods of independent guidance perform relatively poorly under missing type II, as they do not take into account CI. Notably, DMTG and DMTHG remain effective, though the guidance of C_3 only exists when $t \leq 1$.

Table 4: W_2 distances and standard deviations of each methods under different missing types. All experiments are repeated 50 times and evaluated on 1000 samples. All parameters are tuned to minimize the W_2 distance. The best results are highlighted in bold, and the second-best results are underlined.

(a) Missing type I (overlapping conditions)

Method	W_2 ($\downarrow$)	Guidance scales
DMDG-I (15)	<u>3.056</u>(2.740)	$\lambda_1 = \lambda_2 = 1.5$
DMIDG-I	5.251(5.625)	$\lambda_1 = \lambda_2 = 1.5$
DMHG-I (16)	**2.147**(1.537)	$\lambda_1 = \lambda_2 = 1.5$
DMIHG-I	4.279(4.023)	$\lambda_1 = \lambda_2 = 1.5$
Regressor	5.136(4.696)	N/A
GAIN	5.467(3.879)	N/A
Forest Diffusion	5.854(5.067)	N/A
KNN	4.853(4.375)	N/A

(b) Missing type II (three datasets)

Method	W_2 ($\downarrow$)	Guidance scales
DMTG (10)	<u>3.857</u>(2.815)	$\lambda_1 = \lambda_2 = 1.5, \lambda_3 = 7$
DMITG	14.460(20.595)	$\lambda_1 = \lambda_2 = 1.5, \lambda_3 = 7$
DMTHG (11)	**3.808**(3.546)	$\lambda_1 = \lambda_2 = 1.5, \lambda_3 = 7$
DMITHG	4.819(5.887)	$\lambda_1 = \lambda_2 = 1.5, \lambda_3 = 7$
Regressor	9.122(4.750)	N/A
GAIN	16.623(14.677)	N/A
Forest Diffusion	12.760(10.391)	N/A
KNN	6.472(3.794)	N/A

C Applying DMDG and DMHG to Other Diffusion Models

To demonstrate the generality of DMDG and DMHG, we apply them to several classic diffusion models, including Denoising Diffusion Probabilistic Models (DDPM, Ho et al., 2020) and Score-Based Generative Models (SGM, Song et al., 2021).

The data generation procedure follows Setting II in Section 4. For the DDPM model, we adopt the default settings in (Ho et al., 2020). The SGM model follows the default configuration in (Yang et al., 2025b). The EDM model uses the same settings as in the main text, except that we set the sampling steps to 1000 in order to match the number of sampling steps used in DDPM and SGM.

The results can be found in Table 5. As shown, DMDG and DMHG are also applicable to both DDPM and SGM. A consistent observation across all models is that DMDG outperforms DMIDG, and DMHG outperforms DMIHG.

Table 5: Means and standard deviations of W_2 distances of each diffusion models and methods (over 50 repeats) under Setting II with 1000 samples. All relevant parameters are tuned to minimize the W_2 distance. The best results are highlighted in bold, and the second-best results are underlined.

Diffusion model	**Method**	W_2 ($\downarrow$)	Guidance scales
EDM (Karras et al., 2022)	DMDG (10)	<u>3.059</u>(3.573)	$\lambda_1 = \lambda_2 = 1.5$
	DMIDG	4.604(6.523)	$\lambda_1 = \lambda_2 = 2$
	DMHG (11)	**2.004**(2.257)	$\lambda_1 = \lambda_2 = 1.5$
	DMIHG	4.128(5.503)	$\lambda_1 = \lambda_2 = 1.5$
DDPM (Ho et al., 2020)	DMDG (10)	<u>3.498</u>(4.835)	$\lambda_1 = \lambda_2 = 6$
	DMIDG	6.189(10.586)	$\lambda_1 = \lambda_2 = 6$
	DMHG (11)	**1.894**(1.805)	$\lambda_1 = 1, \lambda_2 = 6$
	DMIHG	5.026(7.640)	$\lambda_1 = 1, \lambda_2 = 6$
SGM (Song et al., 2021)	DMDG (10)	3.878(3.156)	$\lambda_1 = \lambda_2 = 4.5$
	DMIDG	4.915(4.379)	$\lambda_1 = \lambda_2 = 4.5$
	DMHG (11)	**2.037**(0.920)	$\lambda_1 = 1, \lambda_2 = 4.5$
	DMIHG	<u>3.248</u>(2.534)	$\lambda_1 = 1, \lambda_2 = 4.5$

D Compared Methods in Detail

D.1 DMIDG and DMIHG

Unlike DMDG and DMHG, DMIDG and DMIHG model the joint density $p(C_1, C_2|X_t)$ as the product of two independent densities, $p(C_1|X_t)$ and $p(C_2|X_t)$. This simplification ignores the fact that C_1 and C_2 are generally not conditionally independent given X_t, leading to a biased estimation of the true conditional score, as shown in Figure 1. The deteriorated performances of DMIDG and DMIHG can be observed in Figure 2 and Table 2.

Algorithmically speaking, these methods replace the dependency-aware density $p(C_2|X_t, C_1)$ in (14) with $p(C_2|X_t)$, i.e.:

$$
\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2) \\
&\neq \nabla \log p_t(X_t) + \nabla \log p(C_1|X_t) + \nabla \log p(C_2|X_t) \\
&\approx \nabla \log p_t(X_t) + \nabla \log p(C_1|\mathbb{E}\left[X_0|X_t\right]) + \nabla \log p(C_2|\mathbb{E}\left[X_0|X_t\right]) \\
&\approx s_\theta(X_t, t, \varnothing) - \lambda_1 \nabla \|C_1 - \widehat{f}_1(X_{0|t})\|_2^2 - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t})\|_2^2 \qquad (22) \\
&\approx (1 - \lambda_1) s_\theta(X_t, t, \varnothing) + \lambda_1 s_\theta(X_t, t, C_1) - \lambda_2 \nabla \|C_2 - \widehat{f}_2(X_{0|t})\|_2^2, \qquad (23)
\end{aligned}
$$

where (22) and (23) denote the approximate conditional scores of DMIDG and DMIHG, respectively.

To better illustrate the importance of conditional independence, we use conditional mutual information (CMI) to visualize how the relationship between C_1 and C_2 evolves over t, conditioned on different variables. The CMI, denoted as $I(C_1; C_2|X_t)$, can be interpreted as a measure of the

conditional dependence between C_1 and C_2 given X_t. $I(C_1; C_2|X_t) = 0$ implies $C_1 \perp\!\!\!\perp C_2|X_t$. Conversely, $I(C_1; C_2|X_t) > 0$ means $C_1 \not\perp\!\!\!\perp C_2|X_t$ (Li et al., 2023a;b). It can be observed in Figure 3 that as t increases, both $I(C_1; C_2|X_t)$ and $I(C_1; C_2|X_{0|t})$ increase accordingly, indicating $C_1 \not\perp\!\!\!\perp C_2|X_t$ and $C_1 \not\perp\!\!\!\perp C_2|X_{0|t}$ when t becomes large. In contrast, $I(C_1; C_2|X_{0|t,C_1})$ remains close to zero across all values of t, since C_1 is conditioned.

D.2 Compositional methods

Compositional methods aim to combine different score functions to achieve conditional composition. Under the assumption of $C_1 \perp\!\!\!\perp C_1$, Liu et al. (2022) proposed to decompose the score function by:

$$\begin{aligned}
&\nabla \log p_t(X_t|C_1, C_2) \\
&= \nabla \log p_t(X_t) + \nabla \log p(C_1, C_2|X_t) \\
&\neq \nabla \log p_t(X_t) + \nabla \log p(C_1|X_t) + \nabla \log p(C_2|X_t) \qquad (24) \\
&= \nabla \log p_t(X_t) + \nabla \log \left[p(X_t|C_1)/p(X_t)\right] + \nabla \log \left[p(X_t|C_2)/p(X_t)\right] \\
&\approx s_\theta(X_t, t, \varnothing, \varnothing) + \lambda_1 \left[s_\theta(X_t, t, C_1, \varnothing) - s_\theta(X_t, t, \varnothing, \varnothing)\right] \\
&\quad + \lambda_2 \left[s_\theta(X_t, t, \varnothing, C_2) - s_\theta(X_t, t, \varnothing, \varnothing)\right]. \qquad (25)
\end{aligned}$$

In (24), it is assumed that $C_1 \perp\!\!\!\perp C_2|X_t$. However, as shown in Figure 3, C_1 and C_2 are not conditionally independent given X_t. The bias in the estimation of score function was also observed by Gaudi et al. (2025).

D.3 COIND

COIND is a principled and elegant method capable of logically decomposing and combining multiple conditions (Gaudi et al., 2025). This is achieved by adding an additional penalty term L_{penalty} to the training loss:

$$\begin{aligned}
&L_{\text{penalty}} \\
&\quad = \mathbb{E}_{X_t, C_1, C_2} \| s_\theta(X_t, t, \varnothing, \varnothing) + \left[s_\theta(X_t, t, C_1, \varnothing) - s_\theta(X_t, t, \varnothing, \varnothing)\right] \\
&\qquad\qquad + \left[s_\theta(X_t, t, \varnothing, C_2) - s_\theta(X_t, t, \varnothing, \varnothing)\right] - s_\theta(X_t, t, C_1, C_2) \|_2^2, \qquad (26)
\end{aligned}$$

which effectively enforces $s_\theta(X_t, t, C_1, C_2)$ to be consistent with the assumption that $C_1 \perp\!\!\!\perp C_2|X_t$, as shown in (24).

Although practical, such a penalty could introduce bias in s_θ, as shown in the left panel of Figure 3 and Figure 1. We argue that, as in DMDG and DMHG, directly injecting a C_1-aware guidance of C_2 provides a unbiased score estimation without violating the assumption $C_1 \not\perp\!\!\!\perp C_2|X_t$. In addition, COIND requires access to the complete dataset during training due to the appearance of $s_\theta(X_t, t, C_1, C_2)$. As a result, we are only able to include comparisons with COIND in the simulation study presented in Section 4, and cannot apply it to real-world datasets.

D.4 Conditioning pre-Trained diffusion models with Reinforcement Learning (CTRL)

Zhao et al. (2025b) proposed incorporating additional conditions into a pre-trained CDM via reinforcement learning. Their method involves fine-tuning a pre-trained neural network $nn_\theta\left(X_t, t, C_1\right)$. Let $nn_{\theta,\psi}(X_t, t, C_1, C_2)$ denote the neural network subject to fine-tuning, where θ represents the original parameters and ψ denotes the parameters for processing C_2, extended by Controlnet (Zhang et al., 2023). The loss function used to fine-tune the CDM consists of two components:

$$\int_{t_{\min}}^{t_{\max}} \frac{\| nn_\theta\left(X_t, t, C_1\right) - nn_{\theta,\psi}(X_t, t, C_1, C_2) \|_2^2}{2t^3} dt - \gamma \log p(C_2|X_{t_{\min}}, C_1) \qquad (27)$$

where γ is a fixed guidance scale. The first component is the weighted discrepancy between the trajectories of the original diffusion model and the fine-tuned diffusion model. The second component is the negative log-likelihood, or in terms of RL, the reward function. $X_{t_{\min}}$ is the generated sample at the terminate of the reverse process. By minimizing (27), the fine-tuned model is capable

of generating samples conditioned on C_1 and C_2. When C_1 and C_2 are conditionally independent given X_0, the negative log-likelihood term can be simplified to $-\gamma \log p(C_2|X_{t_{\min}})$.

Although this fine-tuning approach is indeed well-motivated, it still has several drawbacks. First, as a fine-tuning technique, it depends on a sufficiently strong pre-trained model. In our setting, this implies that one should first train $nn_\theta(X_t, t, C_1)$ on a $D^{(1)}$ with a huge sample size, then fine-tune $nn_{\theta,\psi}$ on $D^{(2)}$. In practice, such $D^{(1)}$ seldom exists. Second, the fine-tuning method was originally designed for datasets such as $D^{(2)} = \{X_0^{(2)}, C_1^{(2)}, C_2^{(2)}\}$ and the pair (C_1, C_2) explicitly appears in (27). In the case of (1), Zhao et al. (2025b) proposed using $C_1 \perp\!\!\!\perp C_2|X_0$ to eliminate C_1 in the negative log-likelihood, and uniformly sampling C_1 then coupling it with samples in $C_2^{(2)}$. In other words, (C_1, C_2) is drawn from $P_{C_1} \times P_{C_2}$ rather than P_{C_1,C_2}. Such replacement may undermine the fine-tuning of $nn_{\theta,\psi}(X_t, t, C_1, C_2)$ when miss-matched (C_1, C_2) appears. Furthermore, the reward function is the negative log-likelihood only at the end of the reverse process (sparsity), instead of the entire reverse process, which may impair the conditional control. Finally, the scale of negative log-likelihood γ, is predetermined during training, preventing flexible adjustment at generation time.

D.5 Imputation methods

We compared regressor imputation, GAIN (Yoon et al., 2018), Forest diffusion (Jolicoeur-Martineau et al., 2024) and KNN imputation. Regressor imputation is the simplest approach, where the missing $C_2^{(1)}$ and $C_1^{(2)}$ are replaced by $\widehat{f}_2(X_0^{(1)})$ and $\widehat{f}_1(X_0^{(2)})$. However, this replacement neglects the stochastic nature of $C_2^{(1)}$ and $C_1^{(2)}$ because $\widehat{f}_1$ and $\widehat{f}_2$ only estimate the conditional mean (see (1)). GAIN and Forest Diffusion are both generative-model-based imputation algorithms. The former leverages generative adversarial networks (GANs), while the latter is built upon diffusion models. Although both methods are capable of capturing the stochasticity of C_1 and C_2, their performance is limited due to the structure of block-wise missingness. In the experiments presented in Section 4, the best imputation method is KNN. Due to the block-wise missing structure, KNN depends solely on information from X_0, which aligns with the data generation mechanism described in (1). However, most imputation-based approaches still perform worse than guidance methods and CTRL.

E Additional Experimental Details and Results

E.1 Calculating W_2 distance and extra results of De novo drug design

In section 5.2, we adopt COATI, a powerful variational autoencoder (VAE), to encode molecules into a latent space and decode latent vectors back into molecules (Kaufman et al., 2024b). The diffusion models then operate within the latent space. To evaluate how well the generated molecules align with the target conditional distributions, we construct two reference sets. We first select 500 molecules from GEOM-DRUG whose properties C_1 are not only closest to the target C_1 (in terms of ℓ_2 distance), but also satisfy the requirements of Task 1. Similarly, we select 500 molecules from ZINC250k whose properties C_2 are closest to the target C_2 and satisfy the criteria of Task 2. We then compute the W_2 distance between the latent vectors of the generated molecules and those of the corresponding reference molecules. The results are shown in Figure 2.

We provide the detailed success rates for Task 1 and Task 2 (13) shown in Figure 2, paired with the corresponding guidance scales. The results are summarized in Table 6. It can be observed that, compared to DMIDG, DMDG achieves a higher task success rate with smaller guidance scales, which also is the key to its lower W_2 distance to the GEOM dataset.

E.2 Additional metrics for generated molecules of De novo drug design

In Table 7, we report several widely used metrics for evaluating generated molecules, including Validity, Novelty, and Diversity (Li et al., 2024; Shen et al., 2025; Hoogeboom et al., 2022; Gebauer et al., 2022). Validity refers to the proportion of generated molecules that are chemically valid. Novelty measures the proportion of generated molecules that are not in the training set. Diversity is

Table 6: Detailed success rates and corresponding guidance scales used for each method in Figure 2.

(a) DMDG

Task 1&2	Task 1	Task 2	λ_1	λ_2
0.1 %	0.9 %	49.9 %	0	0
12.6 %	26.1 %	52.6 %	6	3
21.3 %	34.6 %	68.3 %	9	6
30.5 %	51.5 %	63.8 %	12	6
39.7 %	78.0 %	54.5 %	21	6
50.5 %	81.2 %	64.6 %	27	9
60.7 %	89.5 %	68.3 %	42	12
70.4 %	93.2 %	75.5 %	60	18
76.0 %	93.4 %	81.5 %	63	21

(b) DMIDG

Task 1&2	Task 1	Task 2	λ_1	λ_2
0.1 %	0.9 %	49.9 %	0	0
10.5 %	12.5 %	98.5 %	9	27
21.1 %	21.5 %	97.7 %	18	33
30.4 %	31.5 %	96.6 %	24	27
40.6 %	43.2 %	95.0 %	33	30
50.1 %	55.2 %	92.4 %	51	33
55.2 %	61.7 %	91.9 %	57	30
58.7 %	66.2 %	90.6 %	63	30
N/A	N/A	N/A	N/A	N/A

(c) DMHG

Task 1&2	Task 1	Task 2	λ_1	λ_2
0.1 %	0.9 %	49.9 %	0	0
10.3 %	22.9 %	82.1 %	3	3
15.9 %	28.3 %	67.2 %	7	6
N/A	N/A	N/A	N/A	N/A

(d) DMIHG

Task 1&2	Task 1	Task 2	λ_1	λ_2
0.1 %	0.9 %	49.9 %	0	0
9.9 %	26.9 %	43.4 %	3	3
15.1 %	28.2 %	65.1 %	5	9
17.3 %	26.3 %	68.5 %	8	15

defined as follows:

$$\text{Diversity} = 1 - \frac{2}{n(n-1)} \sum_{\text{Mol}_1, \text{Mol}_2} \text{Sim}(\text{Mol}_1, \text{Mol}_2),$$

where Sim denotes the Tanimoto similarity, and Mol_1 and Mol_2 refer to two generated molecules in one batch.

Table 7: Every metric is evaluated on a batch with 1000 generated molecules, repeated by 5 times.

Method	**Guidance scales**	**Validity** (%, ↑)	**Uniqueness** (%, ↑)	**Diversity** (↑)
DMDG	$\lambda_1 = 63, \lambda_2 = 21$	99.440(0.351)	99.840(0.114)	0.851(0.001)
DMIDG	$\lambda_1 = 63, \lambda_2 = 30$	99.200(0.228)	98.660(0.571)	0.853(0.001)
DMHG	$\lambda_1 = 7, \lambda_2 = 6$	99.620(0.203)	95.780(1.907)	0.836(0.001)
DMIHG	$\lambda_1 = 8, \lambda_2 = 15$	99.680(0.193)	96.000(1.596)	0.844(0.001)
Uncond	$\lambda_1 = \lambda_2 = 0$	99.660(0.167)	99.620(0.228)	0.881(0.001)

E.3 Diffusion models trained on aggregated vs. single datasets

One of our core arguments is that, compared to training on a single dataset, leveraging aggregated datasets is expected to yield a better diffusion model. To validate this claim, we compare models trained solely on a single dataset $D^{(1)}$ with those trained on $D^{(1)} \cup D^{(2)}$ under Setting II in Section 4. When using only $D^{(1)}$, we assume the access to $\widehat{f}_2$.

The results can be found in Table 8. Diffusion models trained on $D^{(1)} \cup D^{(2)}$ exhibit consistently smaller W_2 distances than those trained on $D^{(1)}$, which supports our claim.

E.4 Aggregated datasets with varying sample sizes

In Section 5 and Section 4, the datasets $D^{(1)}$ and $D^{(2)}$ have equal or at least similar sample sizes. However, we are also interested in the effectiveness of proposed methods when the sample sizes of $D^{(1)}$ and $D^{(2)}$ are imbalanced. To this end, we continue to evaluate different methods with Setting

Table 8: W_2 distances and standard deviations of diffusion models trained on $D^{(1)}$ and $D^{(1)} \cup D^{(2)}$. All experiments are repeated 50 times and evaluated on 1000 samples.

Method	W_2 (↓) **on** $D^{(1)}$	W_2 (↓) **on** $D^{(1)} \cup D^{(2)}$
DMDG	4.506(5.113)	3.275(3.411)
DMIDG	6.787(8.211)	4.688(6.447)
DMHG	2.752(3.306)	2.386(2.923)
DMIHG	5.947(6.865)	4.919(6.415)

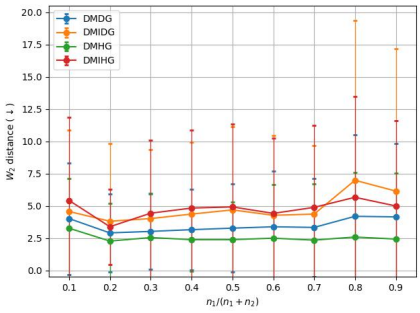

Figure 4: The W_2 distances and standard deviations of DMDG, DMIDG, DMHG and DMIHG when $n_1/(n_1+n_2)$ varies.

II in Section 4. Let n_1 and n_2 denote the sample sizes of $D^{(1)}$ and $D^{(2)}$, respectively. We fix the total number of samples to $n_1 + n_2 = 20000$, and vary the ratio $n_1/(n_1+n_2)$ from 0.1 to 0.9. We then evaluate each method by computing the W_2 distance between the generated samples and the true conditional distribution $P_{X_0|C_1,C_2}$.

The results are shown in Figure 4. Our proposed DMDG and DMHG consistently achieve the better performance and exhibit improved stability compared to DMIDG and DMIHG.

F Implementation Details

The network nn_θ in Section 4 was consistently implemented as a fully connected neural network with 3 hidden layers, each containing 128 neurons and using the SiLU activation function. The guidance networks $\widehat{f}_1$ and $\widehat{f}_2$ shared the same structure but with 2 hidden layers. The training, validation, and test sets are split in a ratio of 87% / 9% / 4%. When training both nn_θ and the $\widehat{f}_1$, $\widehat{f}_2$, we used the Adam optimizer with a learning rate of 1e-3. nn_θ was trained for up to 500 epochs, and the model with the lowest loss was selected. The numbers of training epochs for $\widehat{f}_1$ and $\widehat{f}_2$ were fixed at 50. The network of CTRL (Zhao et al., 2025b) is based on the pre-trained nn_θ and is fine-tuned using the ControlNet (Zhang et al., 2023). The optimizer of fine-tuning remains Adam, but the learning rate is set to 1e-4. Training is performed for up to 150 epochs, and the model with the lowest loss is selected. To compute the integral in (27), we first perform full sampling using EDM, with the number of sampling steps drawn from $\mathcal{U}\{18, 19, \ldots, 128\}$. In the experiments under Setting I, the value of γ is set to 180, while in Setting II, it is set to 1.5. When using GAIN (Yoon et al., 2018) and Forest Diffusion (Jolicoeur-Martineau et al., 2024) for imputation, we adopted the default hyperparameters from their Github repositories, except that the hidden layers of the GAIN network were doubled in width to enhance its fitting capacity.

For molecular generation in Section 5, We adopted the same architecture as COATI-LDM (Kaufman et al., 2024a). Specifically, the U-Net is built using 2-layer weight-normalized SwiGLU layers, and performs three stages of downsampling followed by three stages of upsampling. Additionally, C_1 and C_2 were encoded using sine-cosine embeddings. The training, validation, and test sets are split in a ratio of 76% / 14% / 10%, same with (Hoogeboom et al., 2022). The probability of masking C_1, p_{non} in (4) was set to $0.5 \cdot n_1/(n_1+n_2)$. We used the AdamW optimizer with a learning rate of 1e-3 and a weight decay of 1e-4. The learning rate was scheduled using cosine annealing

with a minimum value of 1e-5. The model was trained for 2400 epochs, which took approximately 20 hours on a single NVIDIA A100. The guidance networks $\widehat{f}_1$ and $\widehat{f}_2$ are variants of the U-Net architecture, with only four downsampling stages and outputting predictions in the final layer. The optimizer is Adam with a learning rate of 1e-3. The maximum epoch was set to 50, and the model with the lowest loss was selected. During CTRL fine-tuning, we used the AdamW optimizer with a learning rate of 5e-4, and set γ to 500. The maximum number of training epochs was set to 1000, and the model with the lowest loss was selected. When applying GAIN to the aggregated GEOM and ZINC250k datasets for imputation, we increased the width of the hidden layers and replaced the original fully connected network with a 6-layer residual neural network. The optimizer used was Adam with a learning rate of 1e-3. For Forest Diffusion, due to the large size of the input dataset, we set the duplication factor to 1 and used LightGBM (LGB) as the base model, with all other parameters kept at their default values.

Unless otherwise specified, all diffusion models discussed in this paper refer to EDM and the sampling steps of the reverse process are set to 18.

G Computational Efficiency

Thanks to the simplicity of the EDM framework, we are able to obtain high-quality samples using only a small number of sampling steps (Karras et al., 2022). Unless otherwise specified, the number of sampling steps is set to 18, which brings significant advantages in computational efficiency. In Figure 5, we report the sampling speed of our model and COATI-LDM under varying batch sizes, along with the decoding time of COATI. For visualization purposes, we set the number of sampling steps of COATI-LDM to 100 and 200, while its default sampling steps is actually 1000. All measurements were conducted on an NVIDIA A100 GPU. As observed, when the sampling step is set to 18, DMDG achieves the fastest sampling speed, and the runtime remains relatively stable across different batch sizes. In contrast, decoding latent vectors using COATI is highly sensitive to the batch size. Larger batches significantly slow down the decoding process. This suggests that a smaller batch size should be preferred when decoding with COATI.

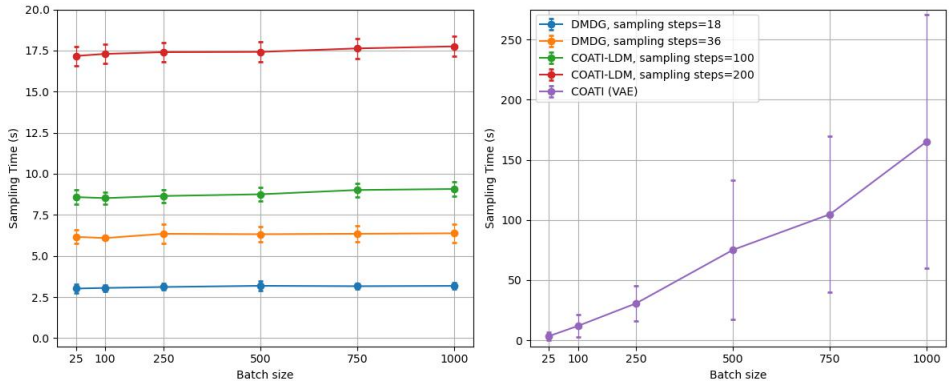

Figure 5: Sampling time comparison of De novo drug design under varying batch sizes. The left panel shows the sampling time of EDM with different sampling steps. The right panel displays the decoding time of COATI across the same batch sizes.

H Selections of Generated Molecules and Images

Figure 6 showcases a selection of molecules that satisfy Task 1 and Task 2 (13) in Section 5.2.

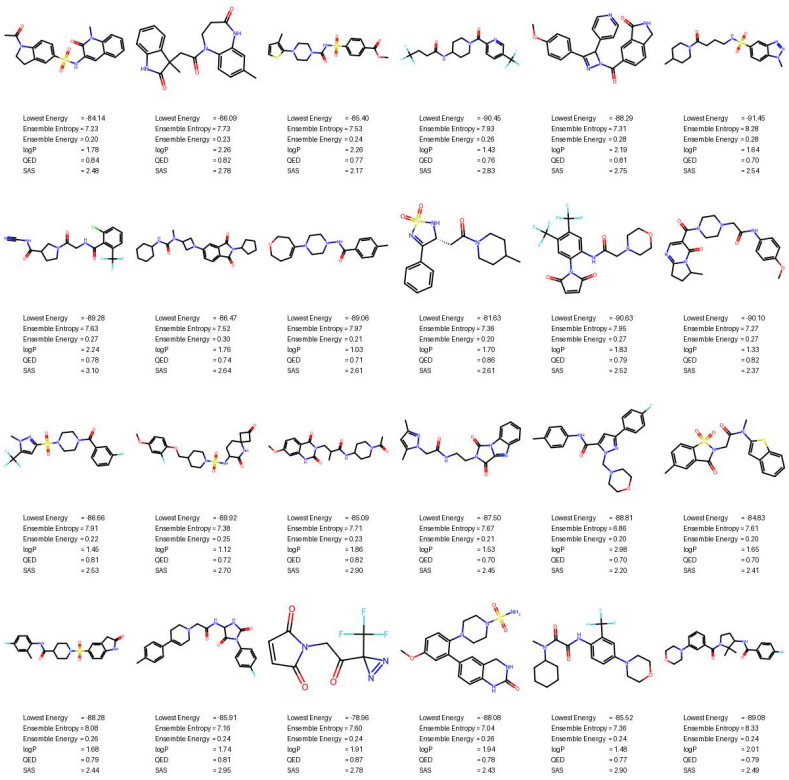

Figure 6: Selected molecules generated by DMDG that satisfy Tasks 1 and 2 (13). The guidance scale is $\lambda_1 = 63,\ \lambda_2 = 21$.