Multi-Bridge Denoising Diffusion Probabilistic Models

Mélodie Monod*
CEREMADE, Université Paris-Dauphine - PSL
CNRS
University of Oxford

Habib Ganjgahi
University of Oxford

Abstract

Image-to-image translation aims to generate a target image conditioned on an observed source image. While diffusion bridge models perform well in one-to-one settings, they are not adapted to problems where multiple sources provide complementary information about a single target. We introduce *Multi-Bridge Denoising Diffusion Probabilistic Models* (MB-DDPMs), a framework for many-to-one translation that builds stochastic bridges between the target and each source, coupled with a unified reverse process that aggregates information across sources. We evaluate MB-DDPM on brain MRI for multiple sclerosis to predict FLAIR contrast from T1, T2 and PD contrasts. We demonstrate strong clinical relevance through near-perfect agreement with ground-truth lesion-based measurements, highlighting its potential for reliable downstream medical analysis. We further evaluate MB-DDPM on image restoration using CIFAR-10 and ImageNet. Across all datasets, MB-DDPM achieves the highest perceptual quality while maintaining reconstruction fidelity that is competitive with, or superior to, that of the strongest baselines.

1 Introduction

Image-to-image translation seeks to generate a plausible target image conditioned on an observed source image. It is a fundamental problem in computer vision, with applications ranging from optimal filtering [21] and data compression [35] to safety-critical domains such as medicine and robotics [28, 14]. Common image-to-image translation tasks are inherently ill-posed [3, 23], as multiple valid targets may correspond to a single source. Consequently, modern approaches rely on conditional generative modeling to learn the distribution of the target image given the observed source image.

Diffusion models have recently emerged as a powerful class of generative models, achieving state-of-the-art results in synthesizing high-fidelity data [12, 29]. Diffusion models define a forward process that gradually corrupts data with noise, and a learned reverse process that iteratively denoises samples. Parameterized by neural networks trained to predict the noise of the forward process (or equivalently the score function), the reverse process enables high-quality generation from pure noise.

However, standard diffusion models are not naturally suited for image-to-image translation, as they assume a fixed prior (typically Gaussian noise) and learn a mapping from noise to data. Extending them to translation tasks therefore requires indirect strategies such as conditioning [37, 4] or modifying the sampling procedure [6, 29, 20]. These approaches often lack a unified theoretical grounding and typically enforce a directional mapping, limiting desirable properties such as cycle consistency [40].

An alternative approach is to learn a transport map between source and target distributions. Flow-matching ODE methods [15, 1, 17] have not matched the empirical success of diffusion models and have rarely been explored for image translation. Schrödinger Bridge methods [5] provide a principled

*Corresponding author (monod@ceremade.dauphine.fr)

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

optimal transport formulation but are computationally expensive. Diffusion bridge models offer a scalable alternative via stochastic bridges between endpoint distributions [26, 39].

Despite their success, existing diffusion bridge models are restricted to one-to-one mappings. In contrast, many scientific applications are inherently many-to-one, where multiple source observations provide complementary information about a single target. For example, in MRI, multiple source contrasts provide complementary information to predict a target contrast.

To address this limitation, we introduce *Multi-Bridge Denoising Diffusion Probabilistic Models* (MB-DDPMs, Figure 1), a principled extension of diffusion bridge models to many-to-one settings. Our approach constructs independent stochastic bridges between the target image and each source image, and couples them through a joint reverse process that aggregates information while preserving tractable intermediate distributions. To the best of our knowledge, this is the first diffusion-based framework to model stochastic bridges for many-to-one image translation.

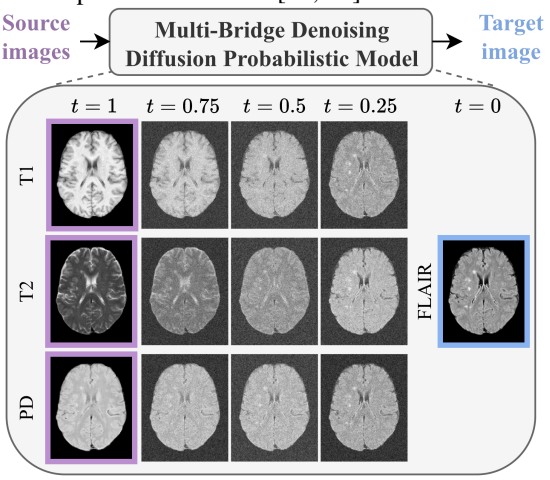


Figure 1: Illustration of **MB-DDPM**. Unlike standard diffusion bridge models that learn a bridge between a single source and target distribution, MB-DDPM constructs multiple stochastic bridges from several source distributions to the target, enabling many-to-one image translation.

We evaluate MB-DDPM on brain MRI scans from patients with multiple sclerosis (MS), using T1-, T2-, and proton density (PD)-weighted contrasts images as sources to predict FLAIR contrast (Figure 1). FLAIR is a key imaging modality for MS assessment as it highlights T2-hyperintense lesions, which correspond to areas of inflammatory damage in white matter, but its acquisition is often time-consuming and susceptible to motion artifacts. We demonstrate that MB-DDPM predicts high-quality FLAIR, achieving near-perfect agreement in lesion volume estimation. This suggests that predicted FLAIR preserves clinically relevant information for lesion detection required for accurate detection and quantification of disease burden [30, 27]. To assess its generality beyond medical imaging, we further evaluate MB-DDPM on two standard natural-image benchmarks, CIFAR-10 [13] and ImageNet [8], performing image restoration from multiple degraded observations. The code to reproduce our experiments is available at `https://github.com/NeuroSML/mb-ddpm`.

We observe consistent and predictable multi-source behavior: incorporating all sources outperforms any single source, while increased disagreement between sources leads to higher predictive error. Our method consistently outperforms existing approaches, including diffusion models for inverse problems [6, 4], conditional diffusion models [37], and one-to-one diffusion bridge methods [16].

2 Background

This section reviews DDPM-based diffusion models and diffusion bridge models with fixed endpoints; details are provided in Appendices A and B, respectively.

2.1 Generative Modeling with Denoising Diffusion Probabilistic Models

Let $\mathbf{x}_0 \in \mathbb{R}^d$ denote a data sample drawn from an unknown distribution,

$$\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0). \tag{1}$$

The goal of generative modeling is to learn a stochastic mechanism that allows sampling from the data distribution $p_{\text{data}}(\mathbf{x}_0)$. DDPMs [12] approach this problem by learning to reverse a fixed Markovian noising process that progressively adds Gaussian noise over T timesteps according to a variance schedule $\beta_{1:T} \in (0, 1)$. The joint distribution of the Forward Diffusion Process (FDP) is defined as

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}\big(\mathbf{x}_t; \sqrt{\alpha_t}\,\mathbf{x}_{t-1}, (1-\alpha_t)\,\mathbf{I}_d\big), \tag{2}$$

where $\alpha_t := 1 - \beta_t$. Define the cumulative product $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. The marginal distribution at timestep $t = 1, \ldots, T$ admits the closed form

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\,\mathbf{I}_d\big) \quad \Rightarrow \mathbf{x}_t \mid \mathbf{x}_0 = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{3}$$

Generating new samples can be done by reversing the FDP and solving the Reverse Diffusion Process (RDP). The joint distribution of the RDP is defined as a reverse-time Markov chain $p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$, starting at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}_d)$. Training is performed by optimizing the variational bound on the negative log-likelihood $p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$. In particular, this reduces to minimizing the divergence between the learned RDP transitions $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ and the forward posterior transition $q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)$ defined as

$$\begin{gathered} q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) = \mathcal{N}\big(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{t-1|0,t}(\mathbf{x}_0, \mathbf{x}_t), \sigma^2_{t-1|0,t}\,\mathbf{I}_d\big), \qquad t = 2, \ldots, T, \\ \text{where } \boldsymbol{\mu}_{t-1|0,t}(\mathbf{x}_0, \mathbf{x}_t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \text{ and } \sigma^2_{t-1|0,t} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \end{gathered} \tag{4}$$

Each RDP transition is designed to match the forward posterior transition in covariance, while its mean is obtained by replacing $\mathbf{x}_0$ with $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$.

Each RDP transition is designed to match the forward posterior transition, with its mean obtained by replacing $\mathbf{x}_0$ with $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$. The estimator $\hat{\mathbf{x}}_0(\mathbf{x}_t, t)$ is defined using Equation (3): $\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \big(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\big/\sqrt{\bar{\alpha}_t}$, where $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ estimates $\boldsymbol{\epsilon}$ using a time-conditional neural network parameterized by θ. The noise-prediction objective of the neural network is:

$$\mathcal{J}_{\mathrm{DDPM}}(\theta) = \mathbb{E}_{t \sim \mathrm{Unif}\{1,\ldots,T\}, \mathbf{x}_0 \sim p_{\mathrm{data}}(\mathbf{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\big(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\big) \right\|^2 \right]. \tag{5}$$

It can be shown that minimizing the noise-prediction objective (5) for a fixed $t = 1, \ldots, T$ yields an estimator $\hat{\mathbf{x}}_0^\star(\mathbf{x}_t, t)$ that achieves the minimum of the variational bound on the negative log-likelihood, by recovering the posterior mean $\hat{\mathbf{x}}_0^\star(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$.

2.2 Diffusion Process with Fixed Endpoints

Standard diffusion models are limited to transporting complex data distributions to a standard Gaussian. Instead, consider the endpoints of a diffusion $(\mathbf{x}_0, \mathbf{x}_T) \in \mathbb{R}^d \times \mathbb{R}^d$, drawn from an unknown joint data distribution

$$(\mathbf{x}_0, \mathbf{x}_T) \sim p_{\mathrm{data}}(\mathbf{x}_0, \mathbf{x}_T). \tag{6}$$

When the endpoints points $\mathbf{x}_0$ and $\mathbf{x}_T$ are fixed, such a process is known as a diffusion bridge [31, 10, 16, 39]. The goal of a diffusion bridge is to learn a stochastic mechanism that allows sampling from the conditional distribution $p_{\mathrm{data}}(\mathbf{x}_0 \mid \mathbf{x}_T) \propto p_{\mathrm{data}}(\mathbf{x}_0, \mathbf{x}_T)$.

To simplify notation, we define the reverse cumulative product $\tilde{\alpha}_t := \prod_{s=t+1}^{T} \alpha_s = \bar{\alpha}_T / \bar{\alpha}_t$. Diffusion bridges define a variant of the FDP conditioned on a terminal state $\mathbf{x}_T$. Intuitively, this defines a probabilistic path starting at $\mathbf{x}_0$ and constrained to terminate at $\mathbf{x}_T$. The joint distribution of the bridge FDP, conditioned on both endpoints, factorizes according to the Markov property as

$$\begin{gathered} q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T) := \prod_{t=1}^{T-1} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T),\ q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T) := \mathcal{N}\Big(\mathbf{x}_t;\, \boldsymbol{\mu}_{t|t-1,T}(\mathbf{x}_{t-1}, \mathbf{x}_T),\, \sigma^2_{t|t-1,T}\,\mathbf{I}_d\Big), \\ \text{where } \boldsymbol{\mu}_{t|t-1,T}(\mathbf{x}_{t-1}, \mathbf{x}_T) = \frac{\sqrt{\alpha_t}(1-\tilde{\alpha}_t)}{1-\tilde{\alpha}_{t-1}}\mathbf{x}_{t-1} + \frac{\sqrt{\tilde{\alpha}_t}\beta_t}{1-\tilde{\alpha}_{t-1}}\mathbf{x}_T, \text{and } \sigma^2_{t|t-1,T} = \frac{(1-\tilde{\alpha}_t)\beta_t}{1-\tilde{\alpha}_{t-1}}. \end{gathered} \tag{7}$$

The marginal distribution at timestep $t = 1, \ldots, T-1$ admits the closed form [16, 39]

$$q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) = \mathcal{N}\Big(\mathbf{x}_t;\, \boldsymbol{\mu}_{t|0,T}(\mathbf{x}_0, \mathbf{x}_T),\, \sigma^2_{t|0,T}\,\mathbf{I}_d\Big), \tag{8}$$

$$\text{where } \boldsymbol{\mu}_{t|0,T}(\mathbf{x}_0, \mathbf{x}_T) = \underbrace{\frac{\sqrt{\bar{\alpha}_t}(1-\tilde{\alpha}_t)}{1-\bar{\alpha}_T}}_{C^{(0)}_{t|0,T}}\mathbf{x}_0 + \underbrace{\frac{\sqrt{\tilde{\alpha}_t}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_T}}_{C^{(T)}_{t|0,T}}\mathbf{x}_T, \text{ and } \sigma^2_{t|0,T} = \frac{(1-\bar{\alpha}_t)(1-\tilde{\alpha}_t)}{1-\bar{\alpha}_T}.$$

Generating new samples can be done by reversing the bridge FDP and solving the bridge RDP starting from $\mathbf{x}_T$. This sampling procedure is adopted in I^{2}SB [16], although a rigorous justification of the associated RDP has not been established.

3 Multi-Bridge Denoising Diffusion Probabilistic Models

Assume K paired random variables $(\mathbf{x}_0, \mathbf{x}_T^k) \in \mathbb{R}^d \times \mathbb{R}^d$, for $k = 1, \ldots, K$, where $\mathbf{x}_0 \in \mathbb{R}^d$ denotes a target observation and $\mathbf{x}_T^k \in \mathbb{R}^d$ denotes the k-th source observation. We assume that the target observation is drawn from an unknown data distribution and that, conditional on $\mathbf{x}_0$, each source observation is generated independently

$$\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0), \qquad \mathbf{x}_T^k \mid \mathbf{x}_0 \sim p_{\text{data}}^k(\mathbf{x}_T^k \mid \mathbf{x}_0), \quad k = 1, \ldots, K. \tag{9}$$

Let $\mathbf{x}_T^{1:K} := \{\mathbf{x}_T^k\}_{k=1}^K$ denote the collection of the K source observations. The resulting joint data distribution over $(\mathbf{x}_0, \mathbf{x}_T^{1:K})$ factorizes as

$$p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K}) = p_{\text{data}}(\mathbf{x}_0) \prod_{k=1}^K p_{\text{data}}^k(\mathbf{x}_T^k \mid \mathbf{x}_0). \tag{10}$$

The objective of MB-DDPMs is to construct a stochastic procedure that approximately samples from the conditional distribution

$$p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{x}_T^{1:K}) \propto p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K}), \tag{11}$$

thereby recovering the target $\mathbf{x}_0$ from the set of sources $\mathbf{x}_T^{1:K}$. We assume access to a training dataset $\{(\mathbf{x}_0, \mathbf{x}_T^{1:K})^{(i)}\}_{i=1}^{n_{\text{train}}} \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K})$, and a test set containing only source observations $\{(\mathbf{x}_T^{1:K})^{(i)}\}_{i=1}^{n_{\text{test}}}$, sampled from the corresponding marginal distribution.

3.1 Forward Diffusion Process

We construct K independent bridge FDPs, each starting from the common target $\mathbf{x}_0$ and terminating at a distinct source $\mathbf{x}_T^k$. Let $\mathbf{x}_{1:(T-1)}^k := \{\mathbf{x}_t^k\}_{t=1}^{T-1}$ denote the intermediate trajectory of the k-th diffusion bridge. For each $k = 1, \ldots, K$, the joint distribution of the intermediate states is given by $q(\mathbf{x}_{1:(T-1)}^k \mid \mathbf{x}_0, \mathbf{x}_T^k)$, as defined in (7).

To simplify notation, define the stacked variables $\mathbf{x}_t^{1:K} := \{\mathbf{x}_t^k\}_{k=1}^K$ and $\mathbf{x}_{1:(T-1)}^{1:K} := \{\mathbf{x}_{1:(T-1)}^k\}_{k=1}^K$. Exploiting the independence of the K bridge FDPs, the multi-bridge FDP defines a Markov chain over the stacked trajectories, with joint distribution given by

$$\begin{aligned}
q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) &= q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) \prod_{t=2}^{T-1} q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_{t-1}^{1:K}, \mathbf{x}_T^{1:K}), \\
q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) &= \mathcal{N}\Big(\mathbf{x}_1^{1:K};\, \boldsymbol{\mu}_{1|0,T}(\mathbf{A}\mathbf{x}_0, \mathbf{x}_T^{1:K}),\, \sigma_{1|0,T}^2\, \mathbf{I}_{dK}\Big), \\
q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_{t-1}^{1:K}, \mathbf{x}_T^{1:K}) &= \mathcal{N}\Big(\mathbf{x}_t^{1:K};\, \boldsymbol{\mu}_{t|t-1,T}(\mathbf{x}_{t-1}^{1:K}, \mathbf{x}_T^{1:K}),\, \sigma_{t|t-1,T}^2\, \mathbf{I}_{dK}\Big),
\end{aligned} \tag{12}$$

where $\mathbf{A} = [\mathbf{I}_d, \ldots, \mathbf{I}_d]^\top \in \mathbb{R}^{dK \times d}$ is the replication matrix, and with $\boldsymbol{\mu}_{t|t-1,T}$ and $\sigma_{t|t-1,T}^2$ in (7). The marginal distribution at timestep follows directly from the independence of the K bridge FDPs:

$$q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) = \mathcal{N}\Big(\mathbf{x}_t^{1:K}; \boldsymbol{\mu}_{t|0,T}(\mathbf{A}\mathbf{x}_0, \mathbf{x}_T^{1:K}), \sigma_{t|0,T}^2\, \mathbf{I}_{dK}\Big), \quad t = 1, \ldots, T-1, \tag{13}$$

with $\boldsymbol{\mu}_{t|0,T}$ and $\sigma_{t|0,T}^2$ in (8). Equivalently, the marginal admits the Gaussian decomposition

$$\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K} = C_{t|0,T}^{(0)} \mathbf{A}\mathbf{x}_0 + C_{t|0,T}^{(T)}\, \mathbf{x}_T^{1:K} + \sigma_{t|0,T}\, \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dK}), \tag{14}$$

where $C_{t|0,T}^{(0)}$ and $C_{t|0,T}^{(T)}$ are the coefficients defining $\boldsymbol{\mu}_{t|0,T}$ as given in (8).

3.2 Reverse Diffusion Process

We define the joint distribution of the multi-bridge RDP as a reverse-time Markov chain with learned transitions, starting from $\mathbf{x}_T^{1:K}$. Because the multi-bridge RDP is not conditioned on $\mathbf{x}_0$, the resulting distribution does not factorize across sources. Its general form is:

$$p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K}) = p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K}) \left(\prod_{t=2}^{T-1} p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}) \right) p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K}). \tag{15}$$

The MB-DDPM likelihood $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T^{1:K})$ is defined as the marginal of the multi-bridge RDP:

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T^{1:K}) = \int p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K}) \, \mathrm{d}\mathbf{x}_{1:(T-1)}^{1:K}. \tag{16}$$

Our goal is to draw samples from $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T^{1:K})$. Since direct sampling from this distribution is intractable, we instead sample by simulating the reverse-time chain, starting from a source samples $\mathbf{x}_T^{1:K}$ and iteratively sampling

$$\mathbf{x}_{T-1}^{1:K} \sim p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K}), \quad \left\{\mathbf{x}_{t-1}^{1:K} \sim p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})\right\}_{t=T-1}^{2}, \quad \mathbf{x}_0 \sim p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K}). \tag{17}$$

3.3 Variational Bound

We define the forward posterior over the stacked variables. For each $k = 1, \ldots, K$ and $t = 2, \ldots, T$, let $q(\mathbf{x}_{t-1}^k \mid \mathbf{x}_0, \mathbf{x}_t^k)$ denote the forward posterior given in (4). Conditioned on $\mathbf{x}_0$, these K distributions are independent. Hence, for $t = 2, \ldots, T$, the multi-bridge forward posterior factorizes as

$$q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K}) = \mathcal{N}\Big(\mathbf{x}_{t-1}^{1:K};\, \boldsymbol{\mu}_{t-1|0,t}(\mathbf{A}\mathbf{x}_0, \mathbf{x}_t^{1:K}),\, \sigma_{t-1|0,t}^2 \, \mathbf{I}_{dK}\Big), \tag{18}$$

with $\boldsymbol{\mu}_{t-1|0,t}$ and $\sigma_{t-1|0,t}^2$ defined in (4).

Training MB-DDPM is performed by optimizing a variational lower bound on the negative MB-DDPM log-likelihood (16).

Proposition 3.1 (Variational Lower Bound for Multi-Bridge Denoising Diffusion Probabilistic Models)**.** *Let $q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})$ denote the multi-bridge FDP* (12)*, and $p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K})$ the corresponding multi-bridge RDP* (15)*. For $t = 2, \ldots, T$, let $q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K})$ denote the multi-bridge forward posterior* (18)*.*

Then, the variational lower bound on the negative log-likelihood (16) *is given by*

$$\begin{aligned} -\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T^{1:K}) \le{}& D_{\mathrm{KL}}\left(q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) \,\|\, p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})\right) \\ &+ \sum_{t=2}^{T-1} \mathbb{E}_{q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\left[D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K}) \,\|\, p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})\right)\right] \\ &+ \mathbb{E}_{q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})\Big]. \end{aligned} \tag{19}$$

The proof of Proposition 3.1 is provided in Appendix F.4.1.

3.4 Parameterization of the Reverse Diffusion Transitions

We introduce a parameterized estimator $\mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) : \mathbb{R}^{dK} \times \mathbb{R}^{dK} \times \mathbb{R} \to \mathbb{R}^d$ for the unknown target $\mathbf{x}_0$. The multi-bridge RDP transitions are parameterized in terms of $\mathbf{m}_\theta$ so as to minimize the variational lower bound (19).

Specifically, each multi-bridge RDP transition is designed to match the forward posterior transition (18), with its mean obtained by replacing $\mathbf{x}_0$ with $\mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)$.

In the multi-bridge FDP, each forward state $\mathbf{x}_t^k$ evolves independently conditioned on the shared target $\mathbf{x}_0$. In contrast, the multi-bridge RDP exhibits a weaker form of independence. Each reverse state $\mathbf{x}_{t-1}^k$ is conditionally independent of other bridges at the same timestep, but it depends on the full set of next-step states $\mathbf{x}_t^{1:K}$ and sources $\mathbf{x}_T^{1:K}$ through the estimator $\mathbf{m}_\theta$ (see Figure 2).

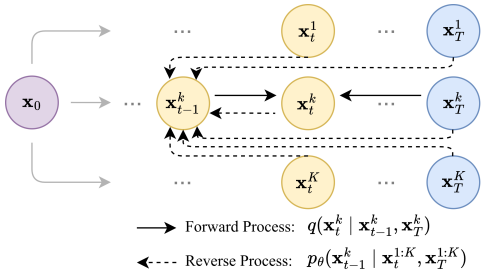


Figure 2: Forward and Reverse Diffusion Processes of MB-DDPM.

The first reverse step corresponds to sampling $\mathbf{x}_{T-1}^{1:K}$ from $\mathbf{x}_T^{1:K}$. Since $\mathbf{x}_{T-1}^{1:K}$ is not yet available, the estimator cannot directly depend on it. However, in the multi-bridge FDP the conditional variance

at $t = T-1$ is very small ($\sigma^2_{T-1|0,T} \ll 1$), implying that $\mathbf{x}^{1:K}_{T-1}$ concentrates around $\mathbf{x}^{1:K}_T$. We therefore approximate $\mathbf{x}^{1:K}_{T-1} \approx \mathbf{x}^{1:K}_T$ when evaluating the estimator, leading to:

$$p_\theta(\mathbf{x}^{1:K}_{T-1} \mid \mathbf{x}^{1:K}_T) = \mathcal{N}\Big(\mathbf{x}^{1:K}_{T-1}; \boldsymbol{\mu}_{T-1|0,T}\big(\mathbf{A}\mathbf{m}_\theta(\mathbf{x}^{1:K}_T, \mathbf{x}^{1:K}_T, T-1), \mathbf{x}^{1:K}_T\big), \sigma^2_{T-1|0,T}\,\mathbf{I}_{dK}\Big). \quad (20)$$

For the timesteps $t = T-1, \dots, 2$, the reverse transitions follow directly from the forward posterior:

$$p_\theta(\mathbf{x}^{1:K}_{t-1} \mid \mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T) = \mathcal{N}\Big(\mathbf{x}^{1:K}_{t-1}; \boldsymbol{\mu}_{t-1|0,t}\left(\mathbf{A}\mathbf{m}_\theta\left(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t\right), \mathbf{x}^{1:K}_t\right), \sigma^2_{t-1|0,t}\,\mathbf{I}_{dK}\Big). \quad (21)$$

For $t = 1$, the forward posterior collapses to a Dirac delta, so we replace it with a Gaussian of small variance $\sigma^2_{\text{rec}} \ll 1$:

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}^{1:K}_1, \mathbf{x}^{1:K}_T) = \mathcal{N}\big(\mathbf{x}_0; \mathbf{m}_\theta(\mathbf{x}^{1:K}_1, \mathbf{x}^{1:K}_T, 1), \sigma^2_{\text{rec}}\,\mathbf{I}_d\big). \quad (22)$$

3.5 Learning an Estimator $\mathbf{m}_\theta$ for $\mathbf{x}_0$

Conditional Estimator. Let $\boldsymbol{\epsilon}_{\theta_{\text{cond}}}(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t)$ denote a neural network predicting the Gaussian noise in the multi-bridge FDP marginal (14) that depends on parameters θ_{cond}. The conditional estimator of $\mathbf{x}_0$ is obtained by inverting the linear Gaussian decomposition (as in standard DDPMs):

$$\hat{\mathbf{x}}^{\text{cond}}_0(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t) := \frac{\sum_{k=1}^K \left(\mathbf{x}^k_t - C^{(T)}_{t|0,T}\mathbf{x}^k_T\right) - \sigma_{t|0,T}\,\mathbf{A}^\top \boldsymbol{\epsilon}_{\theta_{\text{cond}}}(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t)}{K C^{(0)}_{t|0,T}}, \quad (23)$$

where we note that $\mathbf{A}^\top\mathbf{A} = K\,\mathbf{I}_d$, $\mathbf{A}^\top\mathbf{x}^{1:K}_t = \sum_{k=1}^K \mathbf{x}^{1:K}_t$ and $\mathbf{A}^\top\mathbf{x}^{1:K}_T = \sum_{k=1}^K \mathbf{x}^{1:K}_T$. The corresponding noise-prediction objective, averaging over timesteps t and data samples $(\mathbf{x}_0, \mathbf{x}^{1:K}_T)$, is:

$$\begin{aligned}
\mathcal{J}_{\text{cond}}(\theta_{\text{cond}}) &= \mathbb{E}_{t\sim\text{Unif}\{1,\dots,T-1\},\,(\mathbf{x}_0,\mathbf{x}^{1:K}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}^{1:K}_T),\,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_{dK})}\left[\mathcal{L}_{\text{cond}}(\theta_{\text{cond}}, \mathbf{x}_0, \mathbf{x}^{1:K}_T, \boldsymbol{\epsilon}, t)\right] \\
\mathcal{L}_{\text{cond}}(\theta_{\text{cond}}, \mathbf{x}_0, \mathbf{x}^{1:K}_T, \boldsymbol{\epsilon}, t) &= \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}\left(C^{(0)}_{t|0,T}\mathbf{A}\mathbf{x}_0 + C^{(T)}_{t|0,T}\mathbf{x}^{1:K}_T + \sigma_{t|0,T}\boldsymbol{\epsilon}, \mathbf{x}^{1:K}_T, t\right)\right\|^2.
\end{aligned} \quad (24)$$

The following result shows that the noise-prediction objective (24) yields the optimal estimator minimizing the variational bound (19).

Proposition 3.2 (Optimality of the Conditional Estimator)**.** *For fixed $t = 1, \dots, T-1$, let $\boldsymbol{\epsilon}^*_{\theta_{\text{cond}}}(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t)$ denote the minimizer of* (24)*. Then, the corresponding estimator $\hat{\mathbf{x}}^{\text{cond},*}_0(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t)$ defined in* (23)*, parametrized by $\boldsymbol{\epsilon}^*_{\theta_{\text{cond}}}(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t)$, satisfies*

$$\hat{\mathbf{x}}^{\text{cond},*}_0(\mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}^{1:K}_t, \mathbf{x}^{1:K}_T], \quad t = 1, \dots, T-1, \quad (25)$$

and thus minimizes the expected variational bound (19)*.*

The proof of Proposition 3.2 is deferred to Appendix F.4.2.

Although statistically optimal, the conditional estimator becomes numerically unstable as $t \to T$. In this regime, $C^{(0)}_{t|0,T} \to 0$, $\sigma^2_{t|0,T} \to 0$, and $\mathbf{x}^k_t \approx \mathbf{x}^k_T$, so that the contribution of $\mathbf{x}_0$ vanishes (see Figure 3). As a result, the signal-to-noise (SNR) ratio,

$$\text{SNR}^{\text{cond}}_t = \frac{(C^{(0)}_{t|0,T})^2}{\sigma^2_{t|0,T}}, \quad (26)$$

degenerates near the final endpoint (see Figure 4).

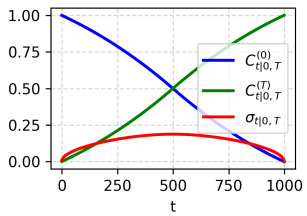


Figure 3: Forward diffusion process coefficients.

Marginal Estimator. To address the numerical instability of the conditional estimator, we introduce a marginal estimator that does not explicitly condition on $\mathbf{x}^{1:K}_T$. We express $\mathbf{x}^{1:K}_t$ in the DDPM-style form

$$\mathbf{x}^{1:K}_t = \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\eta}_t, \quad (27)$$

where $\boldsymbol{\eta}_t$ is a residual variable whose distribution depends on the multi-bridge construction. Let $\boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t)$ denote a neural network trained to predict this residual and that depends on parameters θ_{marg}. The marginal estimator is obtained by inverting (27):

$$\hat{\mathbf{x}}_0^{\text{marg}}(\mathbf{x}_t^{1:K}, t) = \frac{\sum_{k=1}^{K} \mathbf{x}_t^k - \sqrt{1-\bar{\alpha}_t}\mathbf{A}^\top \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t)}{K\sqrt{\bar{\alpha}_t}}. \quad (28)$$

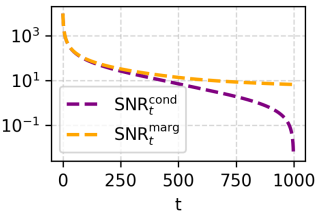


Figure 4: Signal-to-noise ratio (log-scale) for the conditional and marginal estimators.

The marginal estimator has a smooth and finite SNR for all t, $\text{SNR}_t^{\text{marg}} = \bar{\alpha}_t/(1-\bar{\alpha}_t)$. This ensures stable learning even near $t \approx T$ (see Figure 4). The network $\boldsymbol{\eta}_{\theta_{\text{marg}}}$ is trained with the objective

$$\mathcal{J}_{\text{marg}}(\theta_{\text{marg}}) = \mathbb{E}_{t\sim\text{Unif}\{1,\dots,T-1\},\,(\mathbf{x}_0,\mathbf{x}_T^{1:K})\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T^{1:K}),\,\mathbf{x}_t^{1:K}\sim q(\mathbf{x}_t^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\Big[\mathcal{L}_{\text{marg}}(\theta_{\text{marg}}, \mathbf{x}_0, \mathbf{x}_t^{1:K}, t)\Big],$$
$$\mathcal{L}_{\text{marg}}(\theta_{\text{marg}}, \mathbf{x}_0, \mathbf{x}_t^{1:K}, t) = \left\| \frac{\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t) \right\|^2. \quad (29)$$

The loss in (29) does not condition on $\mathbf{x}_T^{1:K}$, so the marginal estimator averages over the possible terminal states $\mathbf{x}_T^{1:K}$ and thus recovers the marginal posterior mean as shown in the next proposition.

Proposition 3.3 (Marginal Posterior Mean Estimator)**.** *For fixed $t = 1, \dots, T-1$, let $\boldsymbol{\eta}^*_{\theta_{marg}}(\mathbf{x}_t^{1:K}, t)$ denote the minimizer of* (29)*. Then, the corresponding estimator $\hat{\mathbf{x}}_0^{marg,\star}(\mathbf{x}_t^{1:K}, t)$ defined in* (28)*, parametrized by $\boldsymbol{\eta}^*_{\theta_{marg}}(\mathbf{x}_t^{1:K}, t)$, recovers the marginal posterior mean:*

$$\hat{\mathbf{x}}_0^{marg,\star}(\mathbf{x}_t^{1:K}, t) = \mathbb{E}_{marg}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}], \quad t = 1, \dots, T-1. \quad (30)$$

The proof of Proposition 3.3 is deferred to Appendix F.4.3.

Hybrid Estimator. To combine the statistical efficiency of the conditional estimator with the numerical stability of the marginal estimator, we implement $\mathbf{m}_\theta$ using the hybrid estimator

$$\hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) := (1-\lambda_t)\,\hat{\mathbf{x}}_0^{\text{marg}}(\mathbf{x}_t^{1:K}, t) + \lambda_t\,\hat{\mathbf{x}}_0^{\text{cond}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t), \quad (31)$$

where $\theta = (\theta_{\text{cond}}, \theta_{\text{marg}})$, $\hat{\mathbf{x}}_0^{\text{cond}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)$ is defined in (23) and $\hat{\mathbf{x}}_0^{\text{marg}}(\mathbf{x}_t^{1:K}, t)$ is defined in (28). The weight $\lambda_t \in [0, 1]$ depends on the conditional SNR (26):

$$\lambda_t := \frac{\text{SNR}_t^{\text{cond}}}{\text{SNR}_t^{\text{cond}} + \text{snr_threshold}}, \quad (32)$$

where, `snr_threshold` is a tunable constant controlling the scale of the interpolation. λ_t is small near $t \approx T$, reflecting low confidence in the conditional estimator, and increases as t decreases. We illustrate the empirical performance trade-off between the marginal and conditional estimators in Appendix D.4.

3.6 Training and Sampling Algorithms

Algorithm 1 Training Conditional Estimator

repeat
  $(\mathbf{x}_0, \mathbf{x}_T^{1:K}) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K})$
  $t \sim \text{Uniform}\{1, \dots, T-1\}$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dK})$
  $\mathbf{x}_t^{1:K} = C^{(0)}_{t|0,T}\mathbf{A}\mathbf{x}_0 + C^{(T)}_{t|0,T}\mathbf{x}_T^{1:K} + \sigma_{t|0,T}\boldsymbol{\epsilon}$
  Gradient step on $\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)\|^2$
until converged

Algorithm 2 Training Marginal Estimator

repeat
  Same as Algorithm 1, 2-5
  $\boldsymbol{\eta}_t = (\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0)/\sqrt{1-\bar{\alpha}_t}$
  Gradient step on $\|\boldsymbol{\eta}_t - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t)\|^2$
until converged

Algorithm 3 Sampling

$\mathbf{x}_T^{1:K} \sim p_{\text{data}}(\mathbf{x}_T^{1:K})$
for $t = T, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dK})$
  if $t = T$ **then**
    $\mathbf{x}_{T-1}^{1:K} = \boldsymbol{\mu}_{T-1|0,T}(\mathbf{A}\hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_T^{1:K}, \mathbf{x}_T^{1:K}, T-1), \mathbf{x}_T^{1:K}) + \sigma_{T-1|0,T}\mathbf{z}$
  else if $t = T-1, \dots, 2$ **then**
    $\mathbf{x}_{t-1}^{1:K} = \boldsymbol{\mu}_{t-1|0,t}\big(\mathbf{A}\hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t), \mathbf{x}_t^{1:K}\big) + \sigma_{t-1|0,t}\mathbf{z}$
  else if $T = 1$ **then**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
    $\mathbf{x}_0 = \hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K}, 1) + \sigma_{\text{rec}}\mathbf{z}$
  end if
end for
return $\mathbf{x}_0$

4 Experiments

Implementation details and dataset descriptions are provided in Appendix D.

4.1 Multi-source Behavior and Calibration Properties

We use CIFAR-10 [13] (32×32) to study MB-DDPM calibration under four degradations – Gaussian noise, Gaussian blur, JPEG compression (quality factor 5), and $4\times$ bicubic super-resolution – each treated as a source image for predicting the clean target image. On average, adding more sources improves performance. The four-source setting outperforms all single-source configurations and achieves the best overall performance across evaluated combinations (Figure 5, Appendix E.1.1), while introducing only marginal runtime overhead (Appendix E.1.2).

To quantify source contributions, we compute a Shapley-style approximation of marginal performance drops when removing each source (Appendix E.1.3), showing different contributions that depend on degradation severity, with less corrupted sources contributing more than heavily degraded ones.

We further analyze the effect of source disagreement on reconstruction quality (Appendix E.2). We find that higher disagreement increases ambiguity and degrades performance, consistent with the expected behavior that conflicting sources provide less reliable guidance.

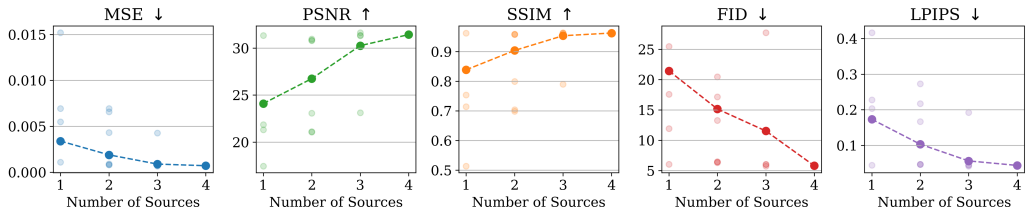


Figure 5: Performance of MB-DDPM for image prediction (k degradations $\rightarrow$ clean) on the CIFAR-10 test set using k =1–4 sources. Faded points correspond to a specific combination of sources. Solid points represent fusion across sources combinations.

4.2 Application to Brain MRI for Multiple Sclerosis

MS is a chronic neurological disease affecting millions worldwide [18]. Among MRI contrasts, FLAIR is particularly critical for MS as it enhances T2 lesion visibility. However, FLAIR acquisition can be time-consuming and prone to artifacts, motivating the need for accurate prediction from routinely acquired contrasts. We evaluate MB-DDPM on FLAIR prediction from T1, T2, and PD contrasts (168×208). We use the NO.MS dataset [7], one of the largest clinical trial datasets for MS. MB-DDPM achieves strong reconstruction quality and consistently outperforms baselines across all quantitative metrics (Figure 1, Figure 6a, Table 1).

To assess clinical relevance, we evaluate downstream T2 lesion volume estimation. T2 lesions are hyperintense (bright) regions in MRI reflecting neuroinflammatory damage and serve as key biomarkers for diagnosis, disease monitoring, and prognosis [19]. We use the segmentation model developed in prior works [30, 27] that takes T1 and FLAIR contrasts as input to predict lesion masks, which are aggregated across slices to obtain volumetric estimates (Appendix E.3). We compute lesion volumes using either predicted or reference FLAIR (together with T1), and compare the resulting estimates (Figure 6b). MB-DDPM achieves near-perfect agreement with ground truth (Pearson = 0.99, ICC = 0.99), indicating accurate recovery of T2 lesions from the MB-DDPM predicted FLAIR images.

4.3 Comparison with Existing Methods

We compare three classes of baselines: (i) score-based diffusion models for inverse problems, including DPS [6] and CDE [4]; (ii) conditional diffusion models (CDM [37]); and (iii) diffusion bridge models (I^2SB [16]), with further details on related work in Appendix C. Since diffusion bridge

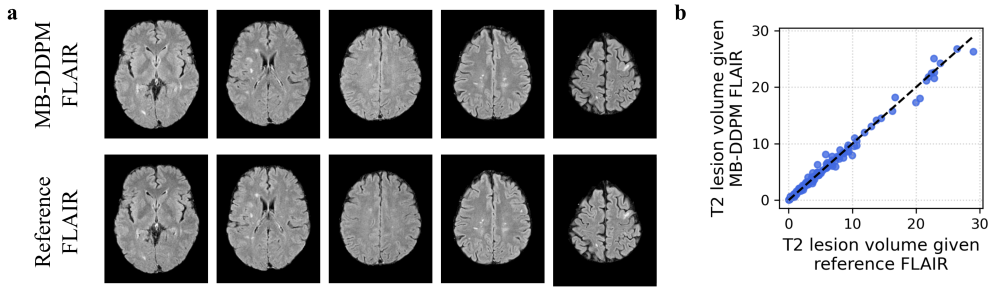


Figure 6: MB-DDPM (a) FLAIR prediction from T1, T2, and PD (test set). Lesions appear as hyperintense (bright) regions (b) Lesion volume estimation (test set), each point is an MRI scan.

models do not support multiple sources, we evaluate them independently on each source–target pair and report two variants: best-source (on FID) and fusion (averaging).

We further evaluate MB-DDPM on image restoration using ImageNet [8] (center-cropped to 128×128) under the same four degradations introduced earlier. We also report results on CIFAR-10 and MRI data introduced in the previous sections.

Across all three datasets, MB-DDPM achieves the lowest FID among all evaluated methods, with particularly substantial improvements on ImageNet and MRI, demonstrating consistently strong perceptual quality (Table 1, Appendix E.4). It also achieves the best or tied-best MSE and PSNR on every dataset, while matching or outperforming in SSIM and LPIPS, all at competitive runtime (Appendix E.4.2).

Score-based methods (DPS, CDE) and conditional diffusion models (CDM) rely on a single noisy state and start from noise, effectively collapsing multiple observations into a single inference pathway. This limits their ability to integrate complementary sources and degrades performance under strong or mixed degradations. Diffusion bridge models (I^2SB) better match the data generation process but remain restricted to single-source settings. In contrast, MB-DDPM jointly models multiple sources within a unified bridge framework, enabling explicit interaction between sources and simultaneous reverse trajectories (Appendix E.5).

Method	MSE ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
CIFAR-10					
DPS [6]	0.007	20.76	0.72	21.79	0.28
CDE [4]	0.002	27.33	0.90	45.06	0.12
CDM [37]	<u>0.001</u>	<u>31.29</u>	<u>0.96</u>	<u>6.01</u>	**0.04**
I^2SB [16] (Best)	0.018	16.90	0.47	19.18	0.38
I^2SB [16] (Fusion)	0.011	18.90	0.55	32.69	0.36
MB-DDPM (ours)	**0.001**	**31.42**	**0.96**	**5.79**	<u>0.04</u>
ImageNet					
DPS [6]	0.009	21.11	0.50	48.75	0.47
CDE [4]	0.009	20.19	0.42	41.57	0.45
CDM [37]	<u>0.003</u>	<u>26.62</u>	**0.78**	<u>7.03</u>	<u>0.19</u>
I^2SB [16] (Best)	0.007	22.26	0.54	29.88	0.39
I^2SB [16] (Fusion)	0.008	21.67	0.50	37.46	0.40
MB-DDPM (ours)	**0.003**	**26.67**	<u>0.77</u>	**5.25**	**0.18**
MRI					
CDE [4]	0.008	26.84	0.77	28.26	0.25
CDM [37]	<u>0.008</u>	<u>27.48</u>	<u>0.87</u>	<u>5.03</u>	<u>0.07</u>
I^2SB [16] (Best)	0.032	21.43	0.74	7.89	0.13
I^2SB [16] (Fusion)	0.014	24.75	0.79	14.74	0.11
MB-DDPM (ours)	**0.007**	**27.93**	**0.87**	**3.47**	**0.06**

Table 1: Method comparison for image prediction across CIFAR-10, ImageNet (4 degradations → clean) and MRI (T1, T2, PD → FLAIR) test set.

4.4 Robustness to Violations of Model Assumptions

MB-DDPM is formulated under three main assumptions: (i) sources provide complementary information about the target; (ii) all sources are available; and (iii) sources are conditionally independent given the target (Equation (9)). In this section, we summarize additional experiments that investigate the behaviour of MB-DDPM when these assumptions do no hold.

First, in Appendix E.6, we investigate whether including non-complementary sources in an otherwise informative source set degrades reconstruction quality, and whether the model can effectively down-weight their contribution. We extend the multi-source experiments on CIFAR-10 (Section 4.1) and find that adding a redundant source has a negligible impact on reconstruction performance, while introducing heavily corrupted, uninformative, or conflicting sources leads to only modest degradation.

Second, in Appendix E.7, we propose an architectural extension to both neural networks for the marginal estimator $\boldsymbol{\eta}_{\theta_{\text{marg}}}$ and the conditional estimator $\boldsymbol{\epsilon}_{\theta_{\text{cond}}}$ to explicitly account for a missing source among $\mathbf{x}_T^{1:K}$. Specifically, a binary source-availability indicator is provided as an additional input to both networks. We evaluate this extension on CIFAR-10 (Section 4.1) under different missingness scenarios. The results show that the model can adapt to varying source availability and continue to exploit the information provided by a source whenever it is present.

Finally, in Appendix E.8, we conduct an experiment in which two Gaussian-noise-corrupted sources are generated with varying degrees of correlation conditionally on the target. The results suggest that MB-DDPM remains effective under mild violations of conditional independence, consistent with its strong empirical performance on MRI data, where correlated artifacts may arise from acquisition protocols or scanner-specific effects. However, as the conditional correlation between sources increases, the benefit of incorporating multiple observations progressively diminishes. Adding a strongly conditionally dependent source can even degrade reconstruction performance.

5 Conclusion

We introduce MB-DDPM, a multi-source diffusion bridge model that integrates complementary sources within a unified generative framework. MB-DDPM constructs independent stochastic forward trajectories from the target to each source, and performs joint generation through a shared reverse diffusion process (Figures 1, 2).

Across both restoration and translation tasks, this formulation resolves ambiguities that cannot be recovered from any single observation, yielding a well-calibrated model. It consistently improves reconstruction perceptual quality across datasets with only marginal overhead as the number of sources increase.

MB-DDPM demonstrates promising performance on a clinically relevant downstream task, with T2 lesion volumes estimated from predicted FLAIR images showing strong agreement with those obtained from real FLAIR images. By conditioning synthesis on the available T1, T2, and PD contrasts, MB-DDPM leverages subject-specific anatomical and structural information, which helps constrain the generated images. While these results provide encouraging evidence that predicted FLAIR preserves clinically relevant lesion information, they do not exclude clinically significant errors, missing or hallucinated findings at the individual lesion level or establish clinical generalization. We therefore envision predicted FLAIR as a potential adjunct to clinical interpretation rather than a replacement for acquired FLAIR scans.

Beyond medical imaging, MB-DDPM provides a general framework for multi-source image restoration and synthesis, with potential applications in domains such as remote sensing, where optical, multispectral, and SAR observations can be combined, as well as computational photography and image restoration from multiple degraded observations. More broadly, the framework is applicable to settings in which multiple observations provide complementary information about a common target and satisfy the conditional-independence assumptions underlying our probabilistic formulation.

The limitations of MB-DDPM include its reliance on a fixed set of source observations being available during both training and sampling, as well as its assumption that sources provide complementary information and are conditionally independent given the target. Although our experiments show that MB-DDPM can effectively mitigate the impact of redundant sources and accommodate missing sources through an appropriate architectural extension, its ability to benefit from multiple observations ultimately depends on the complementary information they provide. In particular, the inclusion of heavily corrupted, uninformative, or conflicting sources can still lead to a modest degradation in reconstruction performance, while strong conditional dependencies between sources can reduce the benefits of multi-source reconstruction.

Acknowledgments and Disclosures of Funding

We are deeply grateful to Yang Sun for providing the T2 lesion volume outputs from the predicted FLAIR images used in Section 4.2. We acknowledge his work in preparing and aggregating the slice-wise MRI inputs into the appropriate format for the segmentation model, passing them through the segmentation model, and transforming its lesion predictions into lesion volume. We also appreciate

his detailed explanations of the segmentation model, which enabled us to document it in Appendix E.3. We thank the reviewers for their constructive feedback and helpful suggestions, which helped improve the quality and clarity of this work. MM acknowledges funding from the French National Research Agency under the France 2030 program (ANR-23-IACL-0008) and declares no competing interest. HG is supported by Oxford-Novartis collaboration for AI in Medicine and declares no competing interest.

References

[1] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[3] M.R. Banham and A.K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, 1997.

[4] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional Image Generation with Score-Based Diffusion Models, 2021. *arXiv preprint arXiv:2111.13606*.

[5] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, 2021.

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

[7] Frank Dahlke, Douglas L Arnold, Piet Aarden, Habib Ganjgahi, Dieter A Häring, Jelena Čuklina, Thomas E Nichols, Stephen Gardiner, Robert Bermel, and Heinz Wiendl. Characterisation of MS phenotypes across the age span using a novel data set integrating 34 clinical trials (NO.MS cohort): Age is a key contributor to presentation. *Multiple Sclerosis Journal*, 27(13):2062–2076, January 2021.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, page 248–255. IEEE, June 2009.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

[10] Jeremy Heng, Valentin De Bortoli, Arnaud Doucet, and James Thornton. Simulating diffusion bridges with score matching. *Biometrika*, 112(4), 2025.

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[14] Guofa Li, Yifan Yang, Xingda Qu, Dongpu Cao, and Keqiang Li. A deep learning based image enhancement approach for autonomous driving at night. *Knowledge Based System*, 213(C), 2021.

[15] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[16] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-Image Schrödinger Bridge. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[17] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations*, 2023.

[18] Tullman J. Mark. Overview of the Epidemiology, Diagnosis, and Disease Progression Associated With Multiple Sclerosis. *The American Journal of Managed Care*, 19, 2013.

[19] Marisa P. McGinley, Carolyn H. Goldschmidt, and Alexander D. Rae-Grant. Diagnosis and Treatment of Multiple Sclerosis: A Review. *JAMA*, 325(8):765, 2021.

[20] Alessandro Micheli, Mélodie Monod, and Samir Bhatt. Diffusion Models for Inverse Problems in the Exponential Family. *arXiv*, 2025.

[21] Mukesh C Motwani, Mukesh C Gadiya, Rakhi C Motwani, and Frederick C Harris. Survey of image denoising techniques. *Proceedings of GSPX*, 27:27–30, 2004.

[22] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[23] William Hadley Richardson. Bayesian-Based Iterative Method of Image Restoration$*$. *Journal of the Optical Society of America*, 62(1):55–59, 1972.

[24] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-Image Diffusion Models. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022.

[25] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2023.

[26] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger Bridge Matching. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023.

[27] Amitis Shidani, Tyler Farghly, Yang Sun, Habib Ganjgahi, and George Deligiannidis. Beyond real data: Synthetic data through the lens of regularization. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

[28] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving Inverse Problems in Medical Imaging with Score-Based Generative Models. In *International Conference on Learning Representations*, 2022.

[29] 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.

[30] Yang Sun, Laura Gaetano, Elizabeth Fisher, Dieter Häring, Piet Aarden, Silje Hallan-Nickell, Ann-Christine Pehrs-Jurvillier, Kerstin Bendfeldt, Charlotte Brügger, Robert Bermel, Heinz Wiendl, Douglas L. Arnold, Thomas Nichols, and Habib Ganjgahi. An automated deep learning segmentation tool for accurate and generalizable ms lesion quantification. *Multiple Sclerosis Journal*, 31(3):179, 2025.

[31] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[32] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[34] Pascal Vincent. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[35] Gregory K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1), 1992.

[36] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[37] Zheyuan Zhan, Defang Chen, Jian-Ping Mei, Zhenghe Zhao, Jiawei Chen, Chun Chen, Siwei Lyu, and Can Wang. Conditional Image Synthesis with Diffusion Models: A Survey. *Transactions on Machine Learning Research*, 2025.

[38] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric . In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising Diffusion Bridge Models. In *The Twelfth International Conference on Learning Representations*, 2024.

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. The claims made match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion points out the model assumptions made in the paper and reflects on how these assumptions might be violated in practice and what the implications would be.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Propositions 3.1, 3.2 and 3.3, as well as all the formulas, are numbered and cross-referenced. The assumptions are stated in the statement and the proof is presented in Appendix F.4. The theorems and Lemmas that the proof relies upon are properly referenced.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper fully discloses all information necessary to reproduce the main experimental results. Specifically:

- Pseudocode in Algorithms 1, 2 and 3 for *MB-DDPM* is provided to clarify the algorithmic steps.
- Additional implementation details for *MB-DDPM* are included Appendix D to facilitate replication.
- The source code is made publicly available via a GitHub repository, enabling direct access to the implementation used to produce the reported results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides open access to both the code, along with sufficient instructions to faithfully reproduce the main experimental results:

   - The source code is made publicly available via a GitHub repository, enabling direct access to the implementation used to produce the reported results.
   - The data CIFAR-10 and ImageNet are publicly available. The brain MRI from NO.MS dataset are not publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all training and test details (data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: The paper does not report statistical significance tests. It focuses on settings where statistical testing was not applicable for supporting the main claims.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information on the computer resources is provided in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer:[Yes]

   Justification: To the best of our knowledge and judgment, the research complies fully with the NeurIPS Code of Ethics. In our view, the work does not pose potential harms caused by the research process, nor does it present any adverse societal impact or potentially harmful consequences.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The potential positive societal impacts of the paper are summarized in the conclusion. No negative societal impacts have been identified.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The baseline methods used in the paper are properly credited, with citations to the original papers. For the datasets, any original studies associated with the data are cited. All licenses and terms of use have been respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The source code is made publicly available via a GitHub repository under the MIT License, enabling direct access to the implementation used to produce the reported results.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix Contents

A Denoising Diffusion Probabilistic Models (DDPMs) 23

A.1 Forward Diffusion Process . . . 23
A.2 Reverse Diffusion Process . . . 23
A.3 Variational Bound . . . 23
A.4 Parameterization of Reverse Diffusion Transitions . . . 24
A.5 Learning an Estimator for $\mathbf{x}_0$ . . . 24
A.6 Training and Sampling Algorithms . . . 25

B Denoising Diffusion Probabilistic Models with Fixed Endpoints 26

B.1 Bridge Forward Diffusion Process . . . 26
B.2 Bridge Reverse Diffusion Process . . . 27
B.3 Variational Bound . . . 27
B.4 Parameterization of Bridge Reverse Diffusion Process Transitions . . . 27
B.5 Learning an Estimator for $\mathbf{x}_0$ . . . 28
B.6 Training and Sampling Algorithms . . . 30

C Related Work 31

C.1 Diffusion models for inverse problems . . . 31
C.2 Diffusion Posterior Sampling (DPS) . . . 32
C.3 Conditional Denoising Estimator (CDE) . . . 33
C.4 Conditional Diffusion Models (CDMs) . . . 33
C.5 Image-to-Image Schrodinger Bridge (I^2SB) . . . 33
C.6 Limitations of related work . . . 33

D Implementation details 35

D.1 Data . . . 35
D.2 Diffusion schedule . . . 35
D.3 Training set-up . . . 35
D.4 Tuning `snr_threshold` . . . 36
D.5 Evaluation Metrics . . . 39
D.6 Best-source and fusion . . . 39
D.7 Implementation of baselines . . . 40

E Additional Results 42

E.1 Effect of Source Multiplicity . . . 42
E.1.1 Increasing the number of sources . . . 42
E.1.2 Running time . . . 44
E.1.3 Shapley-style values . . . 45
E.2 Effect of Source Disagreement . . . 46

E.3 T2 Lesion Volume Prediction 51
E.4 Comparison with Baselines 54
E.4.1 Quantitative Comparison 54
E.4.2 Running time 59
E.5 Visualizing the MB-DDPM Generative Process 61
E.6 Effect of Non-Complementary Sources 65
E.7 Handling Missing Sources 66
E.8 Violations of the Source Conditional Independence Assumption 67

F Proofs **68**
F.1 Useful Lemmas 68
F.2 Proofs for Denoising Diffusion Probabilistic Models (DDPMs) 68
F.2.1 Proof of Proposition A.1 68
F.2.2 Proof of Proposition A.2 69
F.3 Proofs for Denoising Diffusion Probabilistic Models with Fixed Endpoints 70
F.3.1 Proof of Propositions B.1 and B.2 70
F.3.2 Proof of Proposition B.3 71
F.3.3 Proof of Proposition B.4 73
F.3.4 Proof of Proposition B.5 74
F.3.5 Proof of Proposition B.6 75
F.4 Proofs for Multi-Bridge Denoising Diffusion Probabilistic Models 75
F.4.1 Proof of Proposition 3.1 75
F.4.2 Proof of Proposition 3.2 77
F.4.3 Proof of Proposition 3.3 80

A Denoising Diffusion Probabilistic Models (DDPMs)

In this section, we review standard results from Denoising Diffusion Probabilistic Models (DDPMs) [12]. DDPMs define a generative model by learning to reverse a fixed Markovian noising process. Let $\mathbf{x}_0 \in \mathbb{R}^d$ denote a data sample from a source distribution $p_{\text{data}}(\mathbf{x}_0)$, and let $\mathbf{x}_{1:T} \in \mathbb{R}^d$ be latent variables for each intermediate timestep.

A.1 Forward Diffusion Process

The forward diffusion process is a fixed Markov chain that gradually adds Gaussian noise to the data over T timesteps according to a variance schedule $\beta_{1:T} \in (0,1)$. Each transition is defined as

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{\alpha_t}\,\mathbf{x}_{t-1}, (1-\alpha_t)\,\mathbf{I}_d\big), \quad \alpha_t = 1 - \beta_t, \tag{33}$$

for $t = 1, \ldots, T$. The joint distribution over the entire trajectory factorizes as

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}).$$

Define the cumulative product $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. The marginal distribution at timestep t has the closed form

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\Big(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\,\mathbf{I}_d\Big), \tag{34}$$

for $t = 1, \ldots, T$, which can equivalently be expressed as a deterministic transformation of standard Gaussian noise:

$$\mathbf{x}_t \mid \mathbf{x}_0 = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{35}$$

A.2 Reverse Diffusion Process

The generative model aims to reverse the forward diffusion process. It defines a reverse-time Markov chain with joint distribution:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \quad p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I}_d).$$

The model likelihood is the marginal

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T})\,\mathrm{d}\mathbf{x}_{1:T}. \tag{36}$$

The goal is to draw samples from $p_\theta(\mathbf{x}_0)$. Since direct sampling from this distribution is intractable, the method instead samples by simulating the reverse-time chain, starting from a data sample $\mathbf{x}_T \sim p(\mathbf{x}_T) = \mathcal{N}(0, \mathbf{I}_d)$ and iteratively sampling

$$\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \quad \text{for } t = T, \ldots, 1.$$

A.3 Variational Bound

We start by deriving the forward posterior distribution, which describes the conditional distribution of the previous step $\mathbf{x}_{t-1}$ given the current step $\mathbf{x}_t$ and the initial state $\mathbf{x}_0$:

$$\begin{aligned} q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) &= \mathcal{N}\big(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{t-1|0,t}(\mathbf{x}_0, \mathbf{x}_t), \sigma^2_{t-1|0,t}\,\mathbf{I}_d\big), \qquad t = 2, \ldots, T, \\ \text{where } \boldsymbol{\mu}_{t-1|0,t}(\mathbf{x}_0, \mathbf{x}_t) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \text{ and } \sigma^2_{t-1|0,t} = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \end{aligned} \tag{37}$$

DDPM optimizes a variational lower bound on the negative log-likelihood (36) given by

$$\begin{aligned} -\log p_\theta(\mathbf{x}_0) \le\; & D_{\mathrm{KL}}\big(q(\mathbf{x}_T \mid \mathbf{x}_0) \| p(\mathbf{x}_T)\big) \\ & + \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\Big[D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) \| p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\big)\Big] \\ & + \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}\big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)\big]. \end{aligned} \tag{38}$$

A.4 Parameterization of Reverse Diffusion Transitions

To define the reverse diffusion transitions, we introduce an estimator $\mathbf{m}_\theta(\mathbf{x}_t, t)$ of the unknown initial state $\mathbf{x}_0$. We then parameterize the reverse diffusion transitions so as to minimize the variational lower bound (38). Specifically, each RDP transition is designed to match the forward posterior transition (37), with its mean obtained by replacing $\mathbf{x}_0$ with $\mathbf{m}_\theta(\mathbf{x}_t, t)$:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\big(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{t-1|0,t}(\mathbf{m}_\theta(\mathbf{x}_t, t), \mathbf{x}_t), \sigma^2_{t-1|0,t}\, \mathbf{I}_d\big), \quad t = 2, \dots, T. \tag{39}$$

For $t = 1$, the forward posterior collapses to a Dirac delta, so we replace it with a Gaussian of small variance $\sigma^2_{\mathrm{rec}} \ll 1$:

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1) = \mathcal{N}(\mathbf{x}_0; \mathbf{m}_\theta(\mathbf{x}_1, 1), \sigma^2_{\mathrm{rec}}\, \mathbf{I}_d). \tag{40}$$

A.5 Learning an Estimator for $\mathbf{x}_0$

Optimal Estimator. The following proposition shows that minimizing the expected variational bound (38) is achieved if and only if the estimator $\mathbf{m}_\theta(\mathbf{x}_t, t)$ recovers the posterior mean of the original data $\mathbf{x}_0$ given $\mathbf{x}_t$.

Proposition A.1 (Variational Bound Minimization for DDPM)**.** *Let $\mathbf{x}_0 \sim p_{\mathrm{data}}(\mathbf{x}_0)$ and $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_0)$ as defined in* (34)*. Let the forward posterior distribution $q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)$ be defined as in* (37) *for $t = 2, \dots, T$. Lastly, let the reverse diffusion transitions $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ be parameterized through an estimator $\mathbf{m}_\theta(\mathbf{x}_t, t)$ as in* (39)-(40) *for $t = 1, \dots, T$.*

Then the expected variational bound (38) *over the data distribution is minimized if and only if*

$$\mathbf{m}^*_\theta(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t], \qquad \textit{for } t = 1, \dots, T. \tag{41}$$

The proof of Proposition A.1 is deferred to Appendix F.2.1.

Training Objective. In practice, $\mathbf{m}_\theta(\mathbf{x}_t, t)$ is implemented using a neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ that predicts the noise component in the forward diffusion marginal (35):

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{42}$$

The corresponding noise-prediction objective, averaging over timesteps t and data samples $\mathbf{x}_0$, is:

$$\begin{aligned} &\mathcal{J}_{\mathrm{DDPM}}(\theta) = \mathbb{E}_{t \sim \mathrm{Unif}\{1,\dots,T\},\, \mathbf{x}_0 \sim p_{\mathrm{data}}(\mathbf{x}_0),\, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\mathcal{L}_{\mathrm{DDPM}}(\theta, \mathbf{x}_0, \boldsymbol{\epsilon}, t)\right], \\ &\mathcal{L}_{\mathrm{DDPM}}(\theta, \mathbf{x}_0, \boldsymbol{\epsilon}, t) = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t\right) \right\|^2. \end{aligned} \tag{43}$$

The following proposition shows that minimizing the noise-prediction objective (43) yields an estimator that achieves the minimum of the variational bound defined in Proposition A.1.

Proposition A.2 (Optimality of the DDPM Estimator)**.** *Let $\mathbf{x}_0 \sim p_{\mathrm{data}}(\mathbf{x}_0)$ and $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_0)$ as defined in* (34)*. For a fixed $t \in \{1, \dots, T\}$, let $\boldsymbol{\epsilon}^*_\theta(\mathbf{x}_t, t)$ be the minimizer of* (43)*. Then, in the infinite-data and infinite-capacity limit,*

$$\boldsymbol{\epsilon}^*_\theta(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\, \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]}{\sqrt{1 - \bar{\alpha}_t}}. \tag{44}$$

Consequently, the induced estimator (42)

$$\hat{\mathbf{x}}^*_0(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\Big(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}^*_\theta(\mathbf{x}_t, t)\Big) \tag{45}$$

recovers the posterior mean,

$$\hat{\mathbf{x}}^*_0(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]. \tag{46}$$

The proof of Proposition A.2 is deferred to Appendix F.2.2.

A.6 Training and Sampling Algorithms

Algorithm 4 DDPM: Training

repeat
  $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$
  $t \sim \text{Uniform}\{1, \dots, T\}$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
  Gradient step on $\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2$
until converged

Algorithm 5 DDPM: Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
for $t = T, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
  if $t > 1$ **then**
    $\mathbf{x}_{t-1} = \boldsymbol{\mu}_{t-1|0,t}(\hat{\mathbf{x}}_0(\mathbf{x}_t, t), \mathbf{x}_t) + \sigma_{t-1|0,t}\mathbf{z}$
  else
    $\mathbf{x}_0 = \hat{\mathbf{x}}_0(\mathbf{x}_1, 1) + \sigma_{\text{rec}}\mathbf{z}$
  end if
end for
return $\mathbf{x}_0$

B Denoising Diffusion Probabilistic Models with Fixed Endpoints

Assume that the endpoints of a diffusion bridge lie in $\mathbb{R}^d$ and are drawn from an unknown joint data distribution:

$$(\mathbf{x}_0, \mathbf{x}_T) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T). \tag{47}$$

The goal is to construct a generative process that can approximately sample from the conditional distribution of interest,

$$p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{x}_T) \propto p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T), \tag{48}$$

enabling sampling of $\mathbf{x}_0$ given $\mathbf{x}_T$.

B.1 Bridge Forward Diffusion Process

We consider a variant of the FDP conditioned on a fixed terminal state $\mathbf{x}_T$, yielding a stochastic bridge between $\mathbf{x}_0$ and $\mathbf{x}_T$. For convenience, we define the reverse cumulative product $\tilde{\alpha}_t$ that represents the fraction of signal remaining from step t to the terminal time T. It satisfies

$$\tilde{\alpha}_t := \prod_{s=t+1}^{T} \alpha_s = \frac{\bar{\alpha}_T}{\bar{\alpha}_t}. \tag{49}$$

We start by describing the transition distribution of the bridge FDP in the next Proposition.

Proposition B.1 (Bridge Forward Diffusion Transition Distribution)**.** *Consider the FDP with transitions $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ as in* (33)*. For $t = 1, \dots, T-1$, the transition distribution of the bridge FDP is*

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T) = \mathcal{N}\Big(\mathbf{x}_t;\, \boldsymbol{\mu}_{t|t-1,T}(\mathbf{x}_{t-1}, \mathbf{x}_T),\, \sigma^2_{t|t-1,T}\, \mathbf{I}_d\Big),$$

$$\textit{where } \boldsymbol{\mu}_{t|t-1,T}(\mathbf{x}_{t-1}, \mathbf{x}_T) = \frac{\sqrt{\alpha_t}(1-\tilde{\alpha}_t)}{1-\tilde{\alpha}_{t-1}}\, \mathbf{x}_{t-1} + \frac{\sqrt{\tilde{\alpha}_t}\beta_t}{1-\tilde{\alpha}_{t-1}}\, \mathbf{x}_T, \textit{ and } \sigma^2_{t|t-1,T} = \frac{(1-\tilde{\alpha}_t)\beta_t}{1-\tilde{\alpha}_{t-1}}.$$

The proof of Proposition B.1 is deferred to Appendix F.3.1.

The joint distribution of the bridge FDP, conditioned on both endpoints, factorizes according to the Markov property as

$$q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T) := \prod_{t=1}^{T-1} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T). \tag{50}$$

The bridge FDP also admits a closed-form marginal distribution at any intermediate timestep as shown in the next Proposition.

Proposition B.2 (Bridge Forward Diffusion Marginal Distribution)**.** *Consider the FDP with transitions $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ as in* (33)*. For $t = 1, \dots, T-1$, the marginal distribution of the bridge FDP is*

$$q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) = \mathcal{N}\Big(\mathbf{x}_t;\, \boldsymbol{\mu}_{t|0,T}(\mathbf{x}_0, \mathbf{x}_T),\, \sigma^2_{t|0,T}\, \mathbf{I}_d\Big)\,,$$

$$\textit{where } \boldsymbol{\mu}_{t|0,T}(\mathbf{x}_0, \mathbf{x}_T) = C^{(0)}_{t|0,T}\, \mathbf{x}_0 + C^{(T)}_{t|0,T}\, \mathbf{x}_T, \textit{ and } \sigma^2_{t|0,T} = \frac{(1-\bar{\alpha}_t)(1-\tilde{\alpha}_t)}{1-\bar{\alpha}_T}, \tag{51}$$

$$\textit{with } C^{(0)}_{t|0,T} = \frac{\sqrt{\tilde{\alpha}_t}(1-\tilde{\alpha}_t)}{1-\bar{\alpha}_T}, \quad C^{(T)}_{t|0,T} = \frac{\sqrt{\tilde{\alpha}_t}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_T}.$$

Proposition B.2 has previously appeared in [16, 39]; for completeness, we provide a self-contained proof in Appendix F.3.1. Equivalently, the bridge FDP marginal can be expressed as a deterministic transformation of standard Gaussian noise:

$$\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T = C^{(0)}_{t|0,T}\, \mathbf{x}_0 + C^{(T)}_{t|0,T}\, \mathbf{x}_T + \sigma_{t|0,T}\, \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{52}$$

B.2 Bridge Reverse Diffusion Process

The generative model aims to reverse the bridge FDP via learned Gaussian transitions. The joint distribution of the bridge RDP is defined as

$$p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T) = p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T) \prod_{t=1}^{T-1} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T). \tag{53}$$

The bridge model likelihood $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T)$ is defined as the marginal of the bridge RDP:

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T) = \int p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T)\, \mathrm{d}\mathbf{x}_{1:(T-1)}. \tag{54}$$

The goal is to draw samples from $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T)$. Since direct sampling from this distribution is intractable, the method instead samples by simulating the reverse-time chain, starting from a data sample $\mathbf{x}_T$ and iteratively sampling

$$\mathbf{x}_{T-1} \sim p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T), \quad \mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T) \text{ for } t = T-1, \dots, 1. \tag{55}$$

B.3 Variational Bound

Training a diffusion bridge is performed by optimizing a variational lower bound on the negative bridge model log-likelilhood (54).

Proposition B.3 (Variational Lower Bound for Bridge DDBM)**.** *Let $q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T)$ be the joint distribution of the bridge FDP* (50)*. Further, let $p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T)$ be the joint distribution of the bridge RDP* (53)*. Lastly, let $q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)$ be the forward posterior distribution* (37) *for $t = 2, \dots, T$.*

The variational lower bound on the negative bridge model log-likelihood (54) *is given by*

$$\begin{aligned} -\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T) \le{}& D_{\mathrm{KL}}\Big(q(\mathbf{x}_{T-1} \mid \mathbf{x}_0, \mathbf{x}_T) \,\|\, p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)\Big) \\ &+ \sum_{t=2}^{T-1} \mathbb{E}_{q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)}\Big[D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)\big)\Big] \\ &+ \mathbb{E}_{q(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{x}_T)}\Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)\Big]. \end{aligned} \tag{56}$$

The proof of Proposition B.3 is provided in Appendix F.3.2.

B.4 Parameterization of Bridge Reverse Diffusion Process Transitions

To define the transitions of the bridge RDP, we introduce an estimator $\mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t)$ for the unknown initial state $\mathbf{x}_0$. The bridge RDP transitions are then parameterized in terms of $\mathbf{m}_\theta$ so as to minimize the variational lower bound (56). Specifically, each RDP transition is designed to match the forward posterior transition (37), with its mean obtained by replacing $\mathbf{x}_0$ with $\mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t)$.

The first reverse step corresponds to sampling $\mathbf{x}_{T-1}$ from $\mathbf{x}_T$. Since $\mathbf{x}_{T-1}$ is not yet available, the estimator cannot directly depend on it. However, in the bridge FDP the conditional variance at $t = T-1$ is very small ($\sigma^2_{T-1|0,T} \ll 1$), implying that $\mathbf{x}_{T-1}$ concentrates around $\mathbf{x}_T$. We therefore approximate $\mathbf{x}_{T-1} \approx \mathbf{x}_T$ when evaluating the estimator, leading to

$$p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T) = \mathcal{N}\Big(\mathbf{x}_{T-1}; \boldsymbol{\mu}_{T-1|0,T}\big(\mathbf{m}_\theta(\mathbf{x}_T, \mathbf{x}_T, T-1), \mathbf{x}_T\big), \sigma^2_{T-1|0,T}\, \mathbf{I}_d\Big). \tag{57}$$

For the timesteps $t = T-1, \dots, 2$, the reverse transitions follow directly from the forward posterior form:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T) = \mathcal{N}\Big(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{t-1|0,t}\big(\mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t), \mathbf{x}_t\big), \sigma^2_{t-1|0,t}\, \mathbf{I}_d\Big). \tag{58}$$

For $t = 1$, the forward posterior collapses to a Dirac delta, so we replace it with a Gaussian of small variance $\sigma^2_{\mathrm{rec}} \ll 1$:

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T) = \mathcal{N}\big(\mathbf{x}_0; \mathbf{m}_\theta(\mathbf{x}_1, \mathbf{x}_T, 1), \sigma^2_{\mathrm{rec}}\, \mathbf{I}_d\big). \tag{59}$$

B.5 Learning an Estimator for $\mathbf{x}_0$

Optimal Estimator. The following proposition shows that minimizing the expected variational bound (56) is achieved if and only if the estimator $\mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t)$ recovers the posterior mean of the original data $\mathbf{x}_0$ given the $\mathbf{x}_t$ and $\mathbf{x}_T$.

Proposition B.4 (Variational Bound Minimization for Bridge DDBM)**.** *Let* $(\mathbf{x}_0, \mathbf{x}_T) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T)$ *and* $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)$ *as defined in* (51)*. Assume the reverse transitions* $p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)$ *and* $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)$ *for* $t = 2, \ldots, T-1$ *are parameterized through an estimator* $\mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t)$ *as in* (57)*,* (58) *and* (59)*.*

Then, the expected variational bound (56) *over the data distribution is minimized if and only if*

$$\mathbf{m}^*_\theta(\mathbf{x}_T, \mathbf{x}_T, T-1) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_T], \quad \mathbf{m}^*_\theta(\mathbf{x}_t, \mathbf{x}_T, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{x}_T], \qquad t = 1, \ldots, T-1. \tag{60}$$

The proof of Proposition B.4 is deferred to Appendix F.3.3. It is important to note that at the terminal step $t = T$, we cannot directly construct an estimator that recovers $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_T]$, as this conditional depends on the unknown data distribution we aim to model. Instead, we approximate $\mathbf{x}_{T-1}$ with $\mathbf{x}_T$ and reuse the estimator for the penultimate step, which is optimal for $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_{T-1}, \mathbf{x}_T]$.

Conditional Estimator. Let $\boldsymbol{\epsilon}_{\theta_{\text{cond}}}(\mathbf{x}_t, \mathbf{x}_T, t)$ denote a neural network predicting the Gaussian noise in the bridge FDP marginal (52) that depends on parameters θ_{cond}. Given a predicted noise, the conditional estimator of $\mathbf{x}_0$ is obtained by inverting the linear Gaussian decomposition:

$$\hat{\mathbf{x}}_0^{\text{cond}}(\mathbf{x}_t, \mathbf{x}_T, t) := \frac{\mathbf{x}_t - C^{(T)}_{t|0,T}\mathbf{x}_T - \sigma_{t|0,T}\,\boldsymbol{\epsilon}_{\theta_{\text{cond}}}(\mathbf{x}_t, \mathbf{x}_T, t)}{C^{(0)}_{t|0,T}}. \tag{61}$$

The corresponding noise-prediction objective, averaging over timesteps t and data samples $(\mathbf{x}_0, \mathbf{x}_T)$, is:

$$\begin{aligned} &\mathcal{J}_{\text{B-cond}}(\theta_{\text{cond}}) = \mathbb{E}_{t\sim\text{Unif}\{1,\ldots,T-1\},\,(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T),\,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}\left[\mathcal{L}_{\text{B-cond}}(\theta_{\text{cond}}, \mathbf{x}_0, \mathbf{x}_T, \boldsymbol{\epsilon}, t)\right] \\ &\mathcal{L}_{\text{B-cond}}(\theta_{\text{cond}}, \mathbf{x}_0, \mathbf{x}_T, \boldsymbol{\epsilon}, t) = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}\left(C^{(0)}_{t|0,T}\mathbf{x}_0 + C^{(T)}_{t|0,T}\mathbf{x}_T + \sigma_{t|0,T}\boldsymbol{\epsilon}, \mathbf{x}_T, t\right)\right\|^2. \end{aligned} \tag{62}$$

The following proposition shows that minimizing the noise-prediction objective (62) yields an estimator that achieves the variational bound minimum established in Proposition B.4.

Proposition B.5 (Conditional Estimator as Posterior Mean for Bridge DDPM)**.** *For fixed* $t = 1, \ldots, T-1$*, let* $\boldsymbol{\epsilon}^*_{\theta_{cond}}(\mathbf{x}_t, \mathbf{x}_T, t)$ *be the minimizer of* (62)*. Then, in the infinite-data and infinite-capacity limit,*

$$\boldsymbol{\epsilon}^*_{\theta_{cond}}(\mathbf{x}_t, \mathbf{x}_T, t) = \frac{\mathbf{x}_t - C^{(0)}_{t|0,T}\,\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{x}_T] - C^{(T)}_{t|0,T}\mathbf{x}_T}{\sigma_{t|0,T}}. \tag{63}$$

Consequently, the induced estimator (61)

$$\hat{\mathbf{x}}_0^{cond,\star}(\mathbf{x}_t, \mathbf{x}_T, t) = \frac{\mathbf{x}_t - C^{(T)}_{t|0,T}\mathbf{x}_T - \sigma_{t|0,T}\,\boldsymbol{\epsilon}^*_{\theta_{cond}}(\mathbf{x}_t, \mathbf{x}_T, t)}{C^{(0)}_{t|0,T}} \tag{64}$$

recovers the posterior mean,

$$\hat{\mathbf{x}}_0^{cond,*}(\mathbf{x}_t, \mathbf{x}_T, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{x}_T]. \tag{65}$$

The proof of Proposition B.5 is deferred to Appendix F.3.4 The same remark regarding the instability of the conditional estimator as $t \to T$ applies here as well; see Section 3.5.

Marginal Estimator. To address the numerical instability of the conditional estimator, we introduce a marginal estimator that does not explicitly condition on $\mathbf{x}_T$. We express $\mathbf{x}_t$ in the DDPM-style form

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\eta}_t, \tag{66}$$

where $\boldsymbol{\eta}_t$ is a residual variable whose distribution depends on the bridge construction.

Let $\boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t)$ denote a neural network trained to predict this residual and that depends on parameters θ_{marg}. The marginal estimator is obtained by inverting (66):

$$\hat{\mathbf{x}}_0^{\text{marg}}(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}. \tag{67}$$

The network is trained with the objective

$$\begin{aligned}
\mathcal{J}_{\text{B-marg}}(\theta_{\text{marg}}) &= \mathbb{E}_{t\sim\text{Unif}\{1,\dots,T-1\},\,(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T),\,\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)}\Big[\mathcal{L}_{\text{B-marg}}(\theta_{\text{marg}}, \mathbf{x}_0, \mathbf{x}_t, t)\Big], \\
\mathcal{L}_{\text{B-marg}}(\theta_{\text{marg}}, \mathbf{x}_0, \mathbf{x}_t, t) &= \left\| \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t) \right\|^2.
\end{aligned} \tag{68}$$

The loss in (68) does not condition on $\mathbf{x}_T$, so the marginal estimator averages over the possible terminal states $\mathbf{x}_T$ and thus recovers the marginal posterior mean as shown in the next proposition.

Proposition B.6 (Marginal Posterior Mean Estimator for Bridge DDBM)**.** *For fixed $t = 1, \dots, T-1$, let $\boldsymbol{\eta}^*_{\theta_{marg}}(\mathbf{x}_t, t)$ be the minimizer of* (68)*. Then, in the infinite-data and infinite-capacity limit,*

$$\boldsymbol{\eta}^*_{\theta_{marg}}(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbb{E}_{marg}[\mathbf{x}_0 \mid \mathbf{x}_t]}{\sqrt{1-\bar{\alpha}_t}}. \tag{69}$$

Consequently, the induced estimator (67)

$$\hat{\mathbf{x}}_0^{marg,*}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\Big(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\eta}_\theta^{marg,*}(\mathbf{x}_t, t)\Big) \tag{70}$$

recovers the marginal posterior mean,

$$\hat{\mathbf{x}}_0^{marg,*}(\mathbf{x}_t, t) = \mathbb{E}_{marg}[\mathbf{x}_0 \mid \mathbf{x}_t]. \tag{71}$$

The proof of Proposition B.6 is deferred to Appendix F.3.5.

The same remark regarding the stability of the marginal estimator as $t \to T$ applies here as well; see Section 3.5.

Comparison with prior work. To the best of our knowledge, prior work on DDPM style diffusion bridges has not employed the conditional estimator. In particular, existing approaches such as I^{2}SB [16] rely only on the marginal estimator. In the multi-bridge setting (Section 3.5), we introduce a more efficient hybrid estimator, which, to our knowledge, has not been explored in previous work.

Hybrid Estimator. To combine the statistical efficiency of the conditional estimator with the numerical stability of the marginal estimator, we can also introduce a hybrid estimator:

$$\hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_t, \mathbf{x}_T, t) := (1-\lambda_t)\,\hat{\mathbf{x}}_0^{\text{marg}}(\mathbf{x}_t, t) + \lambda_t\,\hat{\mathbf{x}}_0^{\text{cond}}(\mathbf{x}_t, \mathbf{x}_T, t), \tag{72}$$

where $\theta = (\theta_{\text{cond}}, \theta_{\text{marg}})$, $\hat{\mathbf{x}}_0^{\text{marg}}(\mathbf{x}_t, t)$ is defined in (67) and $\hat{\mathbf{x}}_0^{\text{cond}}(\mathbf{x}_t, \mathbf{x}_T, t)$ is defined in (61). The weight $\lambda_t \in [0, 1]$ depends on the conditional SNR (26) and is defined as in (32). The same remark regarding the stability and efficiency of the hybrid estimator applies here as well; see Section 3.5.

B.6 Training and Sampling Algorithms

Algorithm 6 Training Conditional Estimator

repeat
  $(\mathbf{x}_0, \mathbf{x}_T) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T)$
  $t \sim \text{Uniform}\{1, \dots, T-1\}$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
  $\mathbf{x}_t = C^{(0)}_{t|0,T}\mathbf{x}_0 + C^{(T)}_{t|0,T}\mathbf{x}_T + \sigma_{t|0,T}\boldsymbol{\epsilon}$
  Gradient step on $\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}(\mathbf{x}_t, t)\|^2$
until converged

Algorithm 7 Training Marginal Estimator

repeat
  Same as Algorithm 6, 2-5
  $\boldsymbol{\eta}_t = (\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)/\sqrt{1-\bar{\alpha}_t}$
  Gradient step on $\|\boldsymbol{\eta}_t - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t)\|^2$
until converged

Algorithm 8 Sampling

$\mathbf{x}_T \sim p_{\text{data}}(\mathbf{x}_T)$
for $t = T-1, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
  if $t = T$ **then**
    $\mathbf{x}_{T-1} = \boldsymbol{\mu}_{T-1|0,T}(\hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_T, \mathbf{x}_T, T-1), \mathbf{x}_T) + \sigma_{T-1|0,T}\mathbf{z}$
  else if $t = T-1, \dots, 2$ **then**
    $\mathbf{x}_{t-1} = \boldsymbol{\mu}_{t-1|0,t}(\hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_t, \mathbf{x}_T, t), \mathbf{x}_t) + \sigma_{t-1|0,t}\mathbf{z}$
  else if $T = 1$ **then**
    $\mathbf{x}_0 = \hat{\mathbf{x}}_0^{\text{hybrid}}(\mathbf{x}_1, \mathbf{x}_T, 1) + \sigma_{\text{rec}}\mathbf{z}$
  end if
end for
return $\mathbf{x}_0$

C Related Work

C.1 Diffusion models for inverse problems

Score-Based Diffusion Models. Score-based diffusion models define the forward noising process via an Itô stochastic differential equation (SDE). Specifically, we focus on the Variance-Preserving (VP) formulation of the SDE presented by [29], which corresponds to the DDPM [12] (Section 2.1 and Appendix A). This framework constructs a forward diffusion process $(\mathbf{x}_t)_{t\in[0,T]}$, with $\mathbf{x}_t \in \mathbb{R}^d$, governed by the following equation:

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0),$$

where $\mathbf{w}_t$ denotes a standard Wiener process and $\beta(t) : \mathbb{R} \to \mathbb{R}_+$ is a continuous noise schedule, which can be viewed as the continuous-time analogue of the discrete β_t used in DDPMs. The forward process is associated with the transition kernel:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}(t)}\mathbf{x}_0, (1-\bar{\alpha}(t))\,\mathbf{I}_d), \tag{73}$$

where $\bar{\alpha}(t) := \exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right)$.

To recover the data-generating distribution, the noising process is reversed by solving the reverse SDE [2]:

$$\mathrm{d}\mathbf{x}_t = -\beta(t)\left(\frac{1}{2}\mathbf{x}_t + \nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t)\right)\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\bar{\mathbf{w}}_t, \quad \mathbf{x}_T \sim p(\mathbf{x}_T),$$

where $\mathrm{d}t$ corresponds to time running backward, $\mathrm{d}\bar{\mathbf{w}}_t$ to the standard Wiener process running backward and $p(\mathbf{x}_T) = \mathcal{N}(0, \mathbf{I}_d)$. Just as a word of caution, here $\mathbf{x}_T$ represents the end of the diffusion process in standard (non-bridge) DDPM as presented in Section 2.1 and Appendix A.

The term $\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t)$ is the score function, which guides the reverse process and it is typically approximated using a neural network $\mathbf{s}_{\boldsymbol{\theta}}$, trained via Denoising Score Matching (DSM) [34] using the objective

$$\mathcal{J}_{\text{DSM}}(\boldsymbol{\theta}) = \mathbb{E}_{t\sim U(\epsilon,T),\mathbf{x}_0\sim p_{\text{data}}(\mathbf{x}_0),\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0)}\Big[\lambda(t)\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta},\mathbf{x}_0,\mathbf{x}_t,t)\Big]\Big], \tag{74}$$

with

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\theta},\mathbf{x}_0,\mathbf{x}_t,t) = \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t|\mathbf{x}_0)\|^2,$$

where from (73)

$$\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}(t)}\mathbf{x}_0}{1-\bar{\alpha}(t)}. \tag{75}$$

and where $\epsilon \approx 0$ is a small positive constant and $\lambda(t) : [0,T] \to \mathbb{R}_+$ is a positive weighting function typically set to $\lambda(t) = 1/\mathbb{E}\left[||\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t|\mathbf{x}_0)||_2^2\right]$ (see [29, Section 3.3]). Once $\boldsymbol{\theta}^*$ is acquired by minimizing (74), one can use the approximation $\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t) \simeq \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_t,t)$.

Inverse Problems. Suppose that $\mathbf{x}_0$ and $\mathbf{y}$ are coupled through a joint distribution $(\mathbf{x}_0,\mathbf{y}) \sim p_{\text{data}}(\mathbf{x}_0,\mathbf{y})$. The objective of inverse problems is to infer unknown $\mathbf{x}_0$ given a set of measurements $\mathbf{y}$, that is to sample from the posterior $p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{y})$.

The score-based diffusion models approaches can be adapted to sample from the posterior by adopting the following reverse process

$$\mathrm{d}\mathbf{x}_t = -\beta(t)\left(\frac{1}{2}\mathbf{x}_t + \nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t \mid \mathbf{y})\right)\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\bar{\mathbf{w}}_t, \quad \mathbf{x}_T \sim p(\mathbf{x}_T), \tag{76}$$

where again $p(\mathbf{x}_T) \sim \mathcal{N}(0,\mathbf{I}_d)$. The reverse-time SDE defines a stochastic trajectory from the prior distribution at time T to the desired posterior distribution at time 0.

In practice, sampling is performed by initializing $\mathbf{x}_T$ with Gaussian noise and numerically solving the reverse SDE from $t = T$ to $t = 0$. Therefore, generating samples from the posterior distribution requires estimating the posterior score $\nabla\mathbf{x}_t \log q(\mathbf{x}_t \mid \mathbf{y})$, which guides the reverse diffusion process towards regions of high posterior probability.

C.2 Diffusion Posterior Sampling (DPS)

Diffusion Posterior Sampling (DPS) [6] uses the framework of diffusion models for inverse problems introduced in Appendix C.1.

Suppose that the likelihood distribution $p_{\mathrm{data}}(\mathbf{y} \mid \mathbf{x}_0)$, which describes the observation process, is known. Specifically, the observation $\mathbf{y}$ is obtained by applying a degradation operator $\mathcal{A}(\cdot)$ to the clean image $\mathbf{x}_0$, followed by additive Gaussian noise:

$$\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \sigma\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d). \tag{77}$$

The corresponding likelihood is

$$p_{\mathrm{data}}(\mathbf{y} \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{y};\, \mathcal{A}(\mathbf{x}_0),\, \sigma^2\mathbf{I}_d\big)\,. \tag{78}$$

DPS assumes that both the forward degradation operator $\mathcal{A}(\cdot)$ and the associated likelihood model are known. As a result, it is only applicable to inverse problems for which the measurement process can be explicitly specified.

It follows from Bayes' rule that the score of the posterior is

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y}) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(\mathbf{y} \mid \mathbf{x}_t). \tag{79}$$

Hence, computing the score of the posterior distribution can be reduced to evaluating two terms: the prior score function, $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$, and the likelihood score function,$\nabla_{\mathbf{x}_t} \log q(\mathbf{y} \mid \mathbf{x}_t)$. The former can be directly obtained using the trained prior score function $\mathbf{s}_{\theta^*}(\mathbf{x}_t, t)$. However, computing the latter is challenging in closed form due to its dependence on time t, as there is only an explicit dependence between $\mathbf{y}$ and $\mathbf{x}_0$.

To address this, DPS proposes to factorize $q(\mathbf{y} \mid \mathbf{x}_t)$ as:

$$q(\mathbf{y} \mid \mathbf{x}_t) = \int p_{\text{data}}(\mathbf{y} \mid \mathbf{x}_0) q(\mathbf{x}_0 \mid \mathbf{x}_t) \mathrm{d}\mathbf{x}_0, \tag{80}$$

which follows from the fact that $\mathbf{y}$ and $\mathbf{x}_t$ are conditionally independent given $\mathbf{x}_0$. The density $q(\mathbf{x}_0 \mid \mathbf{x}_t)$ is generally intractable, making the approximation of the integral in (80) a challenging task.

DPS first remarked that $\mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0 \mid \mathbf{x}_t)}[\mathbf{x}_0]$ can be obtained with the Tweedie's formula:

$$\mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0 \mid \mathbf{x}_t)}[\mathbf{x}_0] = \frac{1}{\sqrt{\bar{\alpha}(t)}}(\mathbf{x}_t + (1 - \bar{\alpha}(t))\, \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)). \tag{81}$$

By replacing $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$ in (81) with the score estimate $\mathbf{s}_{\theta^*}(\mathbf{x}_t, t)$, the conditional expectation can be approximated with:

$$\mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0 \mid \mathbf{x}_t)}[\mathbf{x}_0] \approx \hat{\mathbf{x}}_0^{\text{tweedie}}(\mathbf{x}_t) := \frac{1}{\sqrt{\bar{\alpha}(t)}}(\mathbf{x}_t + (1 - \bar{\alpha}(t))\, \mathbf{s}_{\theta^*}(\mathbf{x}_t, t)). \tag{82}$$

Then, they proposed to approximate $q(\mathbf{x}_0 \mid \mathbf{x}_t)$ as a Dirac delta distribution centered at the Tweedie's estimate (82) and to approximate the likelihood score function with:

$$q(\mathbf{y} \mid \mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0 \mid \mathbf{x}_t)}[p_{\text{data}}(\mathbf{y} \mid \mathbf{x}_0)] \approx p_{\text{data}}(\mathbf{y} \mid \mathbf{x}_0 = \hat{\mathbf{x}}_0^{\text{tweedie}}(\mathbf{x}_t)). \tag{83}$$

The likelihood score at time t takes the form:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{y} \mid \mathbf{x}_t) \approx -\frac{1}{\sigma^2} \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0^{\text{tweedie}}(\mathbf{x}_t))\|^2. \tag{84}$$

Finally, the posterior score at time t takes the form:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y}) \approx \mathbf{s}_{\theta^*}(\mathbf{x}_t, t) - \rho \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0^{\text{tweedie}}(\mathbf{x}_t))\|^2, \tag{85}$$

where $\rho = 1/\sigma^2$ is the step size that needs to be tuned.

C.3 Conditional Denoising Estimator (CDE)

Conditional Denoising Estimator (CDE) [4] uses the framework of diffusion models for inverse problems introduced in Appendix C.1.

CDE approximates the posterior score function in the form of a neural network [4] using the paired $(\mathbf{x}_0, \mathbf{y})$ during training. The posterior score function is approximated using a neural network $\mathbf{s}_{\boldsymbol{\theta}}$, with learnable parameters $\boldsymbol{\theta}$, using the objective

$$\mathcal{J}_{\text{CDE}}(\boldsymbol{\theta}) = \mathbb{E}_{t\sim U(\epsilon,T),(\mathbf{x}_0,\mathbf{y})\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{y}),\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0)}\Big[\lambda(t)\mathcal{L}_{\text{CDE}}(\boldsymbol{\theta}, \mathbf{x}_0, \mathbf{x}_t, \mathbf{y}, t)\Big]\Big], \tag{86}$$

with

$$\mathcal{L}_{\text{CDE}}(\boldsymbol{\theta}, \mathbf{x}_0, \mathbf{x}_t, \mathbf{y}, t) = \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0)\|^2,$$

and where $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0)$ is shown in (75). Once $\boldsymbol{\theta}^*$ is acquired by minimizing (86), one can use the approximation

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{y}) \simeq \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, \mathbf{y}, t).$$

C.4 Conditional Diffusion Models (CDMs)

Conditional Diffusion Models (CDMs) for image restoration refer to a class of diffusion models that bias the generative process toward recovering a clean image from degraded measurements $\mathbf{y}$. Works that used CDMs include [25, 24, 32, 37].

CDMs assume the same FDP as a standard DDPM as described in Appendix A.1, where the forward marginal distribution at timestep t is repeated here for convenience:

$$\mathbf{x}_t \mid \mathbf{x}_0 = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d).$$

Given a condition $\mathbf{y} \sim p(\mathbf{y})$ and clean data $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{y})$, the standard conditional noise-prediction objective is:

$$\mathcal{J}_{\text{Cond-DDPM}}(\theta) = \mathbb{E}_{\substack{t\sim\text{Unif}\{1,\dots,T\},\\ \mathbf{y}\sim p(\mathbf{y}),\mathbf{x}_0\sim p_{\text{data}}(\mathbf{x}_0|\mathbf{y}),\\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\big(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \mathbf{y}, t\big)\right\|^2\right]. \tag{87}$$

The same sampling produced as standard DDPM (Algorithm 5) is used except that the target conditioned on $\mathbf{y}$ is estimated with:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, \mathbf{y}, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{y}, t)}{\sqrt{\bar{\alpha}_t}}.$$

There is an equivalence between CDM (score-based) and CDE (noise-prediction-based), analogous to the equivalence between score-based generative models [29] and DDPMs [12]. Specifically, the score prediction and noise prediction parameterizations can be converted into one another, allowing either formulation to represent the same underlying diffusion process.

The main difference between CDM and CDE lies in the sampling procedure. While CDE uses the standard DDPM sampling procedure, CDM relies on solving the corresponding reverse-time SDE using an SDE solver.

C.5 Image-to-Image Schrodinger Bridge (I^{2}SB)

The Image-to-Image Schrodinger Bridge (I^{2}SB) [16] model is equivalent to the bridge model described in Appendix B, but it uses only the marginal estimator. Consequently, training is required solely for the marginal estimator (Algorithm 7), while sampling proceeds as described in Algorithm 8 with $\lambda_t = 0, \forall t$.

C.6 Limitations of related work

Limitation of score-based methods for inverse problems (DPS, CDE) and conditional diffusion models (CDM). In score-based methods for inverse problems (DPS [6], CDE [4]) and conditional diffusion models (CDM [37]), the joint distribution is factorized as

$$q(\mathbf{x}_0, \mathbf{x}_t, \mathbf{y}) = p_{\text{data}}(\mathbf{x}_0, \mathbf{y})q(\mathbf{x}_t \mid \mathbf{x}_0), \tag{88}$$

which relies on the conditional independence assumption $\mathbf{x}_t \perp \mathbf{y} \mid \mathbf{x}_0$ ([6, Equation (7)], [4, Proof of Corollary 1]). Under this assumption, the FDP is independent of the observation $\mathbf{y}$ and progressively removes information from $\mathbf{x}_0$ until reaching the noise distribution. Observation-dependent information is introduced only during the reverse process through the conditional score or likelihood guidance. Therefore, the diffusion process does not define a cycle between the target and the observation: starting from $\mathbf{x}_T$, the reverse process can generate a sample of $\mathbf{x}_0$ conditioned on $\mathbf{y}$, but the observation $\mathbf{y}$ cannot be recovered by applying the forward diffusion process from $\mathbf{x}_0$, as it was not part of the forward transition.

In contrast, diffusion bridge models (e.g., I^2SB and MB-DDPM) define the forward process conditioned on both endpoints which defines the joint distribution:

$$q(\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T) = p_{\mathrm{data}}(\mathbf{x}_0, \mathbf{x}_T) q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T). \tag{89}$$

By conditioning the forward transition on the observation endpoint $\mathbf{x}_T$, the intermediate states $\mathbf{x}_t$ preserve information relevant to both endpoints throughout the diffusion trajectory. As a result, the forward and reverse processes form a consistent bridge between $\mathbf{x}_0$ and $\mathbf{x}_T$, enabling a cyclical interpretation in which one can transition between the two endpoints through the learned bridge dynamics.

This interpretation is related to the broader notion of cycle consistency, introduced in CycleGAN [40], where consistency is enforced by requiring transformations between two domains to remain coherent when composed in both directions. In MB-DDPM, the cyclical property arises from the endpoint-conditioned bridge formulation rather than from an explicit cycle-consistency loss. Thus, endpoint-to-endpoint consistency can be viewed as a fundamental structural property of diffusion bridge models.

Limitations of diffusion bridge models (I^2SB). Diffusion bridge models (I^2SB [16]) are inherently restricted to one-to-one settings, i.e., one target image and one source image. Multi-source inference is therefore handled indirectly via heuristics such as selecting the best source or aggregating independent reconstructions, which either discard complementary information or fail to integrate it coherently.

In contrast, MB-DDPM directly extends diffusion bridges to the multi-source setting by jointly modeling multiple stochastic bridges within a unified reverse process. This enables explicit interaction between sources during sampling and more effective aggregation of complementary information.

D Implementation details

D.1 Data

CIFAR-10. We use the full CIFAR-10 dataset [13], consisting of 50,000 training images and 10,000 test images at a resolution of 32×32.

ImageNet. We use a balanced subset of the ImageNet dataset [8] , consisting of 50,000 training images and 10,000 test images, all center-cropped to a resolution of 128×128. To ensure class balance, we restrict the dataset to at most 100 classes, with up to 600 images per class.

MRI. The brain MRI dataset is derived from a subset of clinical trial data (ClinicalTrials.gov identifier: NCT02792231, sponsor ID: COMB157G2302) included in the Novartis-Oxford Multiple Sclerosis (NO.MS) database [7]. The resulting dataset comprises 353 scans from 297 unique patients. Each patient has between 1 and 3 scans (mean: 1.368, median: 1), with each scan producing a 3D MRI volume composed of 40 to 51 axial brain slices (mean: 46.94, median: 47). Each slice is resized to 168×208 to remove surrounding black borders. We treat each slice as an independent grayscale image of size $1 \times 168 \times 208$, enabling 2D image-based modeling of the volumetric data. To prevent data leakage, we perform a patient-level split, ensuring that all scans from a given patient are assigned exclusively to either the training or test set. The resulting training set consists of 235 scans from 213 patients, comprising 10,984 slices. The test set contains 118 scans from 113 patients, comprising 5,492 slices.

D.2 Diffusion schedule

We use a diffusion process with $T = 1000$ timesteps. The variance schedule $\{\beta_t\}_{t=1}^{T}$ is constructed using a smooth quadratic interpolation between a small initial noise level and a larger final noise level similar to [16] (see Figure A1). Specifically, we define

$$\beta_t = \left((1-\tau)\sqrt{\beta_{\min}} + \tau\sqrt{\beta_{\max}}\right)^2, \quad \tau = \frac{t}{T},$$

where $\beta_{\min} = 10^{-4}$ and $\beta_{\max} = 0.3/T$.

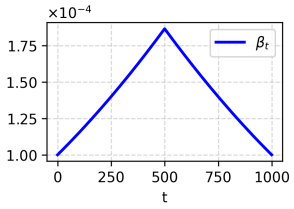


Figure A1: Symmetric noise scheduling

D.3 Training set-up

Architecture. We employ a U-Net architecture similar to those used in diffusion models [12]. The network follows an encoder–decoder structure with skip connections, enabling multi-scale feature aggregation. The input and output are defined by `in_channels` and `out_channels`. Let K denote the number of sources and C the number of image channels. For the conditional estimator network $\boldsymbol{\epsilon}_{\theta_{\text{cond}}}$, the number of input channels is given by $\texttt{in_channels} = 2 \times K \times C$. For the marginal estimator network $\boldsymbol{\eta}_{\theta_{\text{marg}}}$, the input dimensionality is $\texttt{in_channels} = K \times C$. For both networks, the number of output channels is defined as $\texttt{out_channels} = K \times C$.

The network uses a base channel width of $C^{\text{U-net}} = 256$, with feature dimensions scaled across resolution levels using channel multipliers $(1, 2, 4, 8)$. We use 2 residual blocks per resolution level, and apply self-attention at spatial resolution 8 [33].

The diffusion timestep is encoded using sinusoidal embeddings, consisting of sine and cosine functions evaluated at exponentially spaced frequencies. Given a timestep t, the resulting components are concatenated to form an embedding of dimension $C^{\text{U-net}}$. This embedding is subsequently passed through a two-layer multilayer perceptron (MLP), with a SiLU activation between the linear layers, to obtain a projected timestep embedding of dimension $4C^{\text{U-net}}$. This embedding is then provided to the residual blocks throughout the U-Net encoder, bottleneck, and decoder, conditioning the network's denoising operations on the current diffusion timestep.

For the CIFAR-10 experiment, $K = 4$, $C = 3$ and the resulting number of parameters of the conditional estimator network $\boldsymbol{\epsilon}_{\theta_{\text{cond}}}$ and the marginal estimator network $\boldsymbol{\eta}_{\theta_{\text{marg}}}$ are 988.514316 and 988.486668, respectively. For the ImageNet experiment, $K = 4$, $C = 3$ and the resulting number of parameters of the conditional estimator network $\boldsymbol{\epsilon}_{\theta_{\text{cond}}}$ and the marginal estimator network $\boldsymbol{\eta}_{\theta_{\text{marg}}}$ are

988.514316 and 988.486668, respectively. For the MRI experiment, $K = 3$, $C = 1$ and the resulting number of parameters of the conditional estimator network $\boldsymbol{\epsilon}_{\theta_{\text{cond}}}$ and the marginal estimator network $\boldsymbol{\eta}_{\theta_{\text{marg}}}$ are 988.452099 and 988.445187, respectively.

Hyperparameters. Training was performed using a batch size of 512 for CIFAR-10, 128 for ImageNet and 40 for MRI, with a learning rate of 5×10^{-5}. The models were trained for 200 epochs.

Optimizer. We use an exponential moving average (EMA) of the parameters with decay 0.99. Optimization is performed using AdamW. The learning rate schedule follows a cosine annealing strategy.

Compute ressources. We train MB-DDPM using three NVIDIA A100-SXM4-80GB GPUs, and distribute sampling across the three devices by splitting batches over the GPUs.

Running times. Running times across experiments are reported in Tables A2 and A7– A9.

Increasing image size substantially increases computational and memory requirements, as larger images lead to a greater number of feature activations and spatial operations. This effect is reflected in the runtime differences across CIFAR-10 (32×32), ImageNet (128×128), and MRI (168×208), as reported in Tables A7- A9.

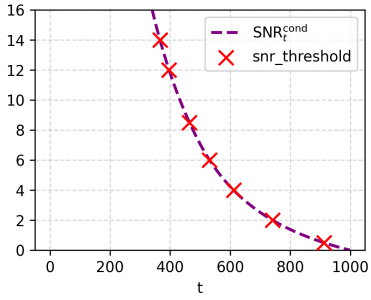
However, as observed in Table A2, increasing the number of sources incurs only a limited runtime overhead.

Code availability. The code is available in the GitHub repository `https://github.com/NeuroSML/mb-ddpm`.

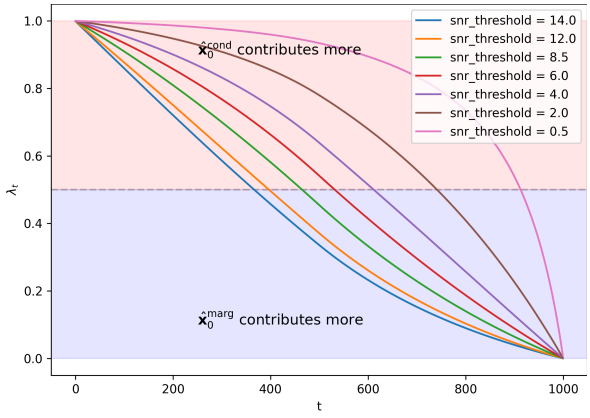
D.4 Tuning `snr_threshold`

The parameter `snr_threshold` is a tunable constant that controls the scale of the interpolation. The weighting coefficient λ_t is small near $t \approx T$, reflecting low confidence in the conditional estimator, and increases as t decreases.

To select `snr_threshold` as a function of t, observe that when `snr_threshold` $\approx \text{SNR}_t^{\text{cond}}$, the two estimators contribute equally, yielding $\lambda_t = 0.5$. While $\text{SNR}_t^{\text{cond}}$ was previously shown on a logarithmic scale in Figure 4, we present it in Figure A2a on the original scale for clarity.



(a) Signal-to-noise ratio of the conditional estimator ($\text{SNR}_t^{\text{cond}}$) as a function of t. The values of `snr_threshold` considered are overlaid with a red cross. For each threshold value, the corresponding intersection with $\text{SNR}_t^{\text{cond}}$ identifies the point at which the marginal and conditional estimators contribute equally.



(b) Weighting coefficient of the conditional estimator, λ_t (32), as a function of t for the selected values of `snr_threshold`.

Figure A2: SNR and weights in the hybrid estimator of the conditional estimator as a function of `snr_threshold`.

Our goal is to identify a range of values where assigning a weight of 0.5 to the conditional estimator is appropriate; that is, where $\mathrm{SNR}_t^{\mathrm{cond}}$ is not too small. Accordingly, we consider a grid of values for `snr_threshold` = $14, 12, 8.5, 6, 4, 2, 0.5$, which correspond to time steps $t = 366,\ 396,\ 464,\ 532,\ 612,\ 741,\ 913$, respectively, such that $\lambda_t = 0.5$ as shown in Figure A2b.

Figures A3, A4 and A5 report the evaluation metrics across different values of `snr_threshold` on the CIFAR-10 test set (10k samples), ImageNet test set (10k samples) and MRI test set (5,492 samples), respectively. We observe that the FID consistently deteriorates as `snr_threshold` increases, indicating that leveraging more the conditional estimate improves the perceptual quality of the generated samples. In contrast, distortion-based metrics such as MSE, PSNR, and SSIM begin to degrade when `snr_threshold` becomes too small. This suggests increased instability in the reconstruction process, where the model tends to rely more heavily on the conditional estimator for later stage, leading to poorer fidelity. This behavior can be attributed to the reduced contribution of reliable marginal information at low `snr_threshold` values.

Based on these observations, we select `snr_threshold` $= 6$ for CIFAR-10 and `snr_threshold` $=$ 2 for ImageNet and MRI, as these values provide a suitable trade-off between perceptual quality and reconstruction fidelity.

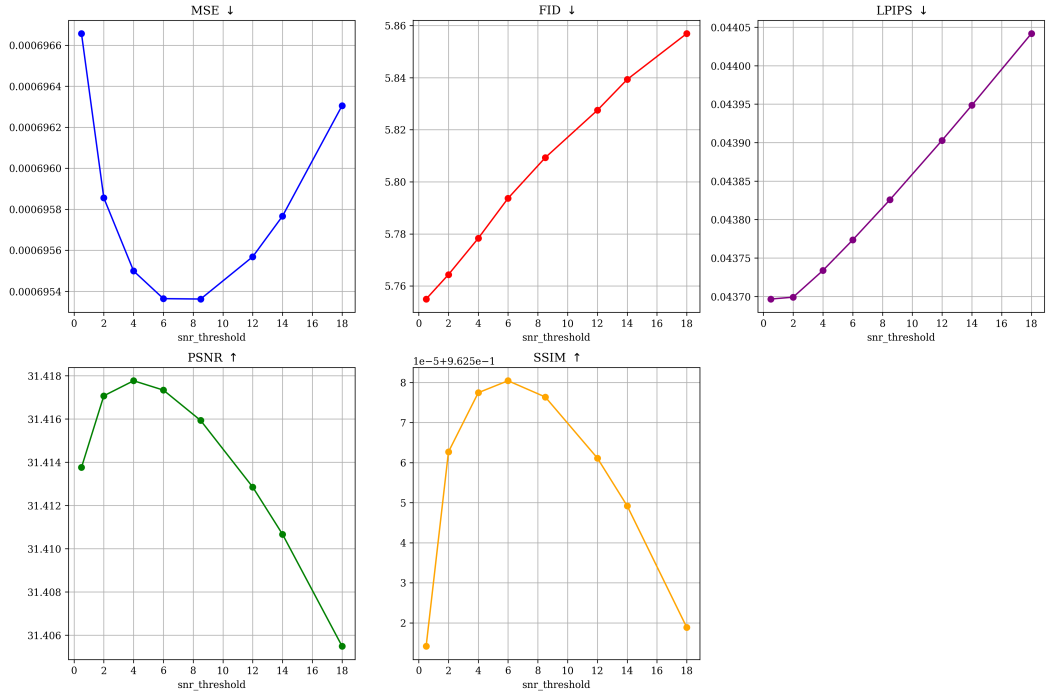


Figure A3: MSE, PSNR, SSIM, FID and LPIPS evaluated on CIFAR-10 test images (10k) predicted with MB-DDPM using four degradation sources (Gaussian blur, JPEG compression, $\times 4$ bicubic super-resolution, and Gaussian noise), shown for different values of `snr_threshold`.

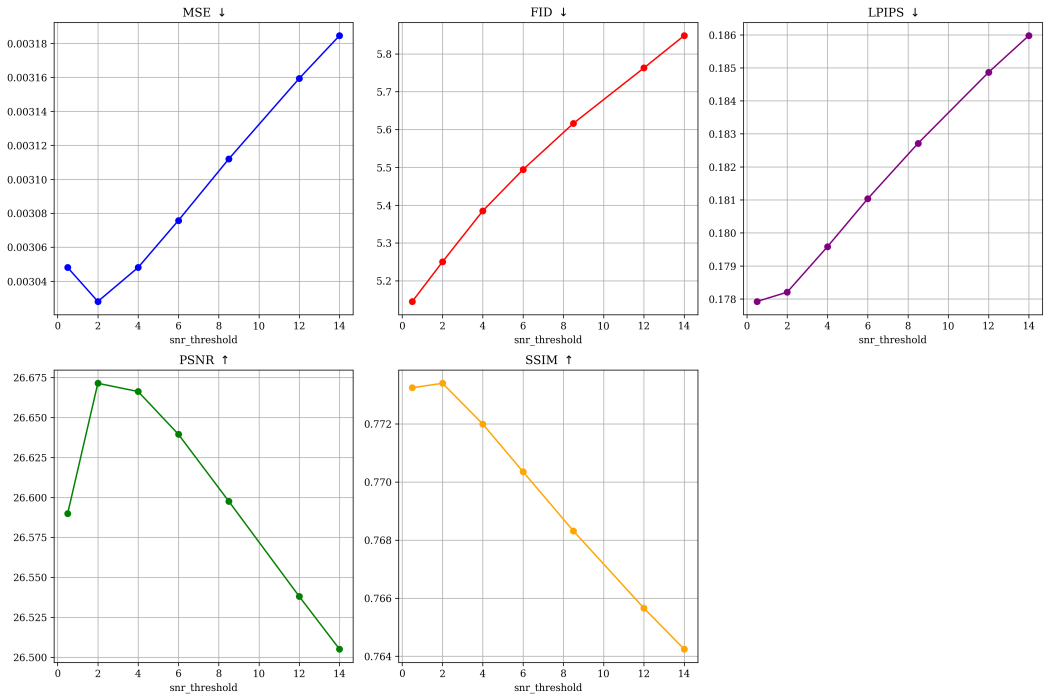


Figure A4: MSE, PSNR, SSIM, FID and LPIPS evaluated on ImageNet test images (10k) predicted with MB-DDPM using four degradation sources (Gaussian blur, JPEG compression, $\times 4$ bicubic super-resolution, and Gaussian noise), shown for different values of `snr_threshold`.

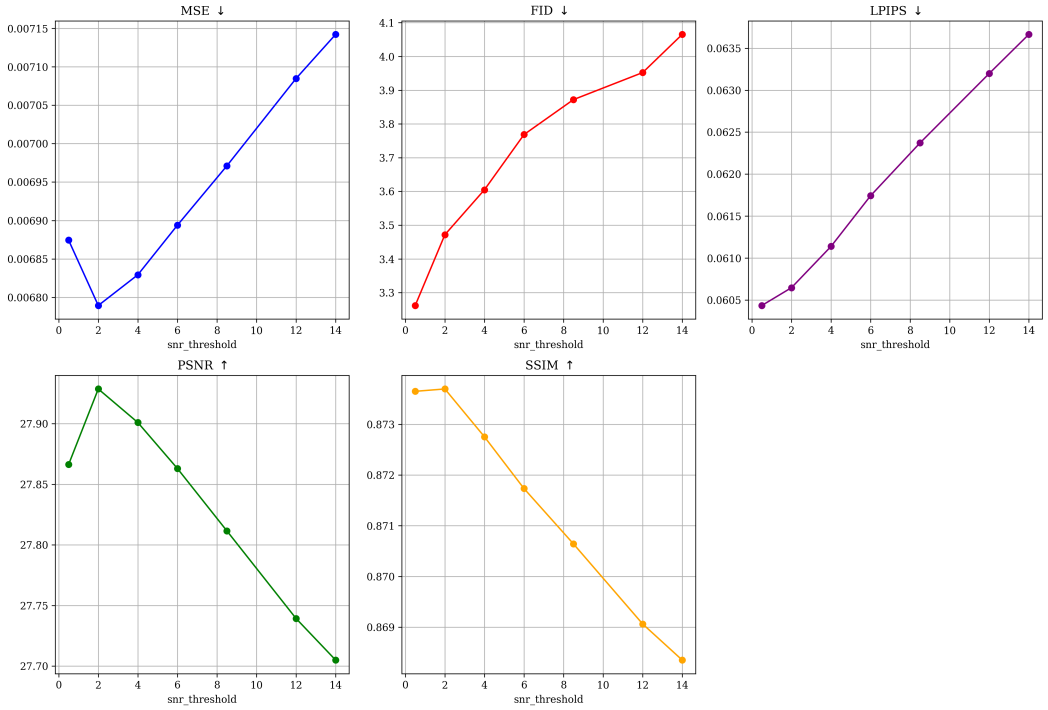


Figure A5: MSE, PSNR, SSIM, FID and LPIPS evaluated on MRI test set (5,492 samples) predicted with MB-DDPM using three source contrasts (T1, T2 and PD), shown for different values of `snr_threshold`.

D.5 Evaluation Metrics

Distortion-based metrics. We report Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM). MSE measures the average squared difference between the predicted image and the ground truth. PSNR is a logarithmic transformation of MSE that is more aligned with human perception of reconstruction fidelity. SSIM [36] evaluates structural similarity by comparing local luminance, contrast, and texture patterns, and is commonly used to assess perceptual image quality.

Perceptual metrics. To evaluate perceptual realism, we report Fréchet Inception Distance (FID) [11] and Learned Perceptual Image Patch Similarity (LPIPS) [38]. FID measures the distance between feature distributions of generated and real images using a pretrained Inception network, capturing both quality and diversity at the dataset level. LPIPS computes a perceptual similarity score based on deep features extracted from pretrained networks, providing a measure of visual similarity that correlates well with human judgment.

ICC. The Intraclass Correlation Coefficients (ICC) is a common measure to evaluate the performance of lesion volume predictor. Let y_s denote the true T2 lesion volume and $\hat{y}_s$ the predicted T2 lesion volume for scan $s = 1, \ldots, S$, with $k = 2$ methods. Define the grand mean, scan means, and method means as

$$\bar{y} = \frac{1}{Sk} \sum_{s=1}^{S} (y_s + \hat{y}_s), \quad \bar{y}_{s\cdot} = \frac{1}{2} (y_s + \hat{y}_s), \quad \bar{y}_{\cdot 1} = \frac{1}{S} \sum_{s=1}^{S} y_s, \quad \bar{y}_{\cdot 2} = \frac{1}{S} \sum_{s=1}^{S} \hat{y}_s.$$

The total sum of squares is decomposed into between-scan, between-method, and residual (error) components:

$$SS_{\text{total}} = \sum_{s=1}^{S} \left[(y_s - \bar{y})^2 + (\hat{y}_s - \bar{y})^2 \right],$$

$$SS_{\text{scans}} = k \sum_{s=1}^{S} (\bar{y}_{s\cdot} - \bar{y})^2, \quad SS_{\text{methods}} = S \sum_{j=1}^{2} (\bar{y}_{\cdot j} - \bar{y})^2,$$

$$SS_{\text{error}} = SS_{\text{total}} - SS_{\text{scans}} - SS_{\text{methods}}.$$

Mean squares are then computed as

$$MS_{\text{scans}} = \frac{SS_{\text{scans}}}{S-1}, \quad MS_{\text{methods}} = \frac{SS_{\text{methods}}}{k-1}, \quad MS_{\text{error}} = \frac{SS_{\text{error}}}{(S-1)(k-1)}.$$

The Intraclass Correlation Coefficient (ICC), corresponding to a two-way random-effects model for absolute agreement, is given by

$$\text{ICC} = \frac{MS_{\text{scans}} - MS_{\text{error}}}{MS_{\text{scans}} + (k-1)MS_{\text{error}} + \frac{k}{S}(MS_{\text{methods}} - MS_{\text{error}})}.$$

In this formulation, the two methods correspond to the true (y_s) and predicted ($\hat{y}_s$) T2 lesion volumes, and the ICC quantifies their agreement across scans while accounting for both scan-level variability and systematic differences between the two measurement methods.

Values of the ICC range from 0 to 1, where higher values indicate stronger agreement or reliability.

Pearson correlation coefficient. The Pearson correlation coefficient is another measure to evaluate the performance of lesion volume predictor. The Pearson correlation coefficient computed between predicted values $\hat{y}$ and true values y measures the strength and direction of the linear relationship between them. It is defined as the covariance of $\hat{y}$ and y divided by the product of their standard deviations. This coefficient ranges from -1 to 1, where 1 indicates perfect positive linear correlation, -1 indicates perfect negative correlation, and 0 indicates no linear relationship.

D.6 Best-source and fusion

To evaluate the effect of source multiplicity (Section 4.1, Appendix E.1.1), and to enable comparison with I^2SB, we introduce the notions of best-source and fusion.

Best-source. The best-source setting considers models trained on individual source–target pairs. For a given sample, the best source is defined as the source whose corresponding prediction yields the lowest FID. Formally, let $\hat{\mathbf{x}}_0(\mathbf{x}_T^k)$ denote the predicted target obtained from source $\mathbf{x}_T^k$. The best-source is as

$$k^* = \arg\min_{k\in\{1,\dots,K\}} \mathrm{FID}\big(\{\hat{\mathbf{x}}_0(\mathbf{x}_T^{k,(i)})\}_{i=1}^{n_{\text{test}}}, \{\mathbf{x}_0^{(i)}\}_{i=1}^{n_{\text{test}}}\big),$$

and the corresponding best-source prediction is $\hat{\mathbf{x}}_0(\mathbf{x}_T^{k^*})$.

Fusion. Fusion aims to combine predictions obtained from multiple sources. Given a set of sources, we aggregate the corresponding predictions into a single output by averaging. Formally, the fused prediction is defined as

$$\hat{\mathbf{x}}_0^{\text{fusion}} = \frac{1}{K}\sum_{k=1}^{K} \hat{\mathbf{x}}_0(\mathbf{x}_T^k).$$

The resulting fused predictions are treated as the final outputs and are used to compute all downstream evaluation metrics.

D.7 Implementation of baselines

Diffusion Posterior Sampling (DPS). DPS [6] was introduced in Appendix C.2

Remember our data generating process (9) for the target observation $\mathbf{x}_0$ and the source observations $\mathbf{x}_T^{1:K}$. We set $\mathbf{y} := \mathbf{x}_T^{1:K}$. Therefore, under DPS framework, the likelihood is set to

$$p_{\text{data}}(\mathbf{x}_T^{1:K} \mid \mathbf{x}_0) = \prod_{k=1}^{K} \frac{1}{(2\pi\sigma_k)^d} \exp\left(-\frac{1}{2\sigma_k^2}\|\mathbf{x}_T^k - \mathcal{A}(\mathbf{x}_0)\|^2\right). \tag{90}$$

The likelihood score at time t takes the form:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_T^{1:K} \mid \mathbf{x}_t) \approx -\sum_{k=1}^{K} \frac{1}{\sigma_k^2} \nabla_{\mathbf{x}_t} \|\mathbf{x}_T^k - \mathcal{A}(\hat{\mathbf{x}}_0^{\text{tweedie}}(\mathbf{x}_t))\|^2. \tag{91}$$

Finally, the posterior score at time t takes the form:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_T^{1:K}) \approx \mathbf{s}_{\theta^*}(\mathbf{x}_t, t) - \sum_{k=1}^{K} \rho_k \nabla_{\mathbf{x}_t} \|\mathbf{x}_T^k - \mathcal{A}(\hat{\mathbf{x}}_0^{\text{tweedie}}(\mathbf{x}_t))\|^2, \tag{92}$$

where $\rho_k = 1/\sigma_k$ is the step-size.

To train the unconditional score model $\mathbf{s}_\theta(\mathbf{x}_t, t)$ (74), we use the same diffusion schedule as in [12]. Specifically, we set $T = 1000$ and define the forward process variances $\{\beta_t\}_{t=1}^T$ using a linear schedule, with values increasing from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. Furthermore, we use the same training setup as for MB-DDPM, including the network architecture, optimization procedure, and hyperparameters, as described in Appendix D.3.

We only apply DPS to the CIFAR-10 and ImageNet experiments, where the degradation operators $\mathcal{A}(\cdot)$ are known. Specifically, these experiments involve controlled degradation processes, including clean image $\to$ JPEG compression, clean image $\to$ Gaussian blur, clean image $\to$ Gaussian noise, and clean image $\to$ super-resolution. In contrast, we do not apply DPS to the MRI experiments, as the underlying degradation operators are unknown for the considered transformations (FLAIR $\to$ T1, FLAIR $\to$ T2, and FLAIR $\to$ PD).

Compared with MB-DDPM, DPS is more challenging to tune, as it requires selecting a separate hyperparameter ρ_k for each source. These parameters control the relative contribution of each source during the reconstruction process, whereas MB-DDPM automatically learns how to balance the contribution of different sources during the reverse process (Appendix E.1.3). We obtained the best performance using $\rho_k = 0.3$ for the JPEG compression and Gaussian blur sources, and $\rho_k = 0.2$ for the Gaussian noise and super-resolution sources.

To sample given the trained unconditional score model, we employed [6, Algorithm 1]. We use the code from [6] available at the GitHub repository `https://github.com/dps2022/diffusion-posterior-sampling` under the Creative Commons Attribution-ShareAlike 4.0 International License.

Conditional Denoising Estimator (CDE). CDE [4] was introduced in Appendix C.3

We set $\mathbf{y} := \mathbf{x}_T^{1:K}$. The posterior score at time t is approximated with $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_T^{1:K}) \simeq \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, \mathbf{x}_T^{1:K}, t)$.

To train the posterior score function $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{x}_T^{1:K}, t)$ (86), we use the same diffusion schedule as in [12]. Specifically, we set $T = 1000$ and define the forward process variances $\{\beta_t\}_{t=1}^T$ using a linear schedule, with values increasing from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. Furthermore, we use the same training setup as for MB-DDPM, including the network architecture, optimization procedure, and hyperparameters, as described in Appendix D.3.

To sample, we employed the *Predictor-Corrector* (PC) sampling algorithm proposed by [29, Algorithm 1, 3, 5]. The predictor step was implemented using the Euler-Maruyama solver, while the corrector step used the annealed Langevin dynamics. The signal-to-noise ratio was set to $r = 0.6$ after experimentation. We use the code from [29] available in the Github Repository `https://github.com/yang-song/score_sde_pytorch` under the Apache-2.0 license.

Conditional Diffusion Models (CDMs). CDM [37] was introduced in Appendix C.4.

We set $\mathbf{y} := \mathbf{x}_T^{1:K}$ and $p(\mathbf{y})p_{\text{data}}(\mathbf{x}_0 \mid \mathbf{y}) := p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K})$.

To train the conditional noise $\boldsymbol{\epsilon}_\theta\big(\mathbf{x}_t, \mathbf{x}_T^{1:K}, t\big)$ (87), we use the same diffusion schedule as in [12]. Specifically, we set $T = 1000$ and define the forward process variances $\{\beta_t\}_{t=1}^T$ using a linear schedule, with values increasing from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. Furthermore, we use the same training setup as for MB-DDPM, including the network architecture, optimization procedure, and hyperparameters, as described in Appendix D.3.

The sampling procedure is identical to standard DDBM shown in Algorithm 5. We adapted our code to account for the small modifications.

Image-to-Image Schrodinger Bridge ($\mathbf{I^2}$SB). I^2SB [16] was introduced in Appendix C.5.

I^2SB is a bridge model that fixes the diffusion endpoints and is therefore limited to a single source image at a time. We fit a bridge model on each source-target pair $(\mathbf{x}_0, \mathbf{x}_T^k)$ for $k = 1, \ldots, K$, and report both the best-performing single source and the fusion across sources (Appendix D.6).

To train the marginal estimator $\boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t)$ (68), we use the same diffusion schedule as the original method [16], which was also used as the MB-DDPM schedule described in Appendix D.2. Furthermore, we use the same training setup as for MB-DDPM, including the network architecture, optimization procedure, and hyperparameters, as described in Appendix D.3.

We use the code from [16] available at the Github Repository `https://github.com/NVlabs/I2SB` under the "NVIDIA Source Code License for I2SB".

E Additional Results

E.1 Effect of Source Multiplicity

E.1.1 Increasing the number of sources

To further analyze the experiments in Section 4.1, Table A1 reports the quantitative results shown in Figure 5. Figure A6 illustrates qualitative predictions obtained using (from left to right): fusion of MB-DDPMs over 1 source combinations; best-source MB-DDPM across 1 source combinations; MB-DDPM model with four sources.

While the four-source configuration achieves a better performance compared to any single-source or fusion, we observe that one specific three-source combination slightly outperforms the full four-source setup. This result highlights that the choice of source combination can influence performance and motivates evaluating multiple combinations rather than relying solely on the maximum number of available sources. Importantly, the observed difference remains marginal, indicating that performance is generally robust to the inclusion of unreliable sources. We further investigate this in Appendix E.6, where we confirm this observation. Moreover, exploring such multi-source subsets is only possible within a framework such as MB-DDPM.

Overall, evaluating several source combinations can be beneficial when computationally feasible. Performance improves on average with the number of sources, and the four-source setting achieves the best aggregate performance across combinations.

Source Images	**MSE** $\downarrow$	**PSNR** $\uparrow$	**SSIM** $\uparrow$	**FID** $\downarrow$	**LPIPS** $\downarrow$
1 source ($K = 1$)					
Gaussian blur	0.00108	31.330	0.962	6.059	0.045
JPEG compression	0.01521	17.457	0.514	25.448	0.417
x4 bicubic SR	0.00694	21.323	0.714	17.590	0.204
Gaussian noise	0.00548	21.870	0.753	11.926	0.228
Fusion	*0.00338*	*24.096*	*0.838*	*21.431*	*0.173*
2 sources ($K = 2$)					
Gaussian blur, JPEG compression	0.00090	30.793	0.958	6.325	0.047
Gaussian blur, x4 bicubic SR	0.00081	30.962	0.959	6.470	0.046
Gaussian blur, Gaussian noise	0.00081	30.825	0.958	6.413	0.047
JPEG compression, x4 bicubic SR	0.00694	21.115	0.704	20.474	0.217
JPEG compression, Gaussian noise	0.00658	21.077	0.698	17.144	0.273
x4 bicubic SR, Gaussian noise	0.00431	23.073	0.799	13.256	0.167
Fusion	*0.00188*	*26.743*	*0.904*	*15.149*	*0.103*
3 sources ($K = 3$)					
Gaussian blur, JPEG compression, Gaussian noise	0.00072	31.303	0.962	6.002	0.044
Gaussian blur, JPEG compression, x4 bicubic SR	0.00074	31.360	0.962	6.028	0.045
Gaussian blur, x4 bicubic super-resolution, Gaussian noise	0.00066	31.638	0.964	5.761	0.042
JPEG compression, x4 bicubic SR, Gaussian noise	0.00426	23.132	0.790	27.750	0.192
Fusion	*0.00087*	*30.247*	*0.954*	*11.531*	*0.056*
4 sources ($K = 4$)					
Gaussian blur, JPEG compression, x4 bicubic SR, Gaussian noise	0.00070	31.417	0.963	5.794	0.044

Table A1: MSE, PSNR, SSIM, FID, and LPIPS of MB-DDPM on the CIFAR-10 test set (10k samples) using 1–4 sources. Fusion results are obtained by averaging the predicted images across all source combinations.

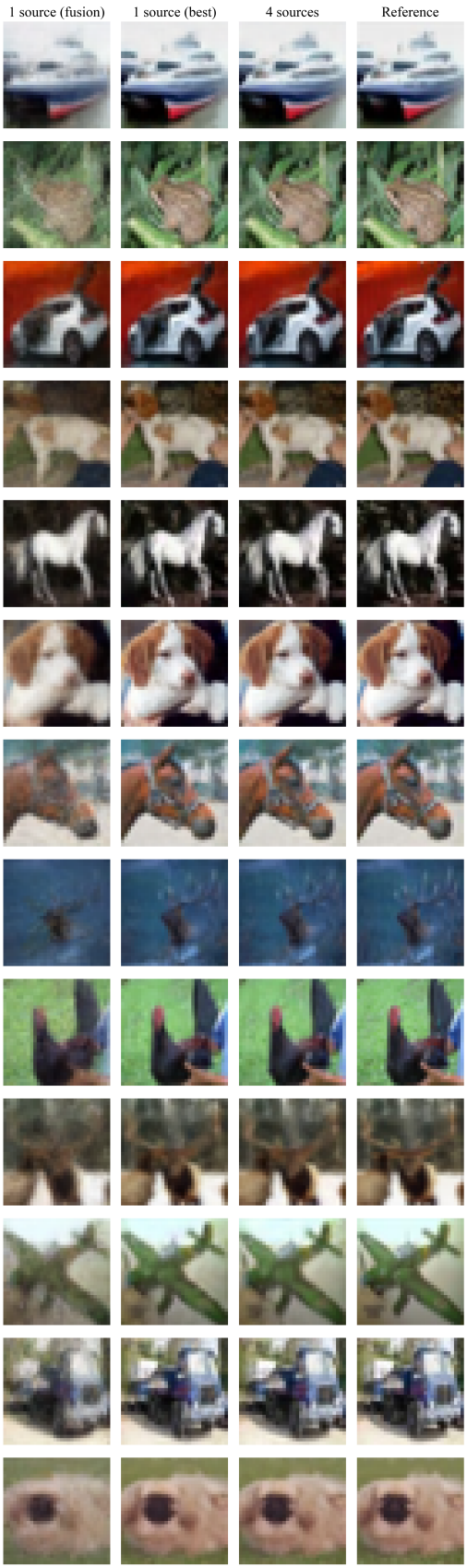


Figure A6: Prediction of MB-DDPM on CIFAR-10 test set (10k samples) using 1 and 4 sources.

E.1.2 Running time

To further analyze the experiments in Section 4.1, we investigate how runtime scales with the number of sources K and observe only a negligible increase in Table A2. This is because MB-DDPM uses a U-Net applied to the concatenated inputs $(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})$ for the conditional estimator (and $\mathbf{x}_t^{1:K}$ for the marginal estimator). While the input channel dimension increases with K, the network architecture and capacity remain fixed.

This is reflected in the number of parameters, which changes only marginally from 988.45M (one source) to 988.51M (four sources), confirming that scaling K does not meaningfully affect model size or complexity.

During sampling, operations across sources are efficiently vectorized, resulting in minimal additional overhead. Moreover, training is parallelizable across conditional and marginal estimators and sampling is parallelizable across samples since trajectories are independent. The reported sampling runtimes are therefore conservative, as they do not fully exploit this parallelism.

Source Images	**Training**		**Sampling**	**Total**
	Conditional	**Marginal**		
1 source ($K = 1$)				
Gaussian blur	7.02	7.07	7.78	21.87
JPEG compression	6.75	7.06	7.68	21.49
x4 bicubic SR	6.98	7.08	7.62	21.68
Gaussian noise	7.03	7.05	7.63	21.71
Average	*6.95*	*7.07*	*7.68*	*21.69*
2 sources ($K = 2$)				
Gaussian blur, JPEG compression	7.03	6.97	7.64	21.64
Gaussian blur, x4 bicubic SR	7.14	6.98	7.63	21.75
Gaussian blur, Gaussian noise	7.13	7.09	7.66	21.88
JPEG compression, x4 bicubic SR	6.99	7.06	7.73	21.78
JPEG compression, Gaussian noise	7.10	6.96	7.64	21.70
x4 bicubic SR, Gaussian noise	7.09	7.09	7.63	21.81
Average	*7.08*	*7.03*	*7.66*	*21.76*
3 sources ($K = 3$)				
Gaussian blur, JPEG compression, Gaussian noise	6.99	6.99	7.63	21.61
Gaussian blur, JPEG compression, x4 bicubic SR	7.05	7.19	7.81	22.05
Gaussian blur, x4 bicubic super-resolution, Gaussian noise	7.06	7.09	7.46	21.61
JPEG compression, x4 bicubic SR, Gaussian noise	7.05	7.18	7.78	22.01
Average	*7.04*	*7.11*	*7.67*	*21.82*
4 sources ($K = 4$)				
Gaussian blur, JPEG compression, x4 bicubic SR, Gaussian noise	7.04	7.23	7.64	21.91
Average	*7.04*	*7.23*	*7.64*	*21.91*

Table A2: Running time (hours) of MB-DDPM for training on the CIFAR-10 train set (50k samples) and sampling on the CIFAR-10 test set (10k samples) using 1–4 sources. *Average* results are obtained by averaging the running time across all source combinations.

E.1.3 Shapley-style values

To further analyze the experiments in Section 4.1, Figure A7 illustrates the change in evaluation metrics when removing a single source from the full set of four degradations. For each source, we approximate its contribution using a Shapley-style marginal effect: we compare the performance of MB-DDPM conditioned on all four sources with the performance obtained when that source is excluded.

Formally, let $\hat{\mathbf{x}}_0(\mathbf{x}_T^{1:K})$ denote the predicted target given sources $\mathbf{x}_T^{1:K}$. For source k, we define its contribution as:

$$C_k^{\mathcal{E}} = \mathbb{E}\big[\mathcal{E}(\mathbf{x}_0, \hat{\mathbf{x}}_0(\mathbf{x}_T^{1:K\setminus k}) - \mathcal{E}(\mathbf{x}_0, \hat{\mathbf{x}}_0(\mathbf{x}_T^{1:K}))\big], \tag{93}$$

where $\mathcal{E}$ is an evaluation metric such as MSE, PSNR, SSIM, or LPIPS. Note that FID is not included since it cannot be computed at the per-sample level. The expectation is approximated across the test samples.

Postitive (resp. negative) changes in MSE and LPIPS (resp. PSNR and SSIM) indicate that removing the source degrades performance, reflecting its relative importance in the multi-source setting.

We observe that milder degradations, such as Gaussian blur, provide the largest performance gains, whereas more severe degradations, such as JPEG compression, can negatively impact performance. This trend is consistent with the observations reported in Table A1.

Like before, this result highlights that the choice of source combination can influence performance and motivates evaluating multiple combinations rather than relying solely on the maximum number of available sources. Importantly, the observed difference remains marginal, suggesting that performance is generally robust to the inclusion of additional sources. We further investigate this in Appendix E.6, where we confirm this observation. Moreover, exploring such multi-source subsets is only possible within a framework such as MB-DDPM.

Overall, evaluating several source combinations can be beneficial when computationally feasible. Performance improves on average with the number of sources, and the four-source setting achieves the best aggregate performance across combinations.

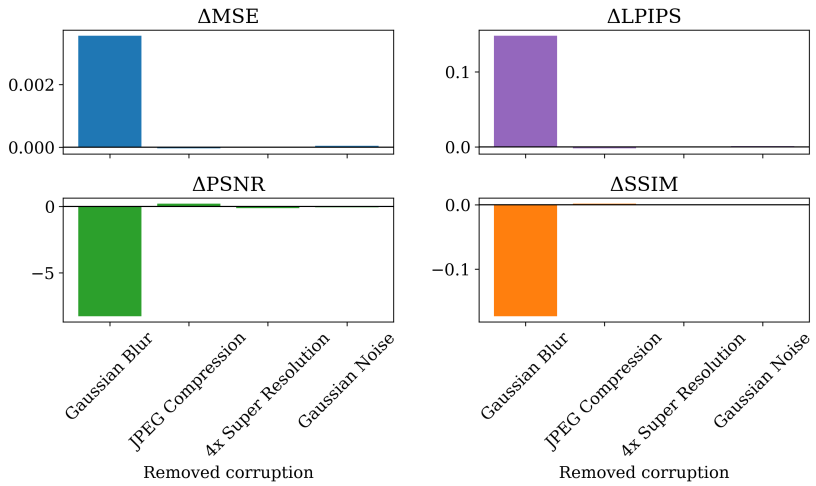


Figure A7: Shapley-style contribution (93) of each source evaluated with MB-DDPM on the CIFAR-10 test set (10k samples). Each bar represents the performance difference between MB-DDPM conditioned on all four sources and MB-DDPM with one source excluded.

E.2 Effect of Source Disagreement

We define the source disagreement for one set of sources $\mathbf{x}_T^{1:K}$ as:

$$D(\mathbf{x}_T^{1:K}) = \frac{2}{K(K-1)} \sum_{k<k'} \mathcal{E}\big(\mathbf{x}_T^k, \mathbf{x}_T^{k'}\big),$$

where $\mathcal{E}$ denotes a pairwise distance metric, such as MSE or LPIPS. The normalization factor $\frac{2}{K(K-1)}$ corresponds to the inverse of the number of unique source pairs, $\binom{K}{2}$, and therefore makes D the mean pairwise disagreement among the sources. Since the scale of $\mathcal{E}$ depends on the dataset and metric, disagreement values are interpreted within each dataset and metric and are not intended to be directly compared across datasets.

We then compute standard observation-level evaluation metrics. Formally, let $\mathbf{x}_0$ be one target observation and let $\hat{\mathbf{x}}_0(\mathbf{x}_T^{1:K})$ denote the predicted target using MB-DDPM given sources $\mathbf{x}_T^{1:K}$. We define the observation-level evaluation metrics as:

$$E(\mathbf{x}_0, \mathbf{x}_T^{1:K}) = \mathcal{E}(\mathbf{x}_0, \hat{\mathbf{x}}_0(\mathbf{x}_T^{1:K})),$$

where $\mathcal{E}$ is an evaluation metric such as MSE or LPIPS.

Figure A8 shows scatter plots of source disagreement ($D(\mathbf{x}_T^{1:K})$) versus evaluation metrics ($E(\mathbf{x}_0, \mathbf{x}_T^{1:K})$) for each sources-target pairs across the three datasets (CIFAR-10 ($K = 4$), ImageNet ($K = 4$), and MRI ($K = 3$)). We observe consistent trends across datasets: as source disagreement increases, reconstruction quality degrades. On average, MSE increases by 0.0008 for a 0.05 increase in disagreement while LPIPS increases by 0.015 for a 0.05 increase in disagreement, indicating a clear performance drop when sources provide conflicting or heterogeneous information.

Finally, Figure A9-A11 visualizes representative samples, including the four most similar sources, the four most dissimilar sources, and intermediate cases across all datasets. These examples further illustrate that reconstruction quality deteriorates as source disagreement increases.

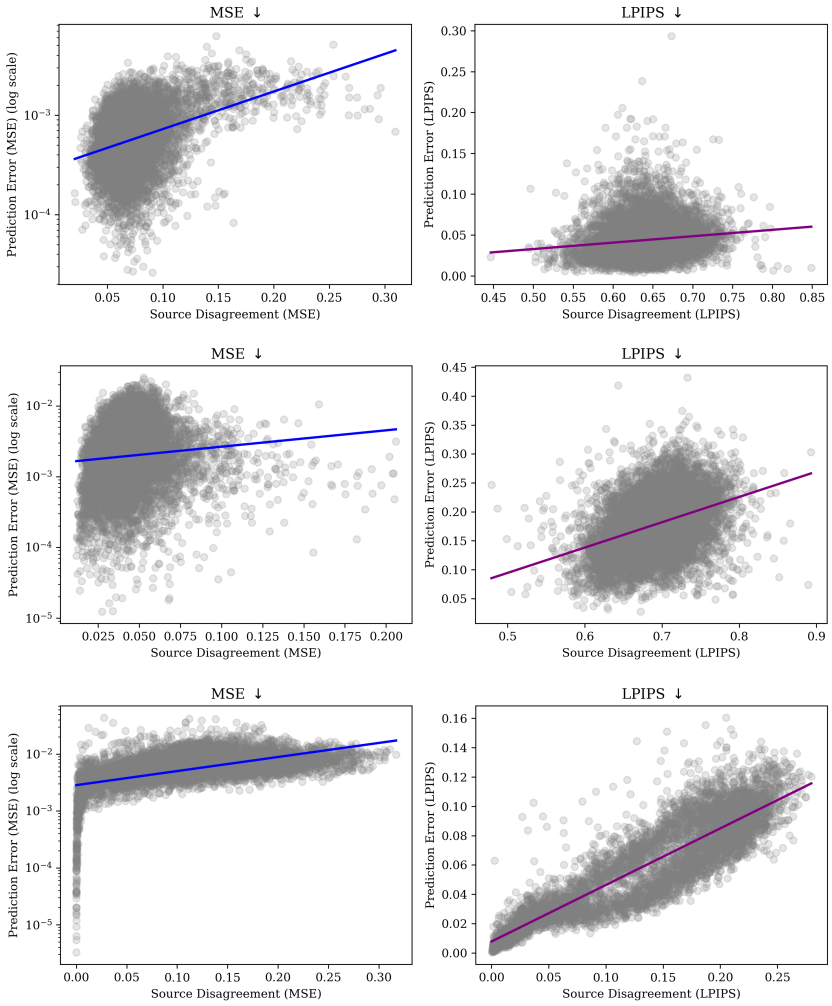


Figure A8: Scatter plots of observation-level source disagreement versus evaluation metrics across CIFAR-10 (top), ImageNet (middle), and MRI (bottom) datasets. Source disagreement is computed using MSE- and LPIPS-based pairwise distances between sources. Each point represent a single observations in the test set ($n_{\text{test}} = 10,000$ for CIFAR-10 and ImageNet, $n_{\text{test}} = 5,492$ for MRI). The line shows a linear regression fit.

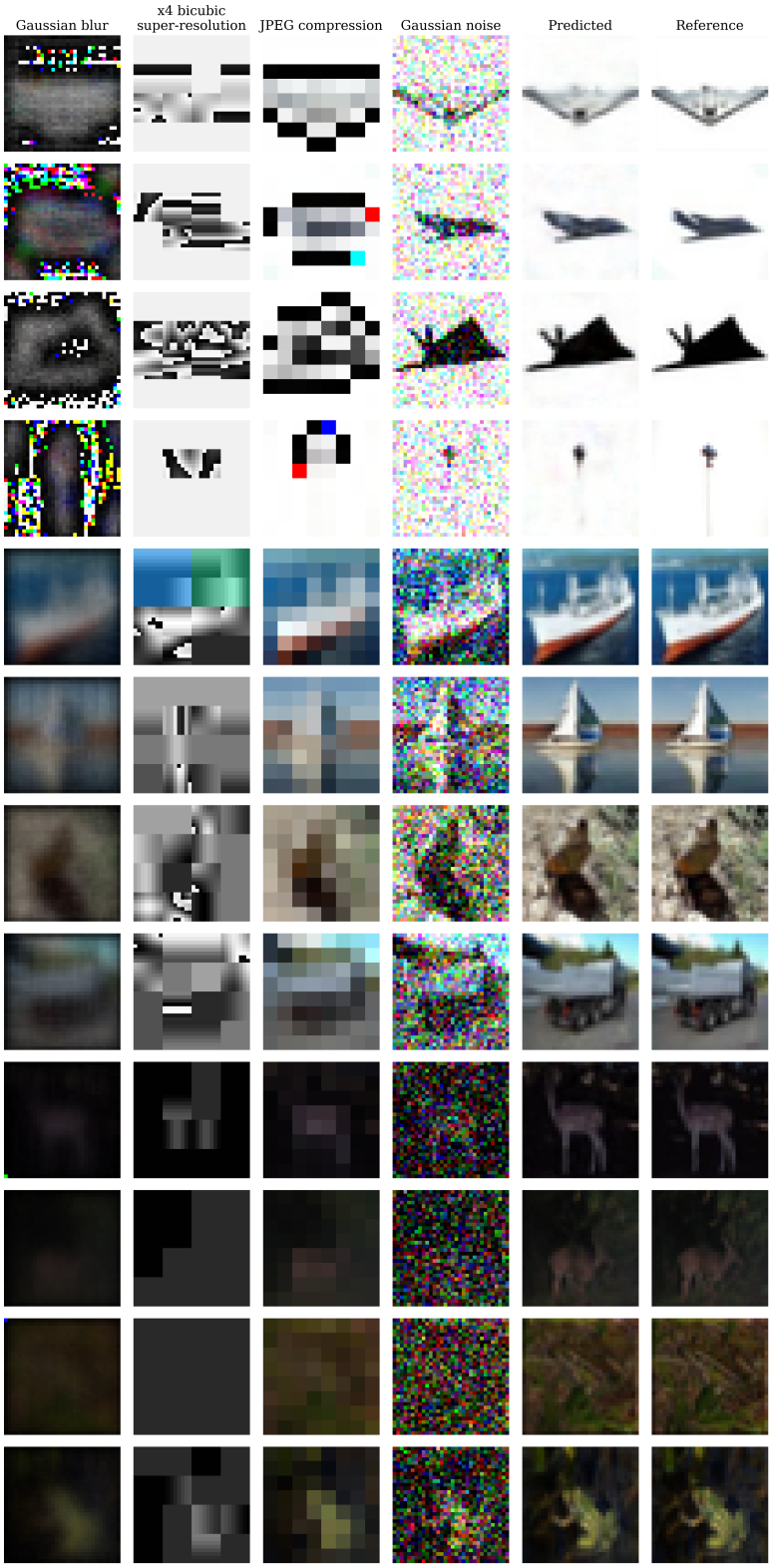


Figure A9: MB-DDPM predictions using four sources (Gaussian noise, Gaussian blur, $4\times$ bicubic super-resolution, and JPEG compression with quality factor 5) on the CIFAR-10 test set. The figure shows the most dissimilar cases (top four rows), intermediate cases (middle four rows), and the most similar cases (bottom four rows).

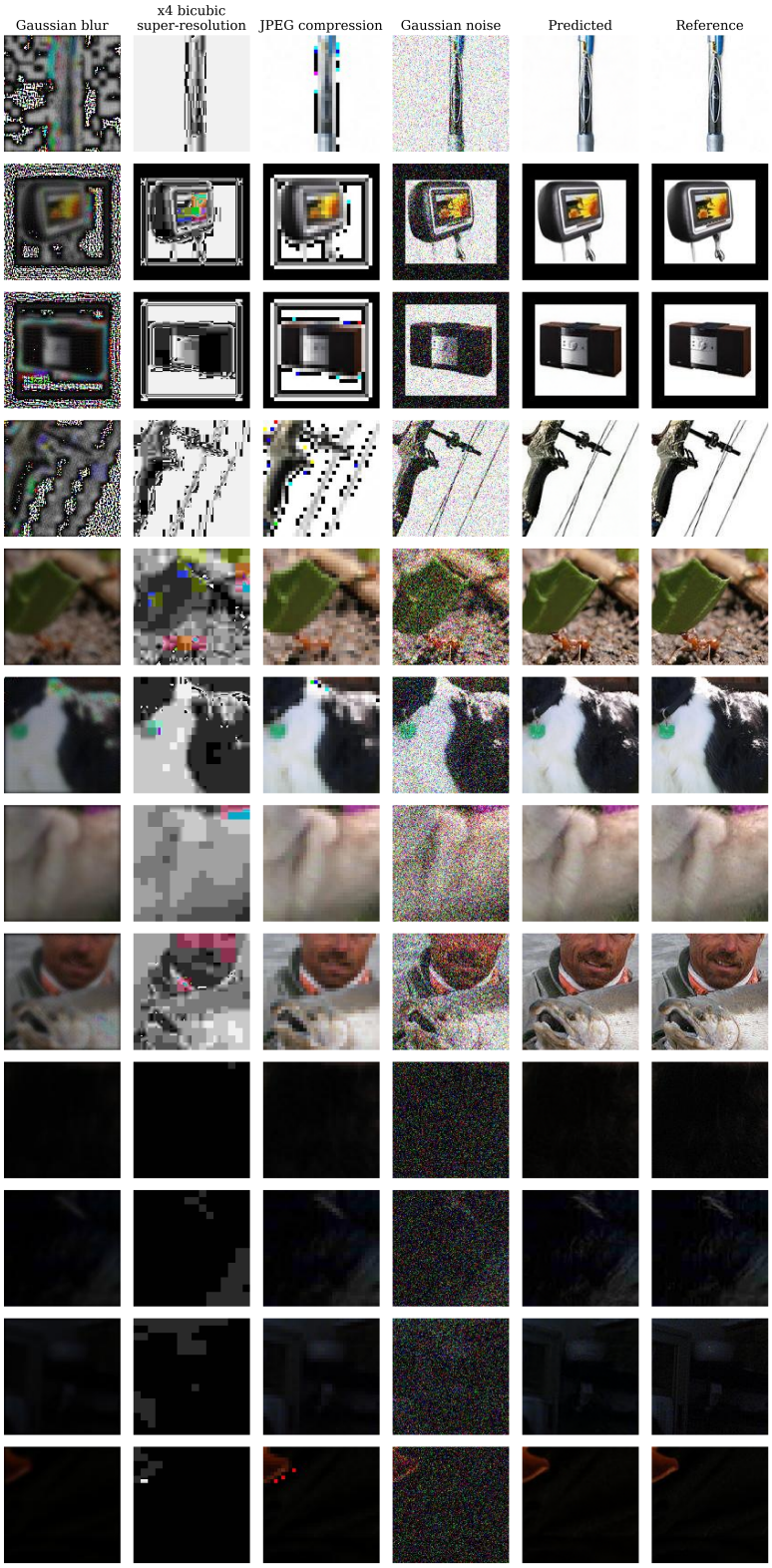


Figure A10: MB-DDPM predictions using four sources (Gaussian noise, Gaussian blur, $4\times$ bicubic super-resolution, and JPEG compression with quality factor 5) on the ImageNet test set. The figure shows the most dissimilar cases (top four rows), intermediate cases (middle four rows), and the most similar cases (bottom four rows).

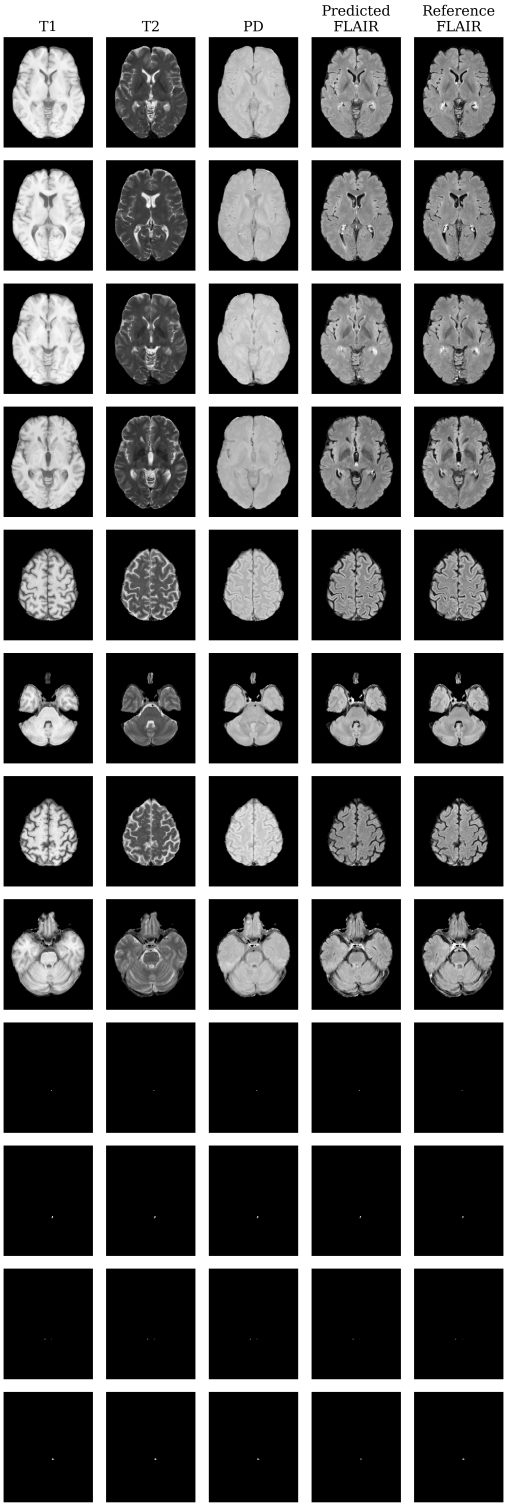


Figure A11: MB-DDPM FLAIR predictions using three sources (T1, T2 and PD contrasts) on the MRI test set. The figure shows the most dissimilar cases (top four rows), intermediate cases (middle four rows), and the most similar cases (bottom four rows).

E.3 T2 Lesion Volume Prediction

To further assess the clinical fidelity of the predicted FLAIR images, we evaluate their utility in a downstream task: T2 lesion volume estimation. Rather than evaluating image quality alone, this experiment measures whether the predicted FLAIR images preserve diagnostically relevant information.

T2 lesion volume is a clinically relevant biomarker in multiple sclerosis (MS), reflecting the overall lesion burden in the brain. These lesions appear on MRI, particularly on FLAIR sequences, as hyperintense (bright) spots due to increased water content associated with demyelination and inflammation (Figure A12).

T2 lesion volume is a non-negative scalar that quantifies the total lesion burden and must be computed at the level of the full brain MRI scan, i.e., across all slices.

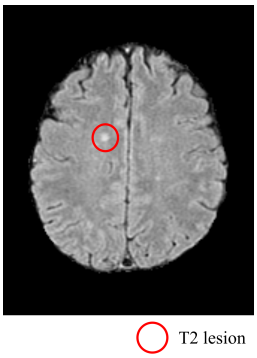


Figure A12: Illustration of T2 lesion visualization on a FLAIR contrast image.

Lesion segmentation model: T1 + FLAIR → T2 lesion mask → T2 lesion volume. The lesion segmentation model was previously developed and trained on the NO.MS dataset in prior work [30, 27]. In this study, we use the pretrained model exclusively for inference, providing it with the input images to obtain lesion segmentations, without any additional training or fine-tuning. The model takes as input axial slices of co-registered T1-weighted and FLAIR images and produces a binary segmentation mask corresponding to T2 hyperintense lesions. Ground-truth lesion annotations are generated using an automated segmentation pipeline and subsequently refined by expert radiologists to ensure high-quality supervision.

The network follows a standard encoder–decoder U-Net architecture with three downsampling and three upsampling stages connected via skip connections. Each encoder block consists of convolutional layers with ReLU activations followed by max-pooling, progressively increasing the number of feature channels ($64 \rightarrow 128 \rightarrow 256$). The decoder mirrors this structure, using upsampling and convolutional layers to recover spatial resolution while integrating high-resolution features from the encoder. The model is trained slice-wise using a Dice loss function on approximately 2,000 MRI scans (corresponding to $\sim$90,000 slices) from the NO.MS dataset.

For each scan, the predicted T2 lesion masks across all slices are aggregated to compute the total T2 lesion volume. Specifically, lesion area is computed per slice and summed across slices, accounting for voxel spacing to obtain a volumetric estimate in mL ($mm^3/1000$). This results in a non-negative scalar representing total lesion burden per scan.

To validate the model, we compared lesion volumes derived from predicted masks against those obtained from ground-truth masks. We observe strong agreement between predicted and reference volumes (Pearson r: 0.976, ICC: 0.945), indicating that the segmentation model provides reliable volumetric estimates.

Evaluation of predicted FLAIR. To assess the fidelity of predicted FLAIR images from MB-DDPM and other baseline methods, we evaluate their performance on downstream lesion volume quantification using the same segmentation pipeline. To isolate the effect of FLAIR synthesis, we compare two input configurations: (i) real T1 with real FLAIR, and (ii) real T1 with predicted FLAIR. In both cases, the same segmentation model is applied without modification. Lesion volumes are computed from the resulting T2 lesion masks. This evaluation measures whether replacing real FLAIR with predicted FLAIR preserves clinically relevant information for lesion detection and quantification.

Importantly, the segmentation model was trained independently of MB-DDPM and the baseline methods, and its outputs were not used for their training, model selection, or hyperparameter tuning. Furthermore, MB-DDPM and the baseline methods were trained exclusively on the available MRI contrasts (T1, T2, and PD) and the target FLAIR images, without access to lesion masks, lesion volumes, or segmentation outputs. The purpose of this analysis is therefore not to assess the absolute

segmentation accuracy or generalization of the segmentation model, but rather to evaluate the extent to which lesion-related information is preserved by FLAIR synthesis. We acknowledge that the exact training cohort of the external segmentation model is unavailable, and thus possible subject overlap cannot be entirely ruled out. Nevertheless, applying the same segmentation model to both real and predicted FLAIR images provides a consistent basis for comparing their impact on downstream lesion volume quantification.

Results. Table A3 reports the Pearson correlation coefficient and the intraclass correlation coefficient (ICC) for each method. Additionally, Figure A13 presents a scatter plot comparing T2 lesion volumes extracted from segmentation outputs for MB-DDPM and the three best performing baseline (CDM, CDE and I^2SB (T2 source)). The x-axis corresponds to volumes obtained from real FLAIR images, while the y-axis shows volumes derived from predicted FLAIR images generated by each method. Each point represents the lesion volume for a single scan (118 scans in the test set). Points below the diagonal indicate underestimation, reflecting methods that fail to fully recover all T2 lesions in the predicted FLAIR images.

Overall, the Pearson coefficient, the ICC and the scatter plot demonstrate excellent agreement between lesion volumes derived from real and predicted FLAIR produced by MB-DDPM. Most data points lie close to the identity line, indicating minimal systematic bias between the two measurements. These results support the reliability of MB-DDPM and suggest that it can serve as a practical surrogate for real FLAIR images in downstream clinical analyses.

In contrast, baseline methods other than CDM generally tend to underestimate lesion volume, as they fail to consistently recover all T2 lesions. This underestimation stems from incomplete reconstruction of hyperintense lesion regions (i.e., the characteristic bright spots on FLAIR), visible in Figure A17.

T2 Lesion Volume		
Method	Pearson Coefficient ↑	ICC ↑
CDE [4]	0.9913	0.9822
CDM [37]	<u>0.9930</u>	<u>0.9927</u>
I^2SB [16] (T1)	0.5325	0.2507
I^2SB [16] (T2)	0.9075	0.6445
I^2SB [16] (PD)	0.8978	0.6935
I^2SB [16] (Fusion)	0.8871	0.5791
MB-DDPM (ours)	**0.9941**	**0.9939**

Table A3: T2 lesion volume concordance evaluated with the Pearson correlation coefficient and ICC is shown. For each metric, the best-performing method is highlighted in bold, and the second-best is underlined. For DPS, CDE, CDM, and MB-DDPM, results are obtained using three source contrasts: T1, T2 and PD. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. Details on baseline methods are provided in Appendix C.

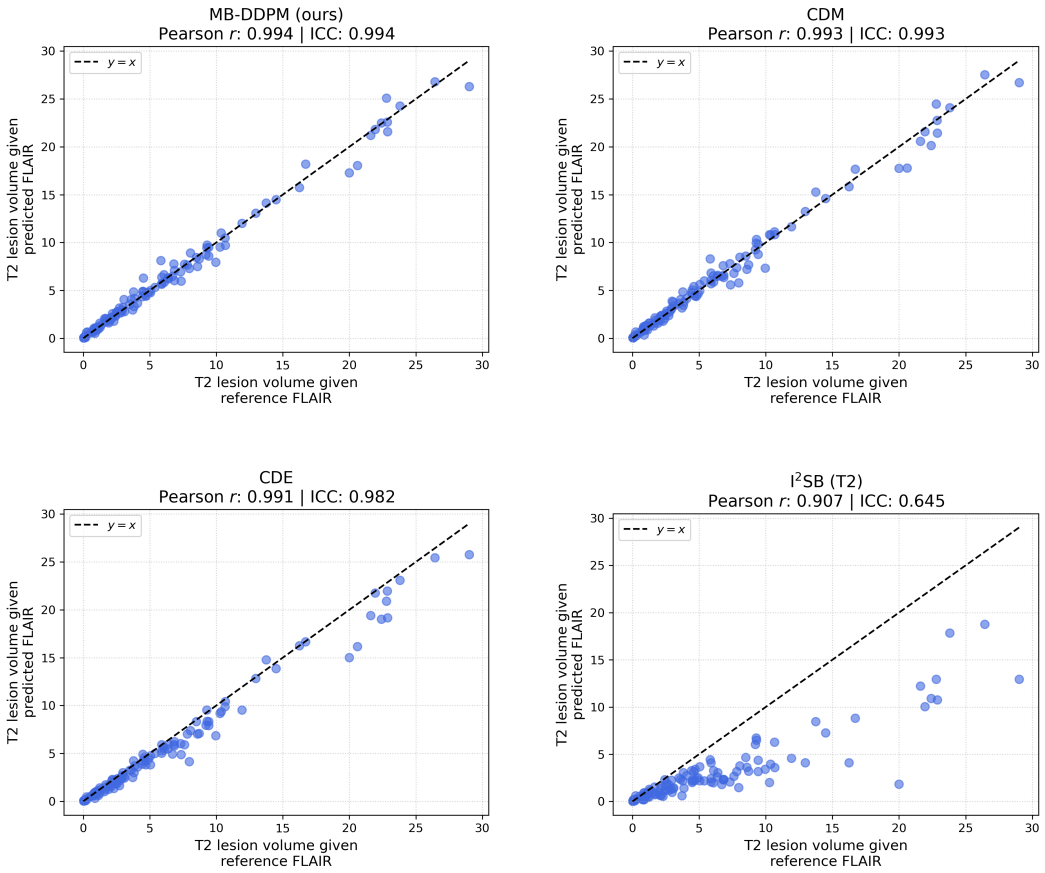


Figure A13: T2 lesion volume prediction by MB-DDPM and the three best-performing baselines (CDM [37], CDE [4] and I^2SB [16] (T2 source)). Each point represents the predicted T2 lesion volume for a MRI scan (aggregated across slices) in the test set. The diagonal line indicates perfect agreement between predicted and reference T2 lesion volumes, corresponding to the ideal case of using true FLAIR. Points below the diagonal correspond to underestimation of T2 lesion volume, indicating methods that fail to fully recover all T2 lesions in the predicted FLAIR contrasts. Details on baseline methods are provided in Appendix C.

E.4 Comparison with Baselines

E.4.1 Quantitative Comparison

The inputs and outputs of MB-DDPM are illustrated in Figure A14 for image restoration (CIFAR-10 and ImageNet, $K = 4$ sources) and MRI ($K = 3$ sources). For CIFAR-10 and ImageNet, each source corresponds to a different degraded observation of the same image, while for MRI each source corresponds to a different imaging contrast.

We report quantitative comparisons in terms of MSE, SSIM, PSNR, FID, and LPIPS across all baselines. Results for CIFAR-10 ($K = 4$), ImageNet ($K = 4$), and MRI data ($K = 3$) are presented in Tables A4, A5, and A6, respectively.

In addition, qualitative comparisons of reconstructed images for each baseline are shown in Figure A15 for CIFAR-10, Figure A16 for ImageNet, and Figure A17 for MRI.

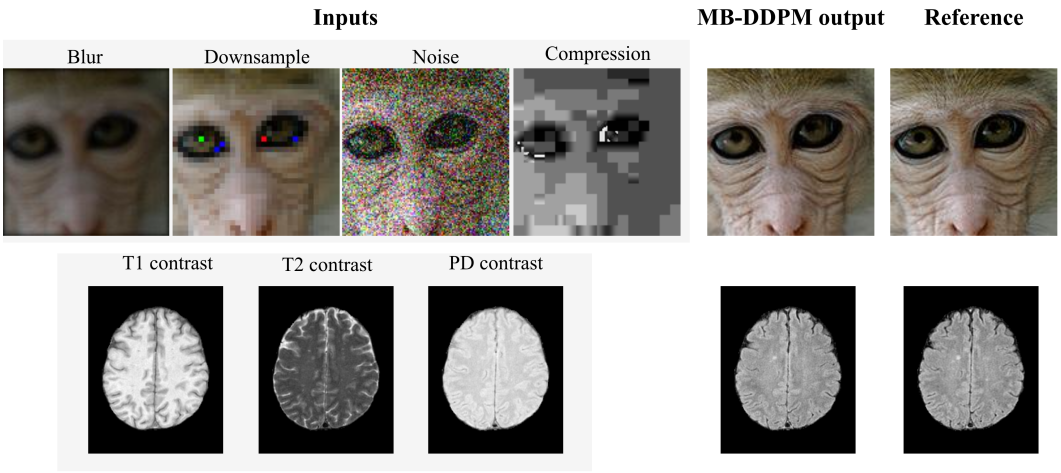


Figure A14: Outputs of MB-DDPM on ImageNet (Top) MRI (Bottom).

CIFAR-10					
Method	MSE ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
DPS [6]	0.00690	20.758	0.721	21.789	0.276
CDE [4]	0.00158	27.330	0.901	45.062	0.116
CDM [37]	<u>0.00073</u>	<u>31.290</u>	<u>0.962</u>	<u>6.013</u>	**0.042**
I^2SB [16] (Gaussian blur)	0.01774	16.898	0.467	19.183	0.383
I^2SB [16] (JPEG compression)	0.01833	16.654	0.448	27.722	0.447
I^2SB [16] (Gaussian noise)	0.01172	18.646	0.524	32.645	0.378
I^2SB [16] (Super-resolution)	0.01488	17.667	0.507	19.815	0.348
I^2SB [16] (Fusion)	0.01105	18.900	0.550	32.685	0.360
MB-DDPM (ours)	**0.00070**	**31.417**	**0.963**	**5.794**	<u>0.044</u>

Table A4: MSE, PSNR, SSIM, FID, and LPIPS evaluated on the CIFAR-10 test set (10k samples). For each metric, the best-performing method is highlighted in bold, and the second-best is underlined. For DPS, CDE, CDM, and MB-DDPM, results are obtained using four source degradations: Gaussian blur, JPEG compression, Gaussian noise, and super-resolution. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. Details on baseline methods are provided in Appendix C.

ImageNet					
Method	MSE ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
DPS [6]	0.00872	21.109	0.504	48.752	0.469
CDE [4]	0.00914	20.191	0.420	41.568	0.448
CDM [37]	<u>0.00308</u>	<u>26.615</u>	**0.775**	<u>7.027</u>	<u>0.187</u>
I^2SB [16] (Gaussian blur)	0.01062	20.621	0.460	32.866	0.419
I^2SB [16] (JPEG compression)	0.01299	19.236	0.419	56.759	0.497
I^2SB [16] (Gaussian noise)	0.00692	22.257	0.541	29.881	0.386
I^2SB [16] (Super-resolution)	0.00972	21.151	0.488	42.936	0.409
I^2SB [16] (Fusion)	0.00804	21.669	0.503	37.464	0.402
MB-DDPM (ours)	**0.00303**	**26.671**	<u>0.773</u>	**5.250**	**0.178**

Table A5: MSE, PSNR, SSIM, FID, and LPIPS evaluated on the ImageNet test set (10k samples). For each metric, the best-performing method is highlighted in bold, and the second-best is underlined. For DPS, CDE, CDM, and MB-DDPM, results are obtained using four source degradations: Gaussian blur, JPEG compression, Gaussian noise, and super-resolution. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. Details on baseline methods are provided in Appendix C.

MRI					
Method	MSE ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
CDE [4]	0.00812	26.840	0.770	28.262	0.247
CDM [37]	<u>0.00751</u>	<u>27.478</u>	<u>0.868</u>	<u>5.026</u>	<u>0.066</u>
I^2SB [16] (T1)	0.01396	24.801	0.791	8.071	0.107
I^2SB [16] (T2)	0.03167	21.425	0.739	7.889	0.133
I^2SB [16] (PD)	0.01462	24.570	0.790	8.501	0.111
I^2SB [16] (Fusion)	0.01402	24.752	0.790	14.744	0.111
MB-DDPM (ours)	**0.00679**	**27.929**	**0.874**	**3.472**	**0.061**

Table A6: MSE, PSNR, SSIM, FID, and LPIPS evaluated on the MRI test set (5,492 samples). For each metric, the best-performing method is highlighted in bold, and the second-best is underlined. For DPS, CDE, CDM, and MB-DDPM, results are obtained using three source contrasts: T1, T2 and PD. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. Details on baseline methods are provided in Appendix C.

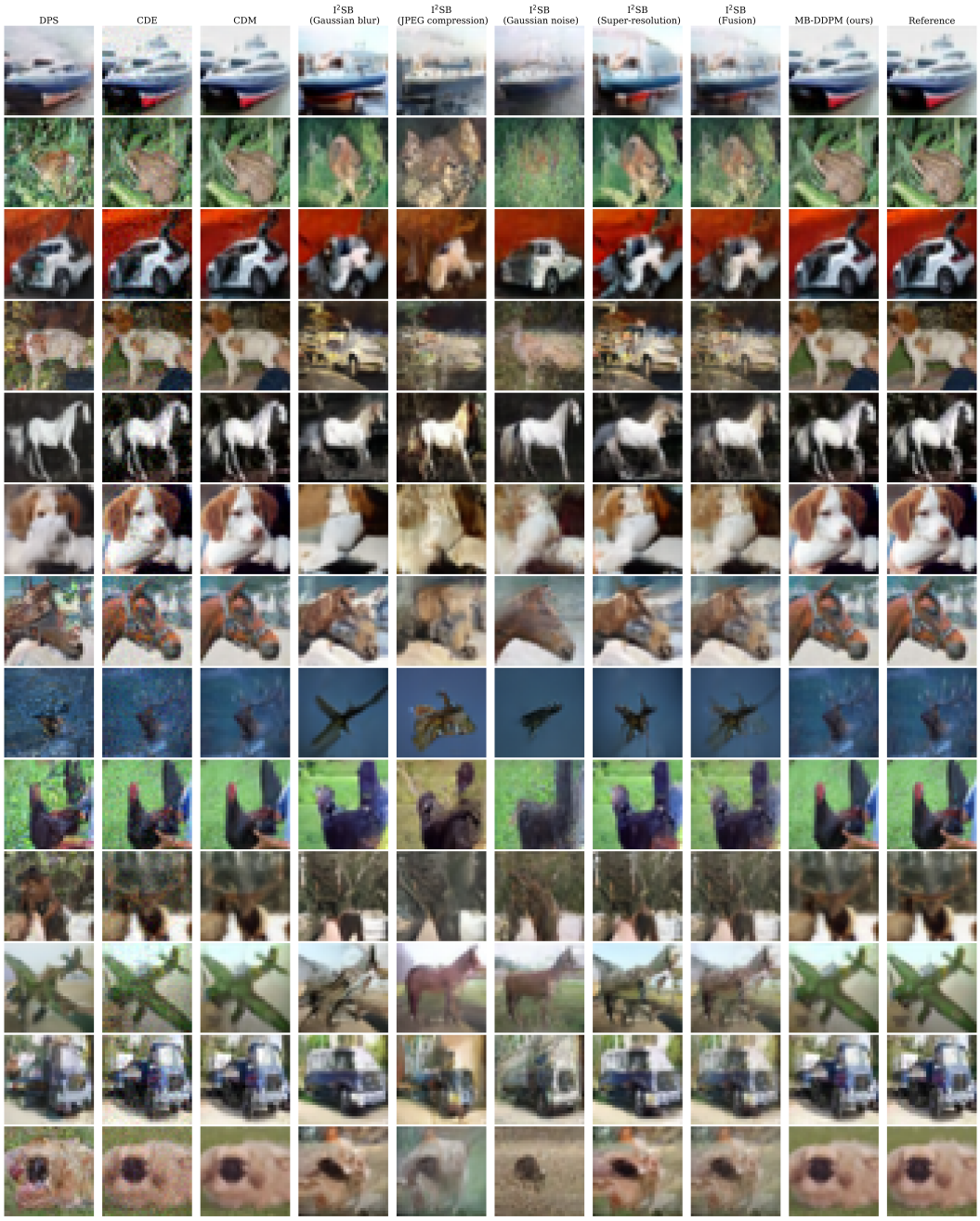


Figure A15: Prediction of DPS [6], CDE [4], CDM [37], I^2SB [16] and MB-DDPM (ours) on CIFAR-10 test set (10k samples). For DPS, CDE, CDM, and MB-DDPM, results are obtained using four source degradations: Gaussian blur, JPEG compression, Gaussian noise, and super-resolution. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. Details on baseline methods are provided in Appendix C.

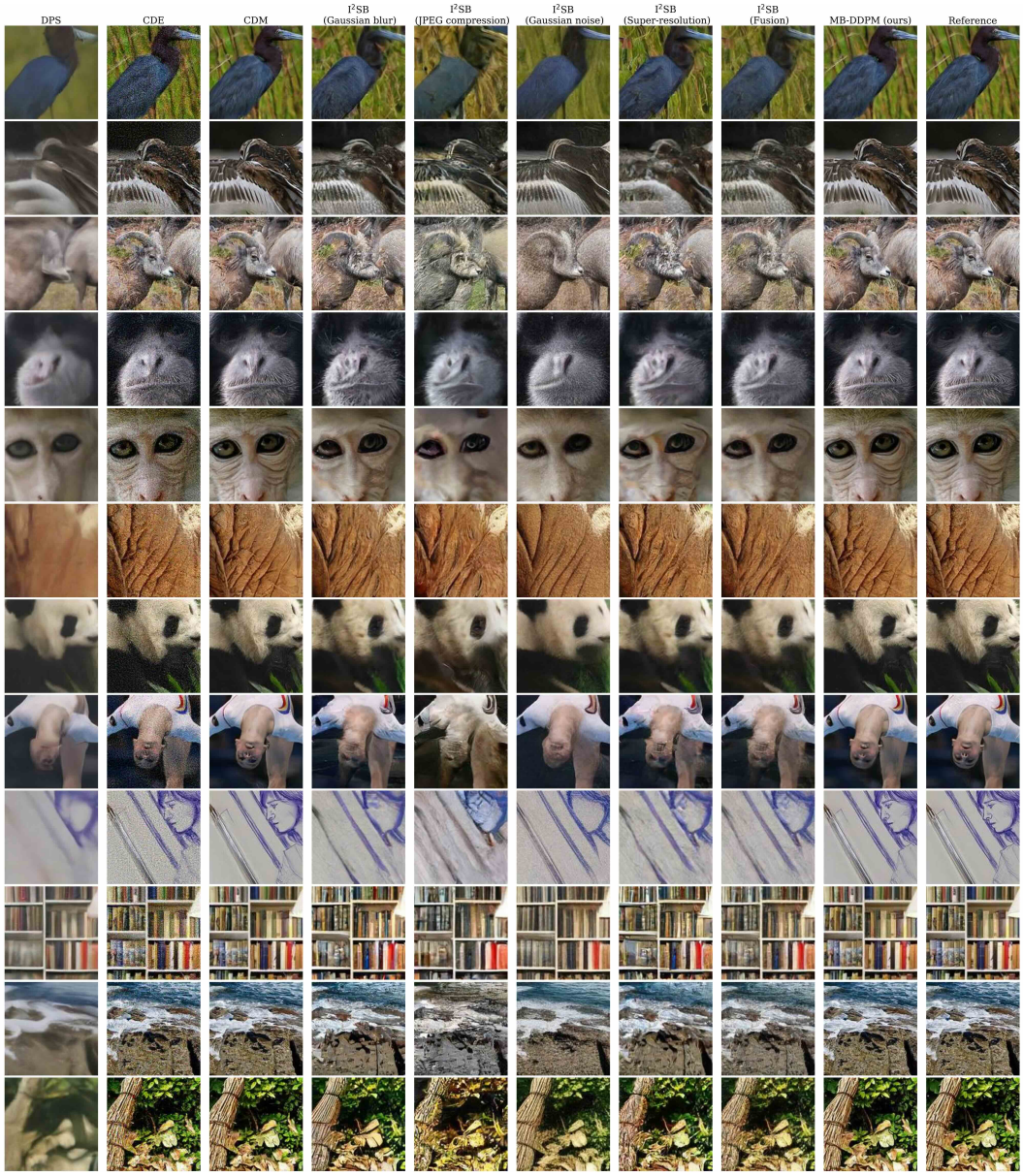


Figure A16: Prediction of DPS [6], CDE [4], CDM [37], I^2SB [16] and MB-DDPM (ours) on ImageNet test set (10k samples). For DPS, CDE, CDM, and MB-DDPM, results are obtained using four source degradations: Gaussian blur, JPEG compression, Gaussian noise, and super-resolution. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. Details on baseline methods are provided in Appendix C.

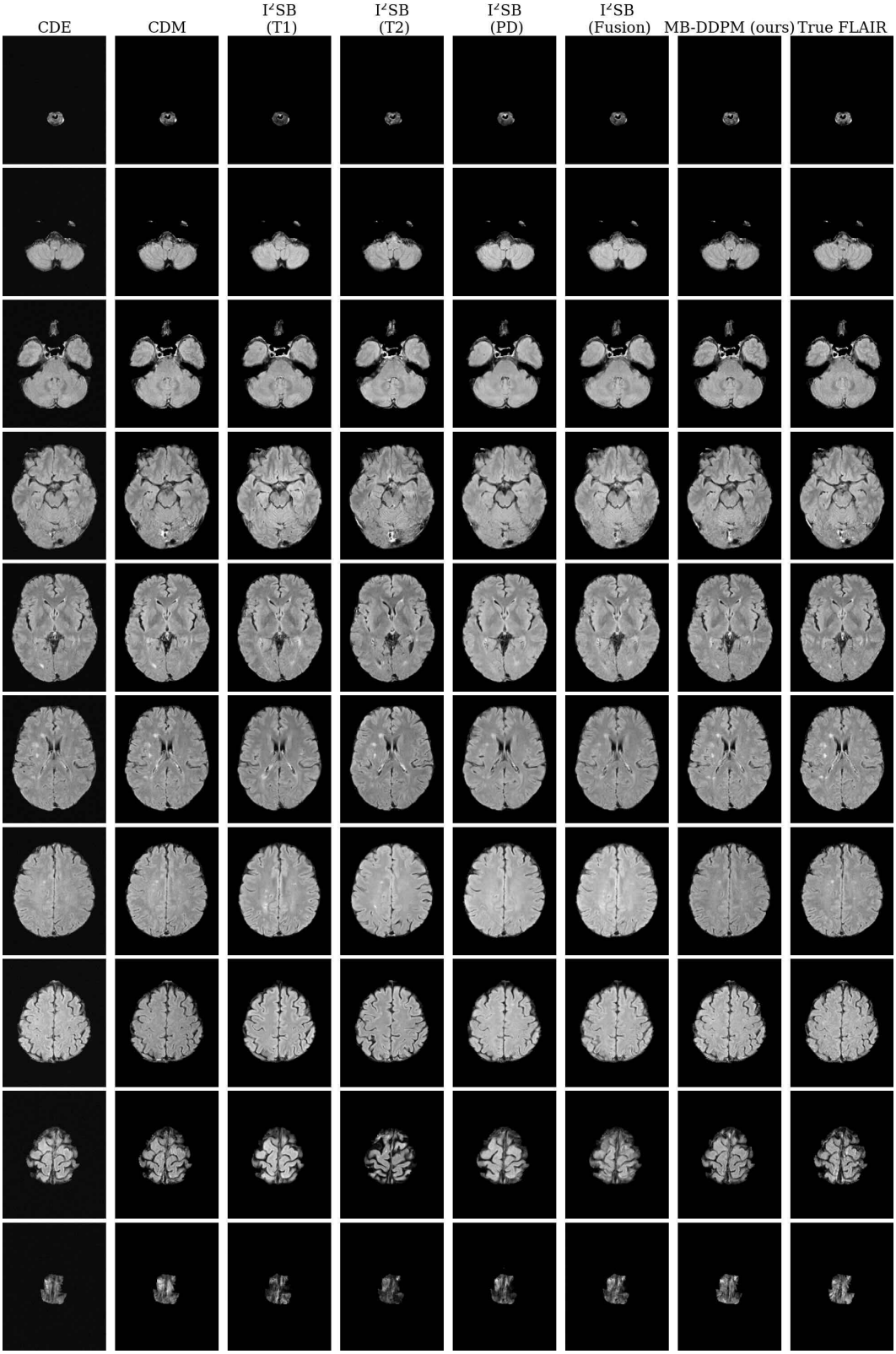


Figure A17: Prediction of DPS [6], CDE [4], CDM [37], I^2SB [16] and MB-DDPM (ours) on MRI test set (5,492 samples). For DPS, CDE, CDM, and MB-DDPM, results are obtained using three source contrasts: T1, T2 and PD. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. Details on baseline methods are provided in Appendix C.

E.4.2 Running time

We report running times across all baselines for CIFAR-10, ImageNet, and MRI in Tables A7, A8, and A9, respectively.

DPS [6] consistently exhibits high sampling costs, which dominate the overall runtime across CIFAR-10 and Imagenet datasets. Indeed, the method requires evaluating `torch.autograd.grad` which introduces a backward pass through the network during sampling. This operation is significantly more costly than a standard forward pass and increases both computation time and memory usage. As a result, DPS is impractical for applications requiring repeated or efficient inference.

In contrast, I^2SB [16] suffers from a different limitation: it requires training and evaluating multiple source–target models, and its reported performance depends on aggregating predictions from all sources. While each individual model is relatively efficient, the total cost increases substantially when all degradations are considered jointly. This leads to significantly higher overall training and sampling times, making I^2SB expensive in scenarios where all sources must be evaluated.

CDE [4] requires longer sampling times than CDM [37] because its Predictor-Corrector sampler involves both an Euler–Maruyama predictor step and an annealed Langevin dynamics corrector step. The additional score network evaluations in the corrector increase the computational cost of each sampling iteration.

CDM [37] provide more balanced trade-offs, reducing sampling cost compared to DPS and CDE and avoiding the multi-model overhead of I^2SB. The sampling is also faster than that of MB-DDPM because the latter requires loading the results on both the conditional and the marginal estimator networks.

MB-DDPM is comparable to CDM, CDE, I^2SB and DPS in term of training time. At inference time, its sampling is faster than DPS and I^2SB, although slower than CDM and CDE. Importantly, sampling trajectories are independent across samples and can therefore be executed in parallel, making the additional sampling cost more manageable in practice.

CIFAR-10			
Method	Training	Sampling	Total
DPS [6]	7.16	14.03	21.19
CDE [4]	7.19	7.63	14.82
CDM [37]	7.19	3.91	11.10
I^2SB [16] (Gaussian blur)	7.07	3.79	10.86
I^2SB [16] (JPEG compression)	7.06	3.81	10.87
I^2SB [16] (Gaussian noise)	7.05	3.86	10.91
I^2SB [16] (Super-resolution)	7.08	3.74	10.82
I^2SB [16] (Total)	28.26	15.20	43.46
MB-DDPM (ours)	7.04, 7.23	7.64	14.87

Table A7: Running time (hours) for training on the CIFAR-10 train set (50k samples) and sampling on the CIFAR-10 test set (10k samples). For DPS, CDE, CDM, and MB-DDPM, results are obtained using four source degradations: Gaussian blur, JPEG compression, Gaussian noise, and super-resolution. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. For MB-DDPM, the reported training times correspond to the conditional and marginal estimators, which can be trained in parallel.

ImageNet			
Method	Training	Sampling	Total
DPS [6]	62:32	162.93	225.25
CDE [4]	62.15	92.54	154.69
CDM [37]	62.15	46.05	108.2
I^2SB [16] (Gaussian blur)	58.75	47.33	106.08
I^2SB [16] (JPEG compression)	58.57	46.06	104.63
I^2SB [16] (Gaussian noise)	58.89	46.35	105.24
I^2SB [16] (Super-resolution)	58.93	46.79	105.72
I^2SB [16] (Total)	235.14	186.53	421.67
MB-DDPM (ours)	59.54, 60.21	93.90	154.11

Table A8: Running time (hours) for training on the ImageNet train set (50k samples) and sampling on the ImageNet test set (10k samples). For DPS, CDE, CDM, and MB-DDPM, results are obtained using four source degradations: Gaussian blur, JPEG compression, Gaussian noise, and super-resolution. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. For MB-DDPM, the reported training times correspond to the conditional and marginal estimators, which can be trained in parallel.

MRI			
Method	Training	Sampling	Total
CDE [4]	41.00	135.78	176.78
CDM [37]	41.00	70.52	111.52
I^2SB [16] (T1)	41.49	68.04	109.53
I^2SB [16] (T2)	40.74	68.99	109.73
I^2SB [16] (PD)	41.05	69.76	110.81
I^2SB [16] (Total)	123.28	206.79	330.07
MB-DDPM (ours)	40.99, 41.28	180.03	221.31

Table A9: Running time (hours) for training on the MRI train set (10,984 samples) and sampling on the MRI test set (5,492 samples). For DPS, CDE, CDM, and MB-DDPM, results are obtained using three sources: T1, T2 and PD contrasts. For I^2SB, results are reported for models trained on each source–target pair, as well as for the fusion of predicted targets. For MB-DDPM, the reported training times correspond to the conditional and marginal estimators, which can be trained in parallel.

E.5 Visualizing the MB-DDPM Generative Process

We illustrates the generative process of MB-DDPM by visualizing the predicted states $\mathbf{x}_t^k$ across all sources $k = 1, \ldots, K$ at timesteps $t \in \{1, 0.8, 0.6, 0.4, 0.2\} \times T$. Results for ImageNet, and MRI are presented in Figures A18, and A19, respectively.

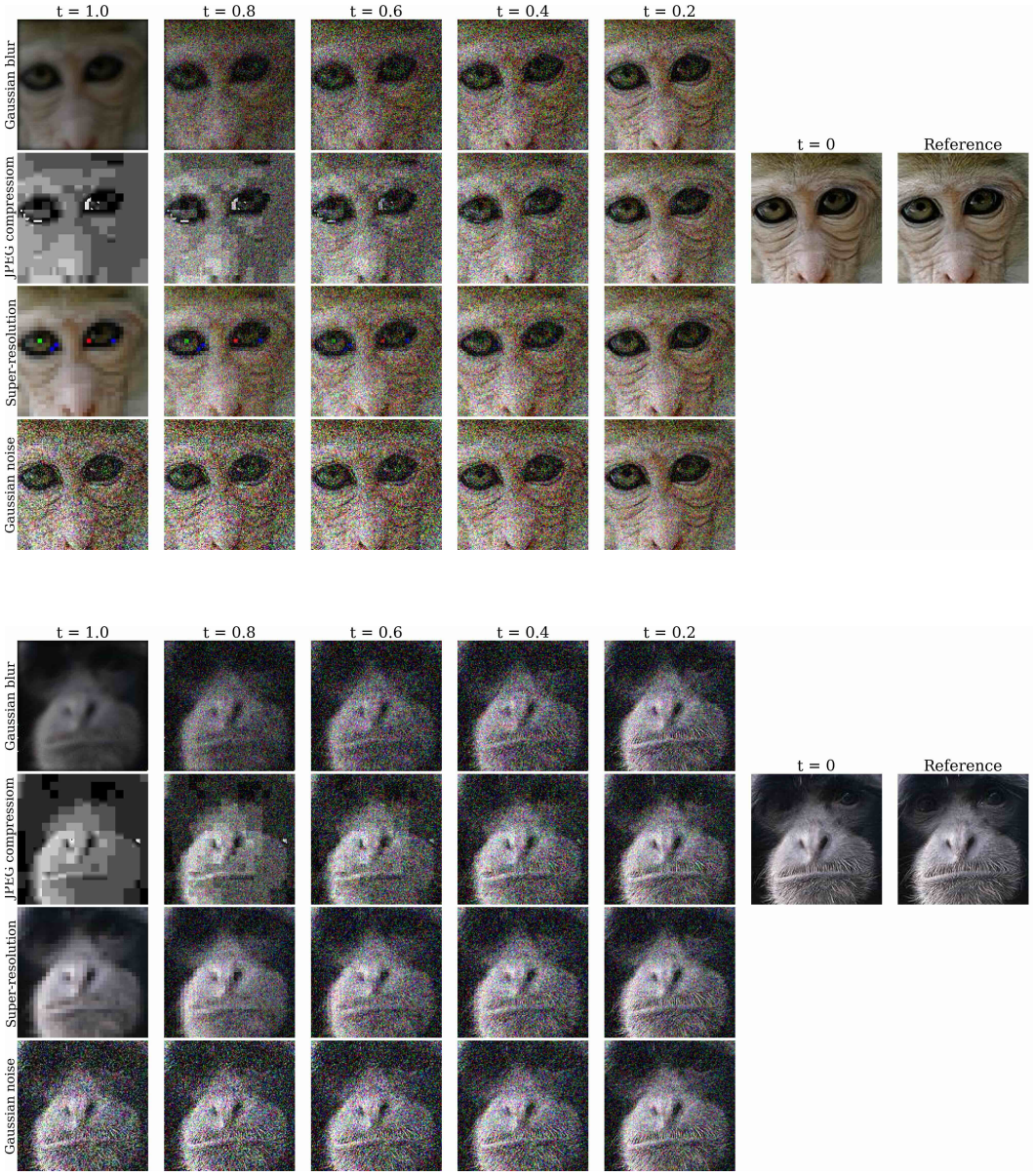


Figure A18: MB-DDPM Generative Process on ImageNet test set. (Part 1)

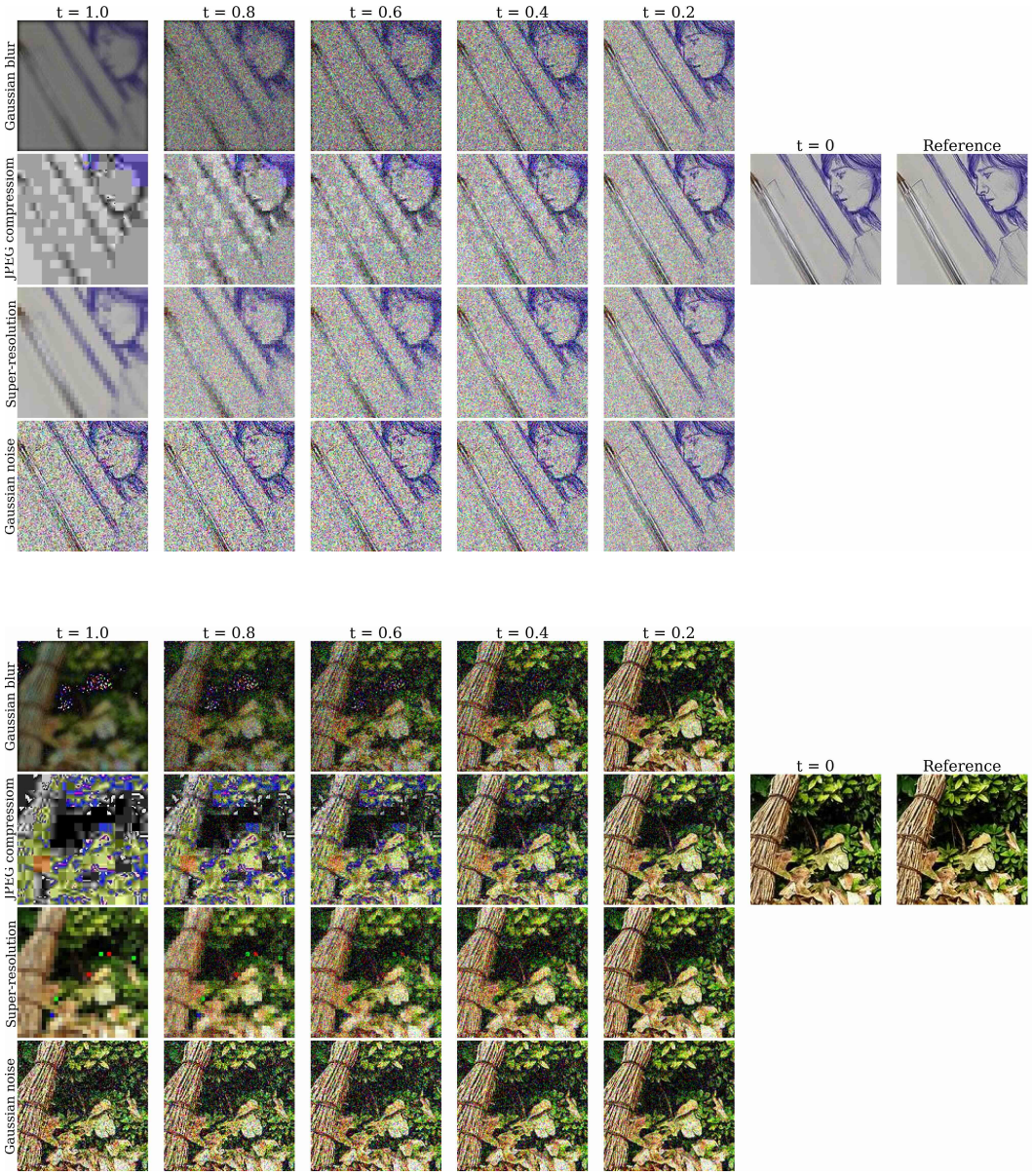


Figure A18: MB-DDPM Generative Process on ImageNet test set. (Part 2)

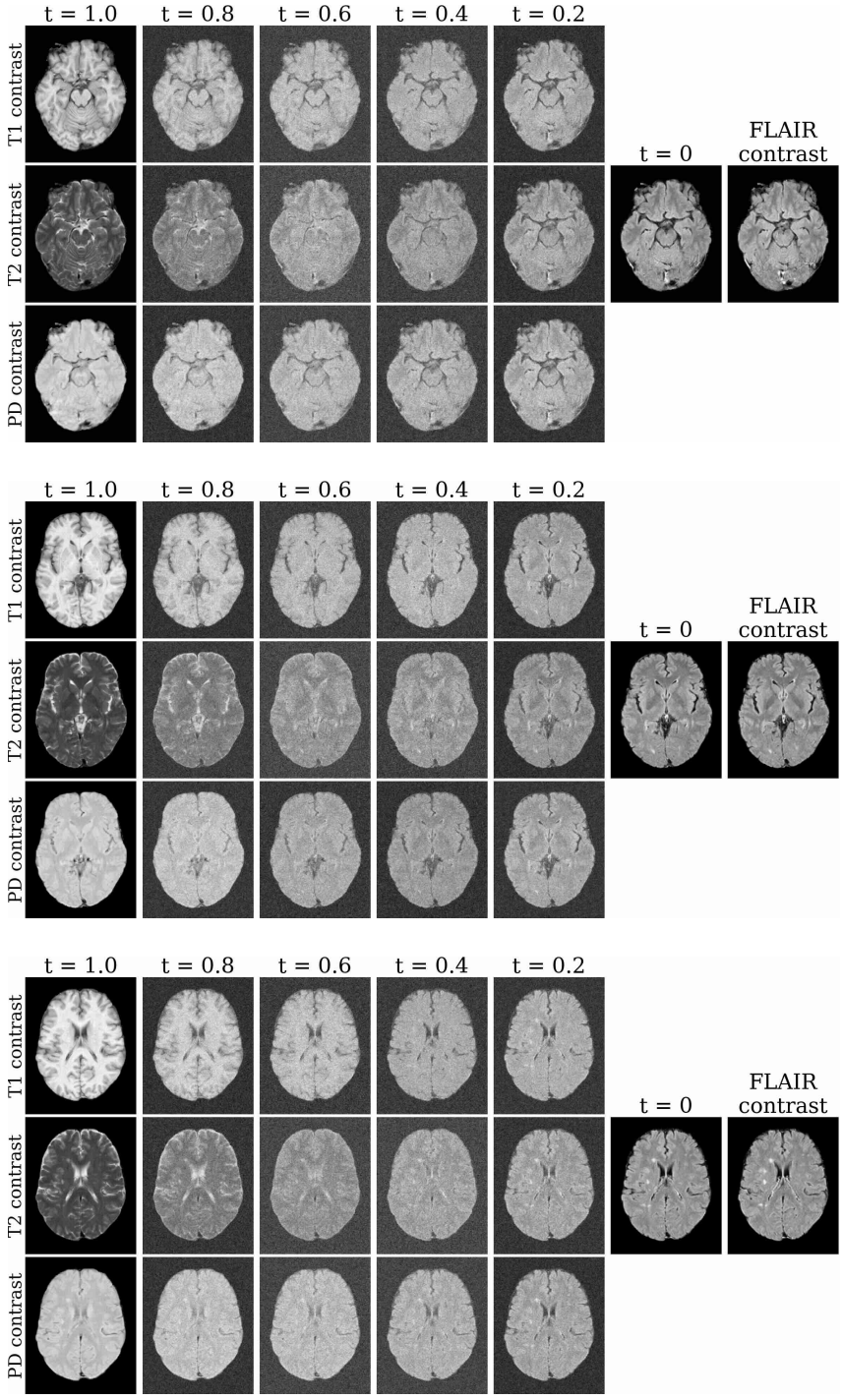


Figure A19: MB-DDPM generative process on the MRI test set. The scan contains 45 slices, of which 10 are shown here.

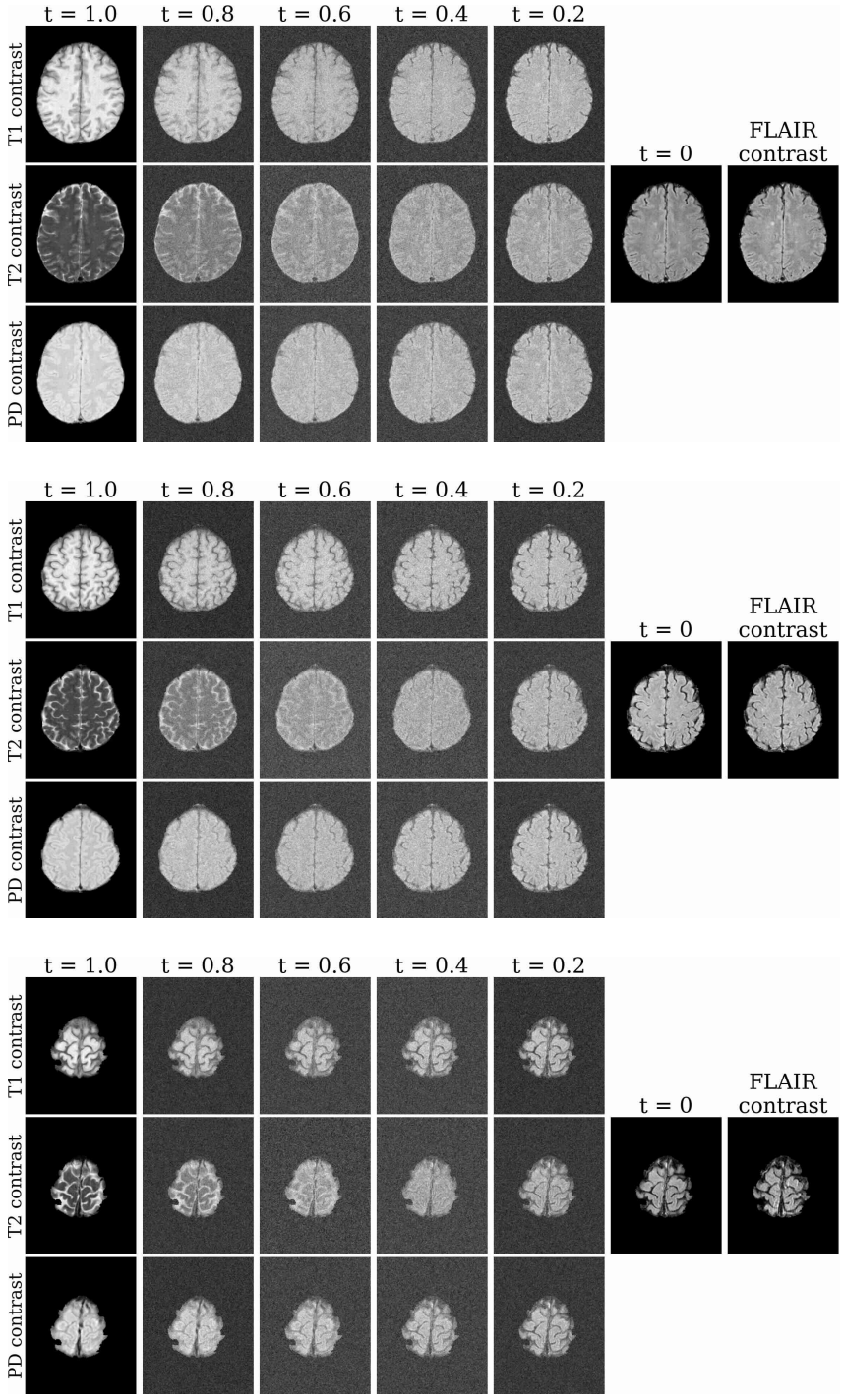


Figure A19: MB-DDPM generative process on the MRI test set. The scan contains 45 slices, of which 10 are shown here. (Part 2)

E.6 Effect of Non-Complementary Sources

MB-DDPM relies on the assumption that sources are complementary. In this appendix, we investigate whether adding a non-complementary source to an otherwise informative source set degrades reconstruction quality, or whether the model can effectively down-weight such a source.

To this end, we conduct an ablation study on CIFAR-10. As the baseline, we use a three-source configuration consisting of Gaussian noise, Gaussian blur, and bicubic super-resolution. We then augment this source set with one additional source corresponding to four challenging scenarios: (i) a heavily corrupted source (JPEG compression with quality factor 5), (ii) a redundant source (a duplicate Gaussian blur observation), (iii) an uninformative source (random uniform noise), and (iv) a conflicting source (a Gaussian-blurred image generated from a different observation).

Table A10 reports the quantitative results, with the values in parentheses indicating the relative change with respect to the baseline three-source configuration.

Overall, the results demonstrate that MB-DDPM is robust to the inclusion of non-complementary sources. In particular, adding a redundant source has a negligible impact on reconstruction performance. Introducing heavily corrupted, uninformative, or conflicting sources leads to only modest degradation, with changes of at most +6.05% in MSE, -0.80% in PSNR, -0.18% in SSIM, +4.32% in FID, and +5.39% in LPIPS. These findings suggest that the model can effectively mitigate the influence of unreliable sources when reconstructing the target image.

Source Images	**MSE** $\downarrow$	**PSNR** $\uparrow$	**SSIM** $\uparrow$	**FID** $\downarrow$	**LPIPS** $\downarrow$
3 sources	0.000661	31.638	0.9642	5.761	0.0420
3 sources + very corrupted	0.000695 (+5.24%)	31.417 (-0.70%)	0.9626 (-0.17%)	5.794 (+0.57%)	0.0438 (+4.15%)
3 sources + redundant	0.000659 (-0.27%)	31.653 (+0.05%)	0.9642 (-0.00%)	5.694 (-1.17%)	0.0421 (+0.11%)
3 sources + uninformative	0.000701 (+6.05%)	31.385 (-0.80%)	0.9625 (-0.18%)	6.010 (+4.32%)	0.0443 (+5.39%)
3 sources + conflicting	0.000695 (+5.14%)	31.420 (-0.69%)	0.9627 (-0.16%)	5.942 (+3.14%)	0.0439 (+4.43%)

Table A10: MSE, PSNR, SSIM, FID, and LPIPS of MB-DDPM on the CIFAR-10 test set (10k samples). The baseline uses three sources (Gaussian blur, $\times 4$ bicubic super-resolution, and Gaussian noise). We then augment the baseline with one additional source corresponding to one of the following scenarios: a heavily corrupted source (JPEG compression, quality factor 5), a redundant source (duplicate Gaussian blur), an uninformative source (random uniform noise), or a conflicting source (Gaussian blur of a different observation). Values in parentheses indicate the relative change with respect to the baseline three-source configuration.

E.7 Handling Missing Sources

In the current formulation, MB-DDPM requires all K sources jointly provided to the neural network during both training and sampling. Consequently, the current architecture does not directly support missing sources at either train or sampling time.

To accommodate a missing source k, we propose to modify the architecture of the network. First, the missing source is represented by setting its input to -1 (or, more generally, to any value that cannot occur in a non-missing input), such that $\mathbf{x}_T^k = -1 \times \mathbf{I}_d$. In addition, a binary indicator y is introduced to specify whether the source is missing, with $y = 1$ if the source is missing (i.e., $\mathbf{x}_T^k = -1 \times \mathbf{I}_d$) and $y = 0$ otherwise. The conditional and marginal networks are modified to take this indicator as an additional input:

$$\boldsymbol{\epsilon}_{\theta_{\text{cond}}}\left(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, y, t\right), \qquad \boldsymbol{\eta}_{\theta_{\text{marg}}}\left(\mathbf{x}_t^{1:K}, y, t\right).$$

The binary missingness indicator is mapped to a learnable embedding vector using a PyTorch embedding layer (`nn.Embedding`). This embedding has dimension $4C^{\text{U-net}}$, matching that of the projected timestep embedding. The two embeddings are then added element-wise to form a combined conditioning vector, which is supplied to the residual blocks throughout the U-Net.

We conduct an experiment to evaluate the robustness of our approach to missing sources. We consider the CIFAR-10 setting described in Section 4.1, which uses four sources: Gaussian noise, Gaussian blur, $4\times$ bicubic super-resolution, and JPEG compression with a quality factor of 5. We investigate three missingness scenarios: (i) all sources are always available, providing the reference baseline; (ii) one source is systematically missing for all samples; and (iii) one source is missing for 50% of the samples and available for the remaining 50%. We choose Gaussian blur as the source to be removed, as our Shapley value analysis in Appendix E.1.3 identified it as an informative source.

For each scenario, we apply the corresponding missingness pattern to both the training and test sets. We train independent neural networks using the resulting training data. We then evaluate each model on the corresponding test set. For the 50% missing setting, we additionally report results separately for samples where Gaussian blur is missing and those where it is available.

The results are shown in Table A11. As expected, removing Gaussian blur, an informative source, leads to a substantial degradation in reconstruction performance. Moreover, this degradation becomes more pronounced as the frequency of missingness increases, with the worst performance observed when the source is systematically missing for all samples. For the 50% missing setting reveals, although performance deteriorates considerably on samples for which Gaussian blur is missing, the degradation is substantially smaller on samples for which the source remains available. For instance, MSE increases by 381.40% on the missing subset, compared with only 38.34% on the non-missing subset, relative to the non-missing baseline. A similar difference is observed for the other reconstruction metrics.

These results highlight the ability of our framework to handle source missingness. In particular, the substantially smaller performance degradation on the non-missing subset demonstrates that the model can adapt to varying source availability rather than suffering uniformly from the presence of missing sources.

Gaussian Blur Source	**MSE** ↓	**PSNR** ↑	**SSIM** ↑	**FID** ↓	**LPIPS** ↓
Non-missing	0.00070	31.417	0.963	5.794	0.044
Missing 100% of times	0.00426 (+512.56%)	23.132 (-26.37%)	0.790 (-17.97%)	27.750 (+378.97%)	0.192 (+339.01%)
Missing 50% times					
Among missing	0.00335 (+381.40%)	24.196 (-22.99%)	0.830 (-13.79%)	-	0.155 (+254.01%)
Among non-missing	0.00096 (+38.34%)	29.993 (-4.53%)	0.950 (-1.27%)	-	0.057 (+29.12%)
Total	0.00215 (+209.77%)	27.096 (-13.75%)	0.890 (-7.53%)	17.111 (+195.35%)	0.106 (+141.50%)

Table A11: MSE, PSNR, SSIM, FID, and LPIPS of MB-DDPM on the CIFAR-10 test set (10k samples) under different Gaussian blur source missingness conditions. Results are reported for the non-missing baseline, and when the source is missing 100% or 50% of the time. For the 50% missing setting, metrics are additionally reported separately for samples where the source is missing, samples where it is available, and the overall test set (Total). FID is not reported for the two subsets separately, as it is computed over the full sample distribution and is not directly comparable across subsets of different sizes. Percentage changes are computed relative to the non-missing baseline.

E.8 Violations of the Source Conditional Independence Assumption

MB-DDPM relies on the assumption that the source observations are conditionally independent given the target observation, as highlighted in (9). In this appendix, we investigate the robustness of the method when this assumption is violated and the source observations exhibit conditional dependencies.

First, the strong empirical performance of MB-DDPM on MRI data (Section 4.2) suggests that moderate deviations from this assumption do not prevent effective reconstruction. In practical settings, such deviations may arise from correlated artifacts caused by patient motion, scanner-specific effects, or acquisition protocols. Despite these potential dependencies, MB-DDPM remains effective, indicating a degree of robustness to mild violations of conditional independence.

Second, we perform a controlled analysis to further characterize this behaviour. Specifically, given two observations of the same target, we generate pairs of Gaussian-noise-corrupted sources with varying levels of correlation conditionally on the target. We consider $K = 2$ sources following the distribution:

$$\mathbf{x}_T^k = \mathbf{x}_0 + \sigma(\sqrt{\rho}\epsilon_s + \sqrt{1-\rho}\epsilon_k), \quad \epsilon_s, \epsilon_k \sim \mathcal{N}(0, \mathbf{I}_d), \quad k = 1, 2, \tag{94}$$

where ρ controls the degree of correlation between the two sources. Increasing ρ progressively violates the conditional independence assumption in (9), with $\rho = 0$ corresponding to conditionally independent observations and larger values representing increasingly correlated degradations.

Table A12 reports the quantitative results, with the values in parentheses indicating the relative change with respect to the baseline configuration using a single Gaussian-noise-corrupted source.

These results clarify the scope of MB-DDPM. The method remains robust to mild violations of the conditional independence assumption, but increasing conditional dependence between sources progressively reduces the benefit of incorporating multiple observations. Under strong conditional dependence, adding an additional source can even degrade performance compared with using a single source.

This behaviour is expected, as strongly conditionally dependent sources provide limited additional information beyond the existing observation while introducing shared noise components that create a mismatch between the assumed and actual joint source distribution.

Source Images	**MSE** $\downarrow$	**PSNR** $\uparrow$	**SSIM** $\uparrow$	**FID** $\downarrow$	**LPIPS** $\downarrow$
One source	0.00548	21.870	0.753	11.926	0.228
Two sources ($\rho = 0$)	0.00472 (-13.83%)	22.512 (+2.93%)	0.773 (+2.61%)	12.551 (+5.25%)	0.217 (-4.59%)
Two sources ($\rho = 0.1$)	0.00554 (+1.10%)	21.773 (-0.44%)	0.753 (-0.11%)	13.533 (+13.48%)	0.252 (+10.84%)
Two sources ($\rho = 0.5$)	0.00594 (+8.51%)	21.511 (-1.64%)	0.726 (-3.62%)	16.110 (+35.09%)	0.255 (+11.97%)

Table A12: MSE, PSNR, SSIM, FID, and LPIPS of MB-DDPM on the CIFAR-10 test set (10k samples). The baseline uses a single Gaussian-noise-corrupted source, while the two-source settings introduce Gaussian noise with increasing conditional dependence controlled by ρ as defined in (94).Values in parentheses indicate the relative change with respect to the baseline single-source configuration.

F Proofs

F.1 Useful Lemmas

We present below several lemmas that will be used throughout the proofs.

Lemma F.1 (Product of Two Gaussians)**.** *Let $\mathbf{x} \in \mathbb{R}^d$, and consider two Gaussian densities*

$$p_1(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \sigma_1^2 \mathbf{I}_d), \qquad p_2(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \sigma_2^2 \mathbf{I}_d). \tag{95}$$

Then the product $p_1(\mathbf{x})p_2(\mathbf{x})$ (up to normalization) is Gaussian:

$$p_1(\mathbf{x})\, p_2(\mathbf{x}) \propto \mathcal{N}\Big(\mathbf{x}; \boldsymbol{\mu}_{1,2}, \sigma_{1,2}^2 \mathbf{I}_d\Big), \tag{96}$$

with

$$\sigma_{1,2}^2 = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1}, \qquad \boldsymbol{\mu}_{1,2} = \sigma_{1,2}^2 \left(\frac{\boldsymbol{\mu}_1}{\sigma_1^2} + \frac{\boldsymbol{\mu}_2}{\sigma_2^2}\right). \tag{97}$$

This follows from standard identities for products of Gaussian densities (see, e.g., [22, Appendix A.2]).

Lemma F.2 (Conditional expectation as the L^2 minimizer)**.** *Let X and Y be square-integrable random variables. Consider the problem of finding a measurable function $f(Y)$ that minimizes*

$$\mathbb{E}_{(X,Y)}\big[\|X - f(Y)\|^2\big]. \tag{98}$$

Then the minimizer is given by the conditional expectation

$$f^*(Y) = \mathbb{E}[X \mid Y]. \tag{99}$$

Moreover, this minimizer is unique almost surely. This is a standard result: the conditional expectation is the L^2-projection onto the $\sigma(Y)$-measurable functions (see, e.g., [9, Chapter 2.4]).

F.2 Proofs for Denoising Diffusion Probabilistic Models (DDPMs)

F.2.1 Proof of Proposition A.1

Proof of Proposition A.1 . The DDPM training objective is obtained by minimizing the variational lower bound (38) repeated below for convenience:

$$\begin{aligned} -\log p_\theta(\mathbf{x}_0) \le \underbrace{D_{\mathrm{KL}}\Big(q(\mathbf{x}_T \mid \mathbf{x}_0) \,\|\, p(\mathbf{x}_T)\Big)}_{\text{Prior matching}} \\ + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\Big[D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\big)\Big]}_{\text{Consistency terms}} \\ + \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}\Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)\Big]}_{\text{Reconstruction term}}. \end{aligned}$$

The bound decomposes into three components: the prior matching term, the consistency terms and the reconstruction term. The prior matching term does not depend on θ and can therefore be ignored when minimizing the bound. We then show that the remaining terms reduce to quadratic losses in the error of the estimator $\mathbf{m}_\theta$.

Consistency terms. For $t = 2, \ldots, T$, we consider

$$D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\big). \tag{100}$$

Both distributions are Gaussian with identical covariance $\sigma_{t-1|0,t}^2 \mathbf{I}_d$, so the KL reduces to squared mean difference:

$$\frac{1}{2\sigma_{t-1|0,t}^2} \Big\| \boldsymbol{\mu}_{t-1|0,t}(\mathbf{x}_0, \mathbf{x}_t) - \boldsymbol{\mu}_{t-1|0,t}\big(\mathbf{m}_\theta(\mathbf{x}_t, t), \mathbf{x}_t\big) \Big\|^2. \tag{101}$$

Using linearity of the mean function $\boldsymbol{\mu}_{t-1|0,t}$ in its first argument, this reduces (up to a multiplicative constant) to

$$\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, t)\|^2 . \tag{102}$$

Taking expectation over $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_0)$ as required by the variational bound, we obtain

$$\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, t)\|^2 \right]. \tag{103}$$

Thus, each intermediate KL term reduces (up to timestep-dependent constants) to a mean squared error in the estimation of $\mathbf{x}_0$.

Reconstruction term. The reconstruction term is

$$\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1) \Big]. \tag{104}$$

The distribution $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)$ is chosen Gaussian with fixed covariance σ_{rec}^2, this term therefore reduces to (up to a multiplicative constant) to a quadratic loss

$$\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \left[\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_1, 1)\|^2 \right]. \tag{105}$$

Conclusion. Combining both components, we conclude that, up to multiplicative constants independent of θ, minimizing the full variational bound is equivalent to minimizing

$$\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, t)\|^2 \right]. \tag{106}$$

Integrating over the data distribution, minimizing the expected variational bound is equivalent to minimizing

$$\sum_{t=1}^{T} \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0), \mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)} \left[\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, t)\|^2 \right]. \tag{107}$$

Applying Lemma (F.2) with $X = \mathbf{x}_0$ and $Y = \mathbf{x}_t$, we obtain that the optimal predictor satisfies

$$\mathbf{m}_\theta^*(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t], \tag{108}$$

for $t = 1, \ldots, T$.

□

F.2.2 Proof of Proposition A.2

Proof of Proposition A.2. We begin with the objective $\mathcal{J}_{\text{DDPM}}$ (43) repeated here for convenience:

$$\mathcal{J}_{\text{DDPM}}(\theta) = \mathbb{E}_{t \sim \text{Unif}\{1,\ldots,T\},\, \mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0),\, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\big(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon},\, t\big)\|^2 \Big].$$

For a fixed timestep $t \in \{1, \ldots, T\}$, the DDPM objective can be written as

$$\mathcal{J}_{\text{DDPM},t}(\theta) = \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0),\, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon}, t)\|^2 \Big]. \tag{109}$$

Using the identity

$$\boldsymbol{\epsilon} = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \bar{\alpha}_t}},$$

obtained from the decomposition in (35), the objective can be rewritten as

$$\mathcal{J}_{\text{DDPM},t}(\theta) = \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0),\, \mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)} \left[\left\| \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \bar{\alpha}_t}} - \boldsymbol{\epsilon}_\theta\big(\mathbf{x}_t,\, t\big) \right\|^2 \right]. \tag{110}$$

Applying Lemma F.2 with $X = \mathbf{x}_0$ and $Y = \mathbf{x}_t$, the unique minimizer is

$$\boldsymbol{\epsilon}_\theta^*(\mathbf{x}_t, t) = \mathbb{E}\left[\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \bar{\alpha}_t}} \Big| \mathbf{x}_t \right].$$

Finally, the conditional expectation can be explicitly computed:

$$\mathbb{E}\left[\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}\Big|\mathbf{x}_t\right] = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]}{\sqrt{1-\bar{\alpha}_t}}.$$

Plugging this into the definition of the estimator $\hat{\mathbf{x}}_0$ in (42) immediately yields

$$\hat{\mathbf{x}}_0^*(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta^*(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}} = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t],$$

which proves the claim. □

F.3 Proofs for Denoising Diffusion Probabilistic Models with Fixed Endpoints

F.3.1 Proof of Propositions B.1 and B.2

Proof of Proposition B.1. The forward diffusion bridge transition distribution

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T)$$

is defined as the conditional distribution of $\mathbf{x}_t$ given both the previous state $\mathbf{x}_{t-1}$ and the terminal state $\mathbf{x}_T$ for $t = 1, \dots, T-1$. By definition of conditional probability and the Markov property,

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T) = \frac{q(\mathbf{x}_t, \mathbf{x}_T \mid \mathbf{x}_{t-1})}{q(\mathbf{x}_T \mid \mathbf{x}_{t-1})} = \frac{q(\mathbf{x}_T \mid \mathbf{x}_t)\, q(\mathbf{x}_t \mid \mathbf{x}_{t-1})}{q(\mathbf{x}_T \mid \mathbf{x}_{t-1})},$$

for $t = 1, \dots, T-1$. From the form of the forward diffusion process transitions in (33), both $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and $q(\mathbf{x}_T \mid \mathbf{x}_t)$ are Gaussian:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\,\mathbf{x}_{t-1}, (1-\alpha_t)\,\mathbf{I}_d),$$
$$q(\mathbf{x}_T \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_T; \sqrt{\tilde{\alpha}_t}\,\mathbf{x}_t, (1-\tilde{\alpha}_T)\,\mathbf{I}_d).$$

By properties of Gaussians,

$$q(\mathbf{x}_T \mid \mathbf{x}_t) \propto \exp\left(-\frac{1}{2(1-\tilde{\alpha}_t)}\left\|\mathbf{x}_T - \sqrt{\tilde{\alpha}_t}\mathbf{x}_t\right\|^2\right) = \exp\left(-\frac{\tilde{\alpha}_t}{2(1-\tilde{\alpha}_t)}\left\|\frac{1}{\sqrt{\tilde{\alpha}_t}}\mathbf{x}_T - \mathbf{x}_t\right\|^2\right)$$

Thus, both $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and $q(\mathbf{x}_t \mid \mathbf{x}_T)$ are Gaussian (up to a normalization) in the same variable $\mathbf{x}_t$. By Lemma F.1, their product is again Gaussian with covariance given by

$$\sigma^2_{t|t-1,T} = \Big(\frac{1}{1-\alpha_t} + \frac{\tilde{\alpha}_t}{1-\tilde{\alpha}_t}\Big)^{-1} = \frac{(1-\alpha_t)(1-\tilde{\alpha}_t)}{1-\tilde{\alpha}_{t-1}}$$

and mean given by

$$\begin{aligned}\boldsymbol{\mu}_{t|t-1,T}(\mathbf{x}_{t-1}, \mathbf{x}_T) &= \sigma^2_{t|t-1,T}\Big(\frac{\sqrt{\alpha_t}}{1-\alpha_t}\,\mathbf{x}_{t-1} + \frac{\sqrt{\tilde{\alpha}_t}}{1-\tilde{\alpha}_t}\,\mathbf{x}_T\Big) \\ &= \frac{\sqrt{\alpha_t}(1-\tilde{\alpha}_t)}{1-\tilde{\alpha}_{t-1}}\,\mathbf{x}_{t-1} + \frac{\sqrt{\tilde{\alpha}_t}(1-\alpha_t)}{1-\tilde{\alpha}_{t-1}}\,\mathbf{x}_T.\end{aligned}$$

This concludes the proof. □

Proof of Proposition B.2. The forward diffusion bridge marginal distribution

$$q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)$$

is defined as the conditional distribution of $\mathbf{x}_t$ given both the initial state $\mathbf{x}_0$ and the terminal state $\mathbf{x}_T$ for $t = 1, \dots, T-1$. By definition of conditional probability and the Markov property,

$$q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) = \frac{q(\mathbf{x}_t, \mathbf{x}_T \mid \mathbf{x}_0)}{q(\mathbf{x}_T \mid \mathbf{x}_0)} = \frac{q(\mathbf{x}_T \mid \mathbf{x}_t)\, q(\mathbf{x}_t \mid \mathbf{x}_0)}{q(\mathbf{x}_T \mid \mathbf{x}_0)},$$

for $t = 1, \dots, T-1$. From the form of the forward diffusion process transitions in (33), both $q(\mathbf{x}_t \mid \mathbf{x}_0)$ and $q(\mathbf{x}_T \mid \mathbf{x}_t)$ are Gaussian:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\,\mathbf{I}_d),$$
$$q(\mathbf{x}_T \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_T; \sqrt{\tilde{\alpha}_T}\,\mathbf{x}_t, (1-\tilde{\alpha}_T)\,\mathbf{I}_d).$$

By properties of Gaussians,

$$q(\mathbf{x}_T \mid \mathbf{x}_t) \propto \exp\left(-\frac{1}{2(1-\tilde{\alpha}_t)}\left\|\mathbf{x}_T - \sqrt{\tilde{\alpha}_t}\mathbf{x}_t\right\|^2\right) = \exp\left(-\frac{\tilde{\alpha}_t}{2(1-\tilde{\alpha}_t)}\left\|\frac{1}{\sqrt{\tilde{\alpha}_t}}\mathbf{x}_T - \mathbf{x}_t\right\|^2\right)$$

Thus, both $q(\mathbf{x}_t \mid \mathbf{x}_0)$ and $q(\mathbf{x}_t \mid \mathbf{x}_T)$ are Gaussian (up to a normalization) in the same variable $\mathbf{x}_t$. By Lemma F.1, their product is again Gaussian with covariance given by

$$\sigma^2_{t|0,T} = \left(\frac{1}{1-\bar{\alpha}_t} + \frac{\tilde{\alpha}_t}{1-\tilde{\alpha}_t}\right)^{-1} = \frac{(1-\bar{\alpha}_t)(1-\tilde{\alpha}_t)}{1-\bar{\alpha}_T}$$

and mean given by

$$\boldsymbol{\mu}_{t|0,T}(\mathbf{x}_0, \mathbf{x}_T) = \sigma^2_{t|0,T}\Big(\frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t}\,\mathbf{x}_0 + \frac{\sqrt{\tilde{\alpha}_t}}{1-\tilde{\alpha}_t}\,\mathbf{x}_T\Big) = \frac{\sqrt{\bar{\alpha}_t}(1-\tilde{\alpha}_t)}{1-\bar{\alpha}_T}\,\mathbf{x}_0 + \frac{\sqrt{\tilde{\alpha}_t}(1-\bar{\alpha}_t)}{1-\bar{\alpha}_T}\,\mathbf{x}_T.$$

This concludes the proof. □

F.3.2 Proof of Proposition B.3

Proof of Proposition B.3. By definition of the bridge model likelihood (54),

$$\begin{aligned}
p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T) &= \int p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T)\, d\mathbf{x}_{1:(T-1)} \\
&= \int q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T) \frac{p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T)}{q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T)}\, d\mathbf{x}_{1:(T-1)} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[\frac{p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T)}{q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T)}\right].
\end{aligned}$$

Applying Jensen's inequality yields the variational bound

$$-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T) \le \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T)}{q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T)}\right].$$

Next, we expand the numerator and denominator using the definitions of the joint distribution of the bridge FDP process (50) and the joint distribution of the bridge RDP (53). This yields

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:(T-1)} \mid \mathbf{x}_T)}{q(\mathbf{x}_{1:(T-1)} \mid \mathbf{x}_0, \mathbf{x}_T)}\right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T) - \sum_{t=2}^{T-1} \log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T)} - \log \frac{p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)}{q(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{x}_T)}\right]
\end{aligned}$$

We use Bayes' rule to express the bridge FDP transition:

$$\begin{aligned}
q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T) &= \frac{q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T) q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_T)}{q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)} \\
&\Rightarrow q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_T) = \frac{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T) q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)}{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_T)},
\end{aligned}$$

for $t = 2, \ldots, T-1$. We identify that by Markov chain's properties, $q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T) = q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)$. Regrouping terms, the bound decomposes into three components:

$$\begin{aligned}
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T) - \sum_{t=2}^{T-1} \log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)}{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)} \frac{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_T)}{q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)} - \log \frac{p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)}{q(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{x}_T)}\right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log \frac{p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)}{q(\mathbf{x}_{T-1} \mid \mathbf{x}_0, \mathbf{x}_T)} - \sum_{t=2}^{T-1} \log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)}{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)\right] \\
&= \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log \frac{p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)}{q(\mathbf{x}_{T-1} \mid \mathbf{x}_0, \mathbf{x}_T)}\right]}_{\text{Terminal Consistency Term}} + \sum_{t=2}^{T-1} \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)}{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t)}\right]}_{\text{Consistency Terms}} \\
&\quad + \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)\right]}_{\text{Reconstruction Term}}
\end{aligned}$$

We now identify each term.

Terminal Consistency Term. Marginalizing out all variables except $\mathbf{x}_{T-1}$ yields

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log\frac{p_\theta(\mathbf{x}_{T-1}\mid\mathbf{x}_T)}{q(\mathbf{x}_{T-1}\mid\mathbf{x}_0,\mathbf{x}_T)}\right]\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{T-1}\mid\mathbf{x}_T)}{q(\mathbf{x}_{T-1}\mid\mathbf{x}_0,\mathbf{x}_T)}q(\mathbf{x}_{1:(T-1)}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{1:(T-1)}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{T-1}\mid\mathbf{x}_T)}{q(\mathbf{x}_{T-1}\mid\mathbf{x}_0,\mathbf{x}_T)}\left(\int q(\mathbf{x}_{1:(T-1)}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{1:(T-2)}\right)d\mathbf{x}_{T-1}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{T-1}\mid\mathbf{x}_T)}{q(\mathbf{x}_{T-1}\mid\mathbf{x}_0,\mathbf{x}_T)}q(\mathbf{x}_{T-1}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{T-1}\\
&=D_{\mathrm{KL}}\Big(q(\mathbf{x}_{T-1}\mid\mathbf{x}_0,\mathbf{x}_T)\,\|\,p_\theta(\mathbf{x}_{T-1}\mid\mathbf{x}_T)\Big)
\end{aligned}
$$

Consistency Terms. For each $t=2,\dots,T-1$, marginalizing over all variables except $(\mathbf{x}_{t-1},\mathbf{x}_t)$ gives

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)}{q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)}\right]\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)}{q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)}q(\mathbf{x}_{1:(T-1)}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{1:(T-1)}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)}{q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)}\left(\int q(\mathbf{x}_{1:(T-1)}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{1,\dots,t-2,t+1,\dots,(T-1)}\right)d\mathbf{x}_{t-1,t}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)}{q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)}q(\mathbf{x}_{t-1},\mathbf{x}_t\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{t-1,t}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)}{q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)}q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t,\mathbf{x}_T)q(\mathbf{x}_t\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{t-1,t}\\
&=\int\left(\int-\log\frac{p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)}{q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)}q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)d\mathbf{x}_{t-1}\right)q(\mathbf{x}_t\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_t\\
&=\int D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)\,\|\,p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)\big)q(\mathbf{x}_t\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_t\\
&=\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)}\Big[D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1}\mid\mathbf{x}_0,\mathbf{x}_t)\,\|\,p_\theta(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_T)\big)\Big].
\end{aligned}
$$

Reconstruction Term. Finally, marginalizing all variables except $\mathbf{x}_1$ yields

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1,\mathbf{x}_T)\right]\\
&=\int-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1,\mathbf{x}_T)q(\mathbf{x}_{1:(T-1)}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{1:(T-1)}\\
&=\int-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1,\mathbf{x}_T)\left(q(\mathbf{x}_{1:(T-1)}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{2:(T-1)}\right)d\mathbf{x}_1\\
&=\int-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1,\mathbf{x}_T)q(\mathbf{x}_1\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_1\\
&=\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0,\mathbf{x}_T)}\left[-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1,\mathbf{x}_T)\right].
\end{aligned}
$$

Combining the three components yields the stated variational bound. □

F.3.3 Proof of Proposition B.4

Proof of Proposition B.4 . The Bridge DDPM's training objective is obtained by minimizing the variational lower bound (56) repeated below for convenience:

$$
\begin{aligned}
-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T) \leq & \underbrace{D_{\mathrm{KL}}\Big(q(\mathbf{x}_{T-1} \mid \mathbf{x}_0, \mathbf{x}_T) \,\|\, p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)\Big)}_{\text{Terminal Consistency Term}} \\
& + \sum_{t=2}^{T-1} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)}\Big[D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)\big)\Big]}_{\text{Consistency Terms}} \\
& + \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0,\mathbf{x}_T)}\Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)\Big]}_{\text{Reconstruction term}}.
\end{aligned}
$$

We show that the terms reduce to quadratic losses in the error of the estimator $\mathbf{m}_\theta$.

Terminal Consistency Terms. The terminal term is

$$
D_{\mathrm{KL}}\Big(q(\mathbf{x}_{T-1} \mid \mathbf{x}_0, \mathbf{x}_T) \,\|\, p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)\Big). \tag{111}
$$

Both distributions are Gaussian with identical covariance $\sigma^2_{T-1|0,T}\mathbf{I}_d$, so the KL reduces to squared mean difference:

$$
\frac{1}{2\sigma^2_{T-1|0,T}} \left\| \boldsymbol{\mu}_{T-1|0,T}(\mathbf{x}_0, \mathbf{x}_T) - \boldsymbol{\mu}_{T-1|0,T}\big(\mathbf{m}_\theta(\mathbf{x}_T, \mathbf{x}_T, T-1), \mathbf{x}_T\big) \right\|^2. \tag{112}
$$

Using linearity the mean function $\boldsymbol{\mu}_{T-1|0,T}$ in its first argument, this reduces (up to a multiplicative constant) to

$$
\left\| \mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_T, \mathbf{x}_T, T-1) \right\|^2. \tag{113}
$$

Consistency terms. For $t = 2, \ldots, T-1$, we consider

$$
D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T)\big). \tag{114}
$$

Both distributions are Gaussian with identical covariance $\sigma^2_{t-1|0,t}\mathbf{I}_d$, so the KL reduces to squared mean difference:

$$
\frac{1}{2\sigma^2_{t-1|0,t}} \left\| \boldsymbol{\mu}_{t-1|0,t}(\mathbf{x}_0, \mathbf{x}_t) - \boldsymbol{\mu}_{t-1|0,t}\big(\mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t), \mathbf{x}_t\big) \right\|^2. \tag{115}
$$

Again, using linearity of the mean function $\boldsymbol{\mu}_{t-1|0,t}$ in its first argument, this reduces (up to a multiplicative constant) to

$$
\left\| \mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t) \right\|^2. \tag{116}
$$

Taking expectation over $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)$ as required by the variational bound we obtain

$$
\mathbb{E}_{\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\left\| \mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t) \right\|^2 \right]. \tag{117}
$$

Reconstruction term. The final term is

$$
\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0,\mathbf{x}_T)}\Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)\Big]. \tag{118}
$$

The distribution $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1, \mathbf{x}_T)$ is chosen Gaussian with fixed covariance σ^2_{rec}, this term therefore is proportional to a quadratic loss

$$
\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0,\mathbf{x}_T)} \left[\left\| \mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_1, \mathbf{x}_T, 1) \right\|^2 \right]. \tag{119}
$$

Conclusion. Combining the consistency term and the reconstruction terms, we conclude that, up to multiplicative constants independent of θ, minimizing the variational bound is equivalent to minimizing

$$\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_T, \mathbf{x}_T, T-1)\|^2 + \sum_{t=1}^{T-1} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t)\|^2 \right]. \tag{120}$$

And integrating across the data distribution, minimizing the expected variational bound is equivalent to minimizing

$$\mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T)} \left[\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_T, \mathbf{x}_T, T-1)\|^2 \right] + \sum_{t=1}^{T-1} \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T), \mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t, \mathbf{x}_T, t)\|^2 \right]. \tag{121}$$

Applying Lemma (F.2) with $X = \mathbf{x}_0$ and $Y = \mathbf{x}_T$, we obtain that the optimal predictor for the terminal step satisfies

$$\mathbf{m}^*_\theta(\mathbf{x}_T, \mathbf{x}_T, t-1) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_T]. \tag{122}$$

Further, applying Lemma (F.2) with $X = \mathbf{x}_0$ and $Y = (\mathbf{x}_t, \mathbf{x}_T)$, we obtain that the optimal predictor satisfies

$$\mathbf{m}^*_\theta(\mathbf{x}_t, \mathbf{x}_T, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{x}_T]. \tag{123}$$

for $t = 1, \ldots, T-1$.

□

F.3.4 Proof of Proposition B.5

Proof of Proposition B.5. We begin with the objective $\mathcal{J}_{\text{B-cond}}$ (62) repeated here for convenience:

$$\mathcal{J}_{\text{B-cond}}(\theta_{\text{cond}}) = \mathbb{E}_{t\sim\text{Unif}\{1,\ldots,T-1\},\, (\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T),\, \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}} \left(C^{(0)}_{t|0,T}\mathbf{x}_0 + C^{(T)}_{t|0,T}\mathbf{x}_T + \sigma_{t|0,T}\boldsymbol{\epsilon}, \mathbf{x}_T, t \right) \right\|^2 \right].$$

For a fixed timestep t, the objective can be written as

$$\mathcal{J}_{\text{B-cond},t}(\theta_{\text{cond}}) = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T),\, \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_d)} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}} \left(C^{(0)}_{t|0,T}\mathbf{x}_0 + C^{(T)}_{t|0,T}\mathbf{x}_T + \sigma_{t|0,T}\boldsymbol{\epsilon}, \mathbf{x}_T, t \right) \right\|^2 \right]. \tag{124}$$

Using the identity

$$\boldsymbol{\epsilon} = \frac{\mathbf{x}_t - C^{(0)}_{t|0,T}\mathbf{x}_0 - C^{(T)}_{t|0,T}\mathbf{x}_T}{\sigma_{t|0,T}},$$

obtained from the decomposition in (52), the objective can be rewritten as

$$\mathcal{J}_{\text{B-cond},t}(\theta_{\text{cond}}) = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T),\, \mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\left\| \frac{\mathbf{x}_t - C^{(0)}_{t|0,T}\mathbf{x}_0 - C^{(T)}_{t|0,T}\mathbf{x}_T}{\sigma_{t|0,T}} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}\left(\mathbf{x}_t, \mathbf{x}_T,\, t\right) \right\|^2 \right]. \tag{125}$$

Applying Lemma (F.2) with $X = \mathbf{x}_0$ and $Y = (\mathbf{x}_t, \mathbf{x}_T)$, the unique minimizer is

$$\boldsymbol{\epsilon}^*_{\theta_{\text{cond}}}(\mathbf{x}_t, \mathbf{x}_T, t) = \mathbb{E}\left[\frac{\mathbf{x}_t - C^{(0)}_{t|0,T}\mathbf{x}_0 - C^{(T)}_{t|0,T}\mathbf{x}_T}{\sigma_{t|0,T}} \middle| \mathbf{x}_t, \mathbf{x}_T \right]. \tag{126}$$

Finally, the conditional expectation can be explicitly computed:

$$\mathbb{E}\left[\frac{\mathbf{x}_t - C^{(0)}_{t|0,T}\mathbf{x}_0 - C^{(T)}_{t|0,T}\mathbf{x}_T}{\sigma_{t|0,T}} \middle| \mathbf{x}_t, \mathbf{x}_T \right] = \frac{\mathbf{x}_t - C^{(0)}_{t|0,T}\,\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{x}_T] - C^{(T)}_{t|0,T}\mathbf{x}_T}{\sigma_{t|0,T}}. \tag{127}$$

Plugging this into the definition of the estimator $\hat{\mathbf{x}}_0^{\text{cond}}$ in (61) immediately yields

$$\hat{\mathbf{x}}_0^{\text{cond}}(\mathbf{x}_t, \mathbf{x}_T, t) = \frac{\mathbf{x}_t - C_{t|0,T}^{(T)} \mathbf{x}_T - \sigma_{t|0,T}\, \boldsymbol{\epsilon}^*_{\theta_{\text{cond}}}(\mathbf{x}_t, \mathbf{x}_T, t)}{C_{t|0,T}^{(0)}} = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{x}_T]. \tag{128}$$

which proves the claim. □

F.3.5 Proof of Proposition B.6

Proof of Proposition B.6. We begin with the objective $\mathcal{J}_{\text{B-marg}}$ (68) repeated here for convenience:

$$\mathcal{J}_{\text{B-marg}}(\theta_{\text{marg}}) = \mathbb{E}_{t\sim\text{Unif}\{1,\dots,T-1\},\, (\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T),\, \mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\left\| \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t) \right\|^2 \right],$$

For a fixed timestep t, the objective can be written as

$$\mathcal{J}_{\text{B-marg},t}(\theta_{\text{marg}}) = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T)\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T),\, \mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0,\mathbf{x}_T)} \left[\left\| \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t) \right\|^2 \right]. \tag{129}$$

To rewrite this objective, define the marginal and conditional data distributions

$$p_{\text{data}}(\mathbf{x}_0) = \int p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T)\, d\mathbf{x}_T, \qquad p_{\text{data}}(\mathbf{x}_T \mid \mathbf{x}_0) = \frac{p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T)}{p_{\text{data}}(\mathbf{x}_0)}. \tag{130}$$

Next, define the marginal forward distribution of $\mathbf{x}_t$ given $\mathbf{x}_0$ by integrating out $\mathbf{x}_T$:

$$q_{\text{marg}}(\mathbf{x}_t \mid \mathbf{x}_0) = \int q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\, p_{\text{data}}(\mathbf{x}_T \mid \mathbf{x}_0)\, d\mathbf{x}_T. \tag{131}$$

Using the law of total expectation, the objective can then be written as

$$\mathcal{J}_{\text{B-marg},t}(\theta_{\text{marg}}) = \mathbb{E}_{\mathbf{x}_0\sim p_{\text{data}}(\mathbf{x}_0),\, \mathbf{x}_t\sim q_{\text{marg}}(\mathbf{x}_t|\mathbf{x}_0)} \left[\left\| \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t, t) \right\|^2 \right]. \tag{132}$$

Applying Lemma F.2 with $X = \mathbf{x}_0$ and $Y = \mathbf{x}_t$, the unique minimizer is

$$\boldsymbol{\eta}^*_{\theta_{\text{marg}}}(\mathbf{x}_t, t) = \mathbb{E}_{\text{marg}} \left[\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} \,\middle|\, \mathbf{x}_t \right]. \tag{133}$$

Finally, the conditional expectation can be explicitly computed:

$$\mathbb{E}_{\text{marg}} \left[\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} \,\middle|\, \mathbf{x}_t \right] = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\, \mathbb{E}_{\text{marg}}[\mathbf{x}_0 \mid \mathbf{x}_t]}{\sqrt{1-\bar{\alpha}_t}}. \tag{134}$$

Plugging this into the definition of the estimator $\hat{\mathbf{x}}_0^{\text{marg}}$ in (67) immediately yields

$$\hat{\mathbf{x}}_0^{\text{marg},*}(\mathbf{x}_t, t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\, \boldsymbol{\eta}^*_{\theta_{\text{marg}}}(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}} = \mathbb{E}_{\text{marg}}[\mathbf{x}_0 \mid \mathbf{x}_t], \tag{135}$$

which proves the claim. □

F.4 Proofs for Multi-Bridge Denoising Diffusion Probabilistic Models

F.4.1 Proof of Proposition 3.1

Proof of Proposition 3.1. By definition of the multi-bridge likelihood (16),

$$\begin{aligned}
p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T^{1:K}) &= \int p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K})\, d\mathbf{x}_{1:(T-1)}^{1:K} \\
&= \int q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\, d\mathbf{x}_{1:(T-1)}^{1:K} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})} \left[\frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})} \right].
\end{aligned}$$

Applying Jensen's inequality yields the variational bound

$$-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T^{1:K}) \leq \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[-\log \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\right].$$

Next, we expand the numerator and denominator using the definitions of the joint distribution of the multi-bridge FDP (12) and the joint distribution of the multi-bridge RDP (15). This yields

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[-\log \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\right] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[-\log p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K}) - \sum_{t=2}^{T-1} \log \frac{p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_{t-1}^{1:K}, \mathbf{x}_T^{1:K})} - \log \frac{p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\right]
\end{aligned}$$

We use Bayes' rule to express the multi-bridge FDP transition

$$\begin{aligned}
q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}) &= \frac{q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_{t-1}^{1:K}, \mathbf{x}_T^{1:K}) q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})} \\
&\Rightarrow q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_{t-1}^{1:K}, \mathbf{x}_T^{1:K}) = \frac{q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}) q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})},
\end{aligned}$$

for $t = 2, \ldots, T-1$. We identify that by Markov chain's properties, $q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}) = q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K})$. Regrouping terms, the bound decomposes into three components:

$$\begin{aligned}
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\Bigg[- \log p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K}) - \sum_{t=2}^{T-1} \log \frac{p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K})} \frac{q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})} \\
&\qquad\qquad - \log \frac{p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\Bigg] \\
&= \mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[- \log \frac{p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})} - \sum_{t=2}^{T-1} \log \frac{p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K})} - \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})\right] \\
&= \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[- \log \frac{p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\right]}_{\text{Terminal Consistency Term}} + \sum_{t=2}^{T-1} \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[- \log \frac{p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K})}\right]}_{\text{Consistency Terms}} \\
&\quad + \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[- \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})\right]}_{\text{Reconstruction Term}}
\end{aligned}$$

We now identify each term.

Terminal Consistency Term. Marginalizing out all variables except $\mathbf{x}_{T-1}^{1:K}$ yields

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[- \log \frac{p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\right] \\
&= \int - \log \frac{p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})} q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) d\mathbf{x}_{1:(T-1)}^{1:K} \\
&= \int - \log \frac{p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})} \left(\int q(\mathbf{x}_{1:(T-1)}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) d\mathbf{x}_{1:(T-2)}^{1:K} \right) d\mathbf{x}_{T-1}^{1:K} \\
&= \int - \log \frac{p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})}{q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})} q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) d\mathbf{x}_{T-1}^{1:K} \\
&= D_{\mathrm{KL}}\Big(q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K}) \,\|\, p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K}) \Big)
\end{aligned}$$

Consistency Terms. For each $t = 2, \dots, T-1$, marginalizing over all variables except $(\mathbf{x}_{t-1}^{1:K}, \mathbf{x}_t^{1:K})$ gives

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[-\log\frac{p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})}\right]\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})}q(\mathbf{x}_{1:(T-1)}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T^{1:K})d\mathbf{x}_{1:(T-1)}^{1:K}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})}\left(\int q(\mathbf{x}_{1:(T-1)}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{1,\dots,t-2,t+1,\dots,(T-1)}^{1:K}\right)d\mathbf{x}_{t-1,t}^{1:K}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})}q(\mathbf{x}_{t-1}^{1:K},\mathbf{x}_t^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T^{1:K})d\mathbf{x}_{t-1,t}^{1:K}\\
&=\int-\log\frac{p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})}q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})q(\mathbf{x}_t^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T^{1:K})d\mathbf{x}_{t-1,t}^{1:K}\\
&=\int\left(\int-\log\frac{p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})}{q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})}q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})d\mathbf{x}_{t-1}^{1:K}\right)q(\mathbf{x}_t^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T^{1:K})d\mathbf{x}_t^{1:K}\\
&=\int D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})\,\|\,p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})\big)q(\mathbf{x}_t^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T^{1:K})d\mathbf{x}_t^{1:K}\\
&=\mathbb{E}_{q(\mathbf{x}_t^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\Big[D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_t^{1:K})\,\|\,p_\theta(\mathbf{x}_{t-1}^{1:K}\mid\mathbf{x}_t^{1:K},\mathbf{x}_T^{1:K})\big)\Big].
\end{aligned}
$$

Reconstruction Term. Finally, marginalizing all variables except $\mathbf{x}_1$ yields

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}_{1:(T-1)}^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1^{1:K},\mathbf{x}_T^{1:K})\right]\\
&=\int-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1^{1:K},\mathbf{x}_T^{1:K})\;q(\mathbf{x}_{1:(T-1)}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T^{1:K})d\mathbf{x}_{1:(T-1)}\\
&=\int-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1^{1:K},\mathbf{x}_T^{1:K})\left(q(\mathbf{x}_{1:(T-1)}^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T)d\mathbf{x}_{2:(T-1)}^{1:K}\right)d\mathbf{x}_1^{1:K}\\
&=\int-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1^{1:K},\mathbf{x}_T^{1:K})\;q(\mathbf{x}_1^{1:K}\mid\mathbf{x}_0,\mathbf{x}_T^{1:K})d\mathbf{x}_1^{1:K}\\
&=\mathbb{E}_{q(\mathbf{x}_1^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})}\left[-\log p_\theta(\mathbf{x}_0\mid\mathbf{x}_1^{1:K},\mathbf{x}_T^{1:K})\right].
\end{aligned}
$$

Combining the three components yields the stated variational bound. □

F.4.2 Proof of Proposition 3.2

To prove Proposition 3.2, we proceed in two steps: first, we characterize the optimal estimator that minimizes the variational bound; second, we show that the conditional estimator attains this optimum.

Step 1. The following proposition characterizes the optimal estimator that minimizes the expected variational bound (19) over the data distribution (10).

Proposition F.1 (Variational Bound Minimization for MB-DDPM)**.** *Let* $(\mathbf{x}_0, \mathbf{x}_T^{1:K}) \sim p_{\mathrm{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K})$ *as in* (10)*, and let* $\mathbf{x}_t^{1:K} \sim q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})$ *as in* (13)*. Assume that the multi-bridge RDP transitions are Gaussian with mean parameterized by an estimator* $\mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)$ *as in* (20)*,* (21)*, and* (22)*. Then the expected variational bound* (19) *is minimized if and only if for* $t = 1, \dots, T-1$

$$
\mathbf{m}_\theta^*(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}]. \tag{136}
$$

Proof of Proposition F.1 . The Multi-Bridge DDPM's training objective is obtained by minimizing the variational upper bound (19) repeated below for convenience:

$$
\begin{aligned}
-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_T) \leq & \underbrace{D_{\mathrm{KL}}\left(q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^k) \,\|\, p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})\right)}_{\text{Terminal Consistency Term}} \\
& + \sum_{t=2}^{T-1} \underbrace{\mathbb{E}_{q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\left[D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^k) \,\|\, p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})\right)\right]}_{\text{Consistency Terms}} \\
& + \underbrace{\mathbb{E}_{q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})\Big]}_{\text{Reconstruction term}}.
\end{aligned}
$$

We show that the terms reduce to quadratic losses in the error of the estimator $\mathbf{m}_\theta$.

Terminal Consistency Terms. The terminal term is

$$
D_{\mathrm{KL}}\left(q(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^k) \,\|\, p_\theta(\mathbf{x}_{T-1}^{1:K} \mid \mathbf{x}_T^{1:K})\right). \tag{137}
$$

Both distributions are Gaussian with identical covariance $\sigma^2_{T-1|0,T}\mathbf{I}_{dK}$, so the KL reduces to squared mean difference:

$$
\frac{1}{2\sigma^2_{T-1|0,T}} \left\| \boldsymbol{\mu}_{T-1|0,T}(\mathbf{A}\mathbf{x}_0, \mathbf{x}_T^{1:K}) - \boldsymbol{\mu}_{T-1|0,T}\big(\mathbf{A}\mathbf{m}_\theta(\mathbf{x}_T^{1:K}, \mathbf{x}_T^{1:K}, T-1), \mathbf{x}_T^{1:K}\big) \right\|^2. \tag{138}
$$

Using linearity of the mean function $\boldsymbol{\mu}_{T-1|0,T}$ in its first argument, this reduces (up to a multiplicative constant) to

$$
\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_T^{1:K}, \mathbf{x}_T^{1:K}, T-1)\right\|^2. \tag{139}
$$

Consistency terms. For $t = 2, \ldots, T-1$, we consider

$$
D_{\mathrm{KL}}\big(q(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_0, \mathbf{x}_t^{1:K}) \,\|\, p_\theta(\mathbf{x}_{t-1}^{1:K} \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})\big). \tag{140}
$$

Both distributions are Gaussian with identical covariance $\sigma^2_{t-1|0,t}\mathbf{I}_{dK}$, so the KL reduces to squared mean difference:

$$
\frac{1}{2\sigma^2_{t-1|0,t}} \left\| \boldsymbol{\mu}_{t-1|0,t}(\mathbf{A}\mathbf{x}_0, \mathbf{x}_t^{1:K}) - \boldsymbol{\mu}_{t-1|0,t}\big(\mathbf{A}\mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t), \mathbf{x}_t^{1:K}\big) \right\|^2. \tag{141}
$$

Again, using linearity of the mean function $\boldsymbol{\mu}_{t-1|0,t}$ in its first argument, this reduces (up to a multiplicative constant) to

$$
\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)\right\|^2. \tag{142}
$$

Taking expectation over $\mathbf{x}_t^{1:K} \sim q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})$ as required by the variational bound we obtain

$$
\mathbb{E}_{\mathbf{x}_t^{1:K} \sim q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\left[\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)\right\|^2\right]. \tag{143}
$$

Reconstruction term. The final term is

$$
\mathbb{E}_{q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\Big[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})\Big]. \tag{144}
$$

The distribution $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K})$ is chosen Gaussian with fixed covariance σ^2_{rec}, this term therefore is proportional to a quadratic loss

$$
\mathbb{E}_{q(\mathbf{x}_1^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})}\left[\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_1^{1:K}, \mathbf{x}_T^{1:K}, 1)\right\|^2\right]. \tag{145}
$$

Conclusion. Combining the consistency term and the reconstruction terms, we conclude that, up to multiplicative constants independent of θ, minimizing the variational bound is equivalent to minimizing

$$\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_T^{1:K}, \mathbf{x}_T^{1:K}, T-1)\right\|^2 + \sum_{t=1}^{T-1} \mathbb{E}_{q(\mathbf{x}_t^{1:K}|\mathbf{x}_0, \mathbf{x}_T^{1:K})}\left[\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)\right\|^2\right]. \quad (146)$$

And integrating across the data distribution, minimizing the expected variational bound is equivalent to minimizing

$$\begin{aligned}&\mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_T^{1:K}) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K})}\left[\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_T^{1:K}, \mathbf{x}_T^{1:K}, T-1)\right\|^2\right] \\ &\quad + \sum_{t=1}^{T-1} \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_T^{1:K}) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K}), \mathbf{x}_t^{1:K} \sim q(\mathbf{x}_t^{1:K}|\mathbf{x}_0, \mathbf{x}_T^{1:K})}\left[\left\|\mathbf{x}_0 - \mathbf{m}_\theta(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)\right\|^2\right]. \quad (147)\end{aligned}$$

Applying Lemma F.2 with $X = \mathbf{x}_0$ and $Y = \mathbf{x}_T^{1:K}$, we obtain that the optimal predictor for the terminal step satisfies

$$\mathbf{m}_\theta^*(\mathbf{x}_T^{1:K}, \mathbf{x}_T^{1:K}, t-1) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_T^{1:K}]. \quad (148)$$

Further, applying Lemma F.2 with $X = \mathbf{x}_0$ and $Y = (\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})$, we obtain that the optimal predictor satisfies

$$\mathbf{m}_\theta^*(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}]. \quad (149)$$

for $t = 1, \ldots, T-1$.

□

It is important to note that at the terminal step $t = T$, we cannot directly construct an estimator that recovers $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_T^{1:K}]$, as this conditional depends on the unknown data distribution we aim to model. Instead, we approximate $\mathbf{x}_{T-1}^{1:K}$ with $\mathbf{x}_T^{1:K}$ and reuse the estimator for the penultimate step, which is optimal for $\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_{T-1}^{1:K}, \mathbf{x}_T^{1:K}]$.

Step 2. The following proposition shows that minimizing the noise-prediction objective (24) yields an estimator that achieves the variational bound minimum established in Proposition F.1.

Proposition F.2 (Conditional Estimator as Posterior Mean for MB-DDPM)**.** *For fixed $t = 1, \ldots, T-1$, let $\boldsymbol{\epsilon}^*_{\theta_{\text{cond}}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)$ denote the minimizer of* (24)*. Then, in the infinite-data and infinite-capacity limit,*

$$\boldsymbol{\epsilon}^*_{\theta_{cond}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) = \frac{\mathbf{x}_t^{1:K} - C^{(0)}_{t|0,T}\,\mathbf{A}\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}] - C^{(T)}_{t|0,T}\mathbf{x}_T^{1:K}}{\sigma_{t|0,T}}. \quad (150)$$

Consequently, the induced estimator (23)

$$\hat{\mathbf{x}}_0^{cond,\star}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) := \frac{\sum_{k=1}^K \left(\mathbf{x}_t^k - C^{(T)}_{t|0,T}\mathbf{x}_T^k\right) - \sigma_{t|0,T}\,\mathbf{A}^\top \boldsymbol{\epsilon}^\star_{\theta_{cond}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)}{K C^{(0)}_{t|0,T}}, \quad (151)$$

recovers the posterior mean,

$$\hat{\mathbf{x}}_0^{cond,*}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}]. \quad (152)$$

Proof of Proposition F.2. We begin with the objective $\mathcal{J}_{\text{cond}}$ (24) repeated here for convenience:

$$\begin{aligned}\mathcal{J}_{\text{cond}}(\theta_{\text{cond}}) = \mathbb{E}_{t \sim \text{Unif}\{1,\ldots,T-1\},\, (\mathbf{x}_0, \mathbf{x}_T^{1:K}) \sim p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K}),\, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{dK})} \\ \left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}\left(C^{(0)}_{t|0,T}\mathbf{A}\mathbf{x}_0 + C^{(T)}_{t|0,T}\mathbf{x}_T^{1:K} + \sigma_{t|0,T}\boldsymbol{\epsilon}, \mathbf{x}_T^{1:K}, t\right)\right\|^2\right]\end{aligned}$$

For a fixed timestep t, the objective can be written as

$$\mathcal{J}_{\text{cond},t}(\theta_{\text{cond}}) = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T^{1:K})\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T^{1:K}),\,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_{dK})} \left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}\left(C_{t|0,T}^{(0)}\mathbf{A}\mathbf{x}_0 + C_{t|0,T}^{(T)}\mathbf{x}_T^{1:K} + \sigma_{t|0,T}\boldsymbol{\epsilon}, \mathbf{x}_T^{1:K}, t\right)\right\|^2\right]$$

Using the identity

$$\boldsymbol{\epsilon} = \frac{\mathbf{x}_t^{1:K} - C_{t|0,T}^{(0)}\mathbf{A}\mathbf{x}_0 - C_{t|0,T}^{(T)}\mathbf{x}_T^{1:K}}{\sigma_{t|0,T}},$$

obtained from the decomposition in (14), the objective can be rewritten as

$$\mathcal{J}_{\text{cond},t}(\theta_{\text{cond}}) = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T^{1:K})\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T^{1:K}),\,\mathbf{x}_t^{1:K}\sim q(\mathbf{x}_t^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})} \left[\left\|\frac{\mathbf{x}_t^{1:K} - C_{t|0,T}^{(0)}\mathbf{A}\mathbf{x}_0 - C_{t|0,T}^{(T)}\mathbf{x}_T^{1:K}}{\sigma_{t|0,T}} - \boldsymbol{\epsilon}_{\theta_{\text{cond}}}\left(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t\right)\right\|^2\right]$$

Applying Lemma F.2 with $X = \mathbf{x}_0$ and $Y = (\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K})$, the unique minimizer is

$$\boldsymbol{\epsilon}^*_{\theta_{\text{cond}}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) = \mathbb{E}\left[\frac{\mathbf{x}_t^{1:K} - C_{t|0,T}^{(0)}\mathbf{A}\mathbf{x}_0 - C_{t|0,T}^{(T)}\mathbf{x}_T^{1:K}}{\sigma_{t|0,T}}\middle|\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}\right].$$

Finally, the conditional expectation can be explicitly computed:

$$\mathbb{E}\left[\frac{\mathbf{x}_t^{1:K} - C_{t|0,T}^{(0)}\mathbf{A}\mathbf{x}_0 - C_{t|0,T}^{(T)}\mathbf{x}_T^{1:K}}{\sigma_{t|0,T}}\middle|\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}\right] = \frac{\mathbf{x}_t^{1:K} - C_{t|0,T}^{(0)}\,\mathbf{A}\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}] - C_{t|0,T}^{(T)}\mathbf{x}_T^{1:K}}{\sigma_{t|0,T}}.$$

Plugging this into the definition of the estimator $\hat{\mathbf{x}}_0^{\text{cond}}$ in (23) immediately yields

$$\hat{\mathbf{x}}_0^{\text{cond},*}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t) = \frac{\sum_{k=1}^K\left(\mathbf{x}_t^k - C_{t|0,T}^{(T)}\mathbf{x}_T^k\right) - \sigma_{t|0,T}\,\mathbf{A}^\top\boldsymbol{\epsilon}^*_{\theta_{\text{cond}}}(\mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}, t)}{KC_{t|0,T}^{(0)}} = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}, \mathbf{x}_T^{1:K}].$$

which proves the claim. □

F.4.3 Proof of Proposition 3.3

Proof of Proposition 3.3. We begin with the objective $\mathcal{J}_{\text{marg}}$ (29) repeated here for convenience:

$$\mathcal{J}_{\text{marg}}(\theta_{\text{marg}}) = \mathbb{E}_{t\sim\text{Unif}\{1,\ldots,T-1\},\,(\mathbf{x}_0,\mathbf{x}_T^{1:K})\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T^{1:K}),\,\mathbf{x}_t^{1:K}\sim q(\mathbf{x}_t^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})} \left[\left\|\frac{\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t)\right\|^2\right],$$

For a fixed timestep t, the objective can be written as

$$\mathcal{J}_{\text{marg},t}(\theta_{\text{marg}}) = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_T^{1:K})\sim p_{\text{data}}(\mathbf{x}_0,\mathbf{x}_T^{1:K}),\,\mathbf{x}_t^{1:K}\sim q(\mathbf{x}_t^{1:K}|\mathbf{x}_0,\mathbf{x}_T^{1:K})} \left[\left\|\frac{\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t)\right\|^2\right], \tag{153}$$

To rewrite this objective, remember the joint data distribution (10) factorizes as,

$$p_{\text{data}}(\mathbf{x}_0, \mathbf{x}_T^{1:K}) = p_{\text{data}}(\mathbf{x}_0)\prod_{k=1}^K p_{\text{data}}^k(\mathbf{x}_T^k \mid \mathbf{x}_0). = p_{\text{data}}(\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_T^{1:K} \mid \mathbf{x}_0).$$

Next, define the marginal forward distribution of $\mathbf{x}_t^{1:K}$ given $\mathbf{x}_0$ by integrating out $\mathbf{x}_T^{1:K}$:

$$q_{\text{marg}}(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0) = \int q(\mathbf{x}_t^{1:K} \mid \mathbf{x}_0, \mathbf{x}_T^{1:K})\, p_{\text{data}}(\mathbf{x}_T^{1:K} \mid \mathbf{x}_0)\, d\mathbf{x}_T^{1:K}. \tag{154}$$

Then, the mean squared error objective can be written as

$$\mathcal{J}_{\mathrm{marg},t}(\theta_{\text{marg}}) = \mathbb{E}_{\mathbf{x}_0 \sim p_{\mathrm{data}}(\mathbf{x}_0),\, \mathbf{x}_t^{1:K} \sim q_{\text{marg}}(\mathbf{x}_t^{1:K} | \mathbf{x}_0)} \left[\left\| \frac{\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} - \boldsymbol{\eta}_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t) \right\|^2 \right].$$

Applying Lemma F.2 with $X = \mathbf{x}_0$ and $Y = \mathbf{x}_t^{1:K}$, the unique minimizer is

$$\boldsymbol{\eta}^*_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t) = \mathbb{E}_{\text{marg}} \left[\frac{\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} \,\middle|\, \mathbf{x}_t^{1:K} \right].$$

Finally, the conditional expectation can be explicitly computed:

$$\mathbb{E}_{\text{marg}} \left[\frac{\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\mathbf{A}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} \,\middle|\, \mathbf{x}_t \right] = \frac{\mathbf{x}_t^{1:K} - \sqrt{\bar{\alpha}_t}\,\mathbf{A}\mathbb{E}_{\text{marg}}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}]}{\sqrt{1-\bar{\alpha}_t}}.$$

Plugging this into the definition of the estimator $\hat{\mathbf{x}}_0^{\text{marg}}$ in (28) immediately yields

$$\hat{\mathbf{x}}_0^{\text{marg},*}(\mathbf{x}_t^{1:K}, t) = \frac{\sum_{k=1}^K \mathbf{x}_t^k - \sqrt{1-\bar{\alpha}_t}\,\mathbf{A}^\top \boldsymbol{\eta}^*_{\theta_{\text{marg}}}(\mathbf{x}_t^{1:K}, t)}{\sqrt{\bar{\alpha}_t}} = \mathbb{E}_{\text{marg}}[\mathbf{x}_0 \mid \mathbf{x}_t^{1:K}],$$

which proves the claim. □