LEARNING DIFFUSION BRIDGE FLOW MAPS WITH CONTINUOUS-TIME CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion Bridge Models (DBMs) provide a diffusion-based framework for image-to-image (I2I) translation. However, high-quality sampling often requires tens of costly neural-network evaluations. Recent consistency-based approaches accelerate sampling by learning direct mappings from intermediate bridge states $\boldsymbol{x}_t$ to the data endpoint $\boldsymbol{x}_0$, but do not learn the entire bridge flow. In this work, we develop a single consistency model that represents the entire finite-time bridge flow, mapping any intermediate state $\boldsymbol{x}_t$ directly to any earlier state $\boldsymbol{x}_s$. Learning the full bridge flow poses two main challenges: how to represent the flow, and how to train it stably in continuous time. We address these issues by using the exact solution of the bridge PF-ODE to structure the learned map, and by deriving a bridge-specific sign correction that prevents the continuous-time update from amplifying flow-map inconsistency. On held-out ImageNet-512 super-resolution and deblurring, BFM reduces FID-VAE over reproduced CDBM by 39.5% and 23.2%, respectively, at the same 4-NFE budget; at 8 NFEs, it reaches FID-VAE comparable to DBMSolver using 20-30 NFEs.

1 INTRODUCTION

Diffusion Bridge Models (DBMs) (Zhou et al., 2023; Liu et al., 2023) provide a principled framework for image-to-image (I2I) translation by modeling stochastic processes between arbitrary endpoint distributions. This formulation supports tasks such as image super-resolution, deblurring, and depth-to-image generation. However, sampling DBMs to generate images can be expensive, as it requires repeated costly neural-network evaluations. Specialized fast bridge solvers (Zheng et al., 2024; Venugopal et al., 2026) substantially reduce this cost, but high-quality generation can still require tens of function evaluations, making high-resolution translation computationally expensive.

Consistency-based bridge methods address this bottleneck by learning direct mappings from intermediate bridge states $\boldsymbol{x}_t$ to the data endpoint $\boldsymbol{x}_0$ (He et al., 2024; Gushchin et al., 2025; Hou et al., 2026). These endpoint mappings can replace much of the iterative sampling procedure, enabling generation with only a few neural-network evaluations. However, the learned mapping is restricted to predicting $\boldsymbol{x}_0$ from $\boldsymbol{x}_t$. More generally, the Bridge PF-ODE induces an entire family of finite-time maps from a state $\boldsymbol{x}_t$ at source time t to a state $\boldsymbol{x}_s$ at destination time s, where $0 \leq s \leq t$, conditioned on the endpoint $\boldsymbol{x}_T$. Existing endpoint-based bridge models do not directly represent this full family of maps. Learning this family enables direct transitions to intermediate times and provides consistency supervision across the entire bridge rather than only at the data endpoint.

Recent flow-map models learn mappings between arbitrary times along an ODE trajectory (Kim et al., 2023; 2024; Geng et al., 2026; Sabour et al., 2025; Boffi et al., 2026). Diffusion bridges, however, possess additional analytic structure: their finite-time solution combines the current state $\boldsymbol{x}_t$, the conditioning endpoint $\boldsymbol{x}_T$, and an interval-dependent denoising quantity through known time-dependent coefficients. We find that a generic average-velocity parameterization exhibits poor conditioning in this setting, particularly over long source-destination intervals. We further find that directly applying continuous-time consistency does not always minimize the flow-map inconsistency: for some source-destination pairs, the update causes severe degradation. This sign-reversal mechanism is specific to the bridge-structured parameterization, unlike prior works (Sabour et al., 2025). These observations motivate us to develop a bridge-structured parameterization as well as a continuous-time training rule that corrects the resulting sign reversal.

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

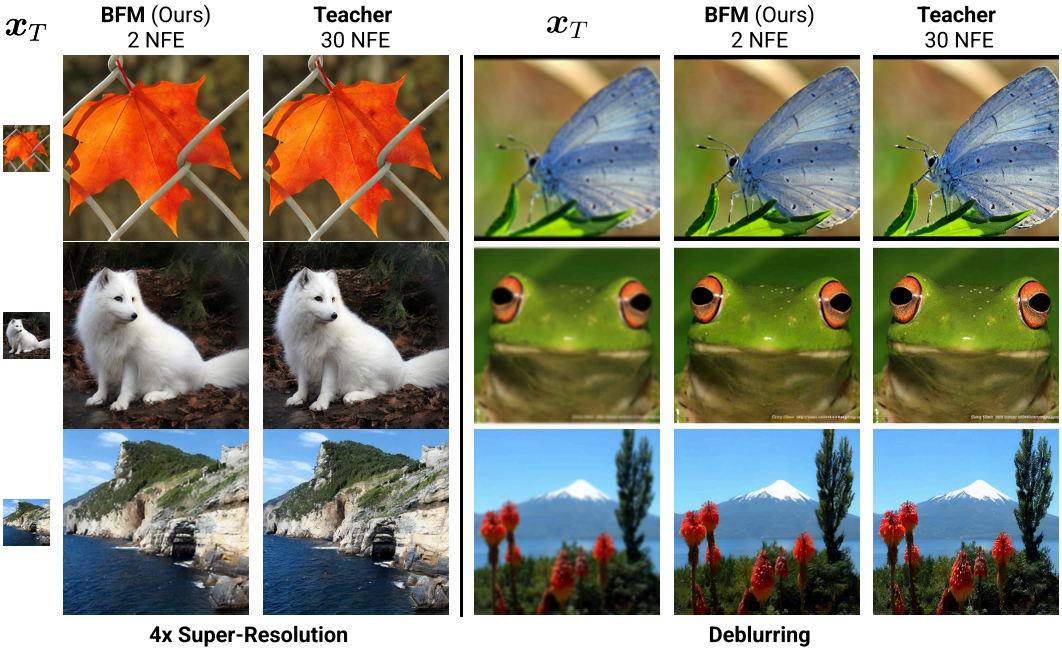

Figure 1: Bridge Flow Model (BFM) outputs on ImageNet-512 super-resolution and Gaussian deblurring. At only 2 NFEs, BFM recovers much of the fine structure produced by the 30-NFE teacher DBM with DBMSolver (Venugopal et al., 2026)–**a 15× reduction in model evaluations**–across diverse content from wildlife and natural landscapes to fine textures such as foliage.

We introduce **Bridge Flow Models (BFMs)**, a continuous-time consistency model that directly maps a bridge state $\boldsymbol{x}_t$ to $\boldsymbol{x}_s$ for any destination time $s \leq t$, given the conditioning endpoint $\boldsymbol{x}_T$. We structure this map according to the exact finite-time solution of the Bridge PF-ODE: known analytic coefficients account for the contributions of $\boldsymbol{x}_t$ and $\boldsymbol{x}_T$, while the neural network represents the remaining interval-dependent denoising quantity. We train the resulting flow map through fixed-destination continuous-time consistency and introduce a sign correction that prevents the update from amplifying the resulting flow-map inconsistency. Our contributions are summarized as follows:

- We formulate diffusion-bridge acceleration as learning the full conditional flow of the Bridge PF-ODE, and introduce a bridge-structured parameterization based on its exact finite-time solution. The parameterization analytically incorporates the known bridge coefficients, recovers the identity map at $s = t$ and endpoint prediction at $s = 0$, and represents arbitrary destinations with a single model.
- We specialize continuous-time consistency training to arbitrary-destination bridge flow maps and identify a bridge-specific sign reversal that can amplify rather than reduce the flow-map inconsistency. We derive a simple correction and show in a controlled ablation that it eliminates the resulting severe degradation.
- We validate BFM through controlled parameterization, sign-correction, full-flow, intermediate-destination, and flow-composition experiments across ImageNet (512×512) restoration, Edges2Handbags, and DIODE. Full-flow supervision improves endpoint sampling even when every flow-map call during inference sets $s = 0$, and BFM at 8 NFEs achieves ImageNet FID-VAE comparable to DBMSolver using 20–30 NFEs.

2 Background and Related Work

2.1 Diffusion Bridges

Diffusion Bridge Models (Zhou et al., 2023) generalize the diffusion framework to stochastic processes connecting two arbitrary distributions, p_{prior} and p_{data}. Given endpoints $\boldsymbol{x}_0 \sim p_{\text{data}}$ and $\boldsymbol{x}_T \sim p_{\text{prior}}$, DBMs use Doob's h-transform (Doob, 1984) to condition a diffusion process to reach $\boldsymbol{x}_T$ almost surely at terminal time T. Conditioned on two endpoints, the resulting *diffusion bridge*

has intermediate marginals $\boldsymbol{x}_t \sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0, \boldsymbol{x}_T)$, and admits the Bridge Probability Flow (PF) ODE:

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(t)\boldsymbol{x}_t - g(t)^2 \left[\frac{1}{2}\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_T) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_T|\boldsymbol{x}_t)\right], \tag{1}$$

where $p(\boldsymbol{x}_T|\boldsymbol{x}_t)$ is the transition kernel from time t to T and is analytically known. Here, $f(t) = \frac{\mathrm{d}}{\mathrm{d}t}\log\alpha_t$ and $g(t)^2 = \frac{\mathrm{d}}{\mathrm{d}t}\sigma_t^2 - 2\sigma_t^2\frac{\mathrm{d}}{\mathrm{d}t}\log\alpha_t$ are the drift and diffusion coefficients, respectively, where α_t and σ_t specify the signal and noise schedules of the diffusion process. A DBM learns the conditional score $\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t|\boldsymbol{x}_T)$ via denoising bridge score matching (Zhou et al., 2023). DBIM (Zheng et al., 2024) and DBMSolver (Venugopal et al., 2026) accelerate Bridge PF-ODE sampling through specialized numerical updates, but remain iterative and require multiple NFEs.

2.2 ODE Flow Maps

Consider a general time-dependent ODE: $\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = v(\boldsymbol{x}_t, t)$. For times $s \le t$, we define its *flow map* $\Phi_{t\to s}(\boldsymbol{x}_t)$ as the state at time s obtained by solving the ODE from the initial state $\boldsymbol{x}_t$ at time t. When the ODE admits a unique solution, its flow map satisfies

$$\Phi_{s\to r}(\Phi_{t\to s}(\boldsymbol{x}_t)) = \Phi_{t\to r}(\boldsymbol{x}_t), \qquad r \le s \le t,$$

together with the identity property $\Phi_{t\to t}(\boldsymbol{x}_t) = \boldsymbol{x}_t$. Thus, we define *trajectory* to refer to the entire continuous curve connecting $\boldsymbol{x}_t$ and $\boldsymbol{x}_s$; and a flow map maps one point on this trajectory to another.

For Diffusion Bridges, the ODE in Equation (1) is additionally conditioned on the terminal endpoint $\boldsymbol{x}_T$. We therefore define its *conditional flow map* as $\Phi_{t\to s}(\boldsymbol{x}_t; \boldsymbol{x}_T)$. In Section 3, we seek to learn this conditional flow map via consistency training.

2.3 Endpoint-Mapping Models

Consistency Models (CMs) (Song et al., 2023; Song & Dhariwal, 2023) learn a *consistency function* that maps any intermediate state $\boldsymbol{x}_t$ on a PF-ODE trajectory directly to the data endpoint $\boldsymbol{x}_0$. In the flow-map notation of Section 2.2, we can write this as $f_\theta(\boldsymbol{x}_t, t) \approx \Phi_{t\to 0}(\boldsymbol{x}_t)$. Accordingly, a valid CM satisfies the *self-consistency* property $f_\theta(\boldsymbol{x}_t, t) = f_\theta(\boldsymbol{x}_s, s)$, where $\boldsymbol{x}_s = \Phi_{t\to s}(\boldsymbol{x}_t)$ for time $s \in [0, t)$, together with the *boundary condition* $f_\theta(\boldsymbol{x}_0, 0) = \boldsymbol{x}_0$.

Discrete-time consistency. Standard CM training enforces the self-consistency property (Song et al., 2023). Given time $s = t - \Delta t$ with $\Delta t > 0$, a discrete-time objective is

$$\mathcal{L}_{\mathrm{disc}} = \mathbb{E}\left[w(t)\, d(f_\theta(\boldsymbol{x}_t, t), f_{\theta^-}(\boldsymbol{x}_s, s))\right],$$

where θ^- is a stop-gradient copy of the model, $\boldsymbol{x}_s$ lies on the same PF-ODE trajectory as $\boldsymbol{x}_t$, $w(t)$ is the weighting function, and $d(\cdot,\cdot)$ is a distance function, such as the squared Euclidean distance. In the diffusion-bridge setting, CDBM (He et al., 2024) applies discrete-time training to learn the mapping from intermediate bridge states to the data endpoint, corresponding to $\Phi_{t\to 0}(\boldsymbol{x}_t; \boldsymbol{x}_T)$. While not consistency-based, IBMD (Gushchin et al., 2025) similarly distills bridge transport into a few-step endpoint mapping via inverse bridge matching, but requires 3× the GPU VRAM.

Continuous-time consistency. For the squared Euclidean distance, the gradient of the discrete objective admits a continuous-time limit as $\Delta t \to 0$ (Song et al., 2023; Lu & Song, 2024):

$$\nabla_\theta \mathbb{E}\left[w(t) f_\theta(\boldsymbol{x}_t, t)^\top \frac{\mathrm{d}}{\mathrm{d}t} f_{\theta^-}(\boldsymbol{x}_t, t)\right],$$

where $\frac{\mathrm{d}}{\mathrm{d}t} f_{\theta^-}(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} f_{\theta^-}(\boldsymbol{x}_t, t)\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \frac{\partial}{\partial t} f_{\theta^-}(\boldsymbol{x}_t, t)$ is the *total derivative* of the consistency function along the PF-ODE trajectory. In the bridge setting, E-Bridge (Hou et al., 2026) applies continuous-time consistency to a redesigned restoration bridge.

2.4 Flow-Map Models

Beyond standard CMs that learn mappings from intermediate $\boldsymbol{x}_t$ to data endpoint $\boldsymbol{x}_0$, recent works generalize this idea to learn mappings between arbitrary times along an ODE trajectory.

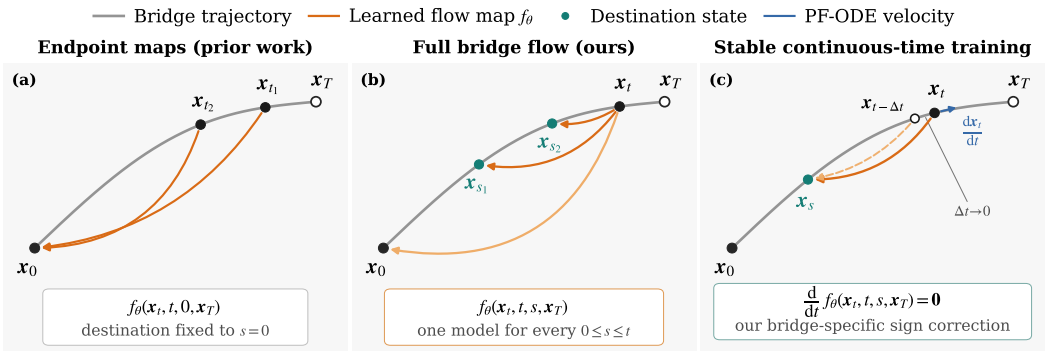

Figure 2: Prior bridge-consistency models learn only the endpoint map $f_\theta(\boldsymbol{x}_t, t, 0, \boldsymbol{x}_T)$ (left). Our Bridge Flow Model (BFM) instead learns a single bridge flow map $f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)$ for arbitrary $0 \le s \le t$, with endpoint prediction recovered as the special case $s = 0$ (middle). For a fixed destination s, mappings from neighboring source states along the same Bridge PF-ODE trajectory must agree. Taking $\Delta t \to 0$ yields the continuous-time consistency condition, whose bridge flow tangent we evaluate via a JVP and stabilize via our bridge-specific sign correction (right; please refer to Sec. D and E for the derivations).

Consistency Trajectory Models (CTMs) (Kim et al., 2023) learn a mapping $G_\theta(\boldsymbol{x}_t, t, s) \approx \Phi_{t\to s}(\boldsymbol{x}_t)$ where $s < t$, enabling direct traversal between arbitrary points of the trajectory. Generalized CTMs (GCTMs) (Kim et al., 2024) further extend this to connect arbitrary distributions via Flow Matching, enabling image translation and restoration. Both include finite-step consistency objectives during training. Recently, Align Your Flow (AYF) (Sabour et al., 2025) formulates continuous-time objectives for learning such finite-time flow maps. MeanFlow (Geng et al., 2026) and AlphaFlow (Zhang et al., 2026) build upon a complementary approach by representing flow maps through an *average velocity* field, allowing the finite-time displacement to be obtained without numerically integrating the instantaneous velocity field during sampling. Boffi et al. (2026) provide a general framework that unifies several flow-map and consistency-training objectives.

These approaches establish several mechanisms for learning finite-time maps between arbitrary points on an ODE trajectory. Our focus is complementary: we study the arbitrary-destination flow of a diffusion bridge and exploit its finite-time analytic structure, which leads to the bridge-specific parameterization and stability issue developed below.

3 Learning Diffusion-Bridge Flow Maps

Recall from Section 2.2 that the Bridge PF-ODE (Equation (1)) induces a conditional flow map $\Phi_{t\to s}(\boldsymbol{x}_t; \boldsymbol{x}_T)$ for times $0 \le s \le t < T$. Our aim is to learn a single model f_θ satisfying

$$f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) \approx \Phi_{t\to s}(\boldsymbol{x}_t; \boldsymbol{x}_T), \qquad 0 \le s \le t < T. \tag{2}$$

Since the destination time s can vary (i.e., $s \in [0, t]$), a model f_θ represents the entire family of bridge maps, with $s = 0$ corresponding to the endpoint mapping learned by consistency-based bridge models such as CDBM (He et al., 2024). However, learning the full bridge flow raises two questions: (i) how do we parameterize it for diffusion bridges, and (ii) how do we train it stably with continuous-time consistency?

We address these questions in turn: we first construct a flow-map parameterization from the finite-time bridge structure, and then derive a stable continuous-time rule for training it. Figure 2 previews this idea before we derive it formally.

3.1 Bridge Flow Structure

We first address how to parameterize f_θ to represent the full bridge flow. Following DBM-Solver (Venugopal et al., 2026), the exact finite-time solution of the Bridge PF-ODE (Equation (1)) can be expressed in terms of the weighted interval average of the denoising function, which we

denote by $\bar{\boldsymbol{D}}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)$:

$$\Phi_{t\to s}(\boldsymbol{x}_t; \boldsymbol{x}_T) = m(t,s)\boldsymbol{x}_t + n(t,s)\boldsymbol{x}_T + o(t,s)\bar{\boldsymbol{D}}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T), \tag{3}$$

where $m(t,s)$, $n(t,s)$, and $o(t,s)$ are known analytic coefficients multiplying $\boldsymbol{x}_t$, $\boldsymbol{x}_T$, and $\bar{\boldsymbol{D}}$, respectively. This decomposition forms the basis of our bridge flow map f_θ.

DBMSolver (Venugopal et al., 2026) approximates $\bar{\boldsymbol{D}}$ through a Taylor expansion of the denoising function, yielding a numerical solver for the Bridge PF-ODE. In contrast, we seek to represent this finite-interval quantity $\bar{\boldsymbol{D}}$ as a function of $(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)$ without numerically integrating the Bridge PF-ODE between t and s.

The decomposition in Eq. 3 naturally recovers two important boundary cases. As $s \to t$, the coefficients satisfy $(m, n, o) \to (1, 0, 0)$, recovering the identity map $\Phi_{t\to t}(\boldsymbol{x}_t; \boldsymbol{x}_T) = \boldsymbol{x}_t$. At $s = 0$, they reduce to $(m, n, o) = (0, 0, 1)$, giving

$$\Phi_{t\to 0}(\boldsymbol{x}_t; \boldsymbol{x}_T) = \bar{\boldsymbol{D}}(\boldsymbol{x}_t, t, 0, \boldsymbol{x}_T). \tag{4}$$

Thus, setting $s = 0$ recovers the endpoint mapping learned by consistency models.

3.2 Bridge-Structured Parameterization

Following the decomposition in Eq. 3, we parameterize the interval-denoising quantity $\bar{\boldsymbol{D}}$ as $\bar{\boldsymbol{D}}_\theta$:

$$\begin{aligned}\bar{\boldsymbol{D}}_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) &\coloneqq c_{\text{skip}}(t)\,\boldsymbol{x}_t + c_{\text{out}}(t)\,F_\theta(c_{\text{in}}(t)\boldsymbol{x}_t, c_{\text{noise}}(t), c_{\text{noise}}(s), c_{\text{in},T}\,\boldsymbol{x}_T) \\ &\approx \bar{\boldsymbol{D}}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T),\end{aligned} \tag{5}$$

where $c_{\text{in},T} \coloneqq 1/\text{Std}[\boldsymbol{x}_T]$ normalizes the conditioning endpoint input $\boldsymbol{x}_T$ to unit variance. Since $\bar{\boldsymbol{D}}$ depends on the destination time s, we condition F_θ on both source time t and destination time s, allowing a single model to represent interval-denoising quantities for arbitrary (t, s) pairs. For brevity, we absorb the input and time scalings into $F_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)$ hereafter.

For c_{in}, c_{skip}, c_{out}, and c_{noise}, we adopt the generalized bridge preconditioning of DDBM (Zhou et al., 2023), which extends the EDM preconditioning (Karras et al., 2022) to account for the two bridge endpoints. Their expressions under our notation are provided in the Appendix.

Substituting $\bar{\boldsymbol{D}}_\theta$ from Eq. 5 into the decomposition in Eq. 3 yields our bridge flow map f_θ:

$$f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) = \underbrace{[m(t,s) + o(t,s)c_{\text{skip}}(t)]\,\boldsymbol{x}_t + n(t,s)\,\boldsymbol{x}_T}_{H(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)} + \underbrace{o(t,s)c_{\text{out}}(t)}_{G(t,s)} F_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T), \tag{6}$$

Here, H denotes the analytic component of the parameterization, while G is the scalar gain applied to the network output. This bridge flow map decomposition will be useful in deriving and analyzing the continuous-time consistency objective in the following subsections.

3.3 Continuous-Time Consistency

We next address how to train the bridge flow map. For a destination time s and fixed endpoint $\boldsymbol{x}_T$, every source $\boldsymbol{x}_t$ along the same Bridge PF-ODE trajectory must map to the same destination $\boldsymbol{x}_s$:

$$\Phi_{t\to s}(\boldsymbol{x}_t; \boldsymbol{x}_T) = \boldsymbol{x}_s \qquad \Longrightarrow \qquad \underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\Phi_{t\to s}(\boldsymbol{x}_t; \boldsymbol{x}_T) = 0}_{\text{consistency condition}}. \tag{7}$$

This fixed-destination property provides a continuous-time consistency condition for learning the bridge flow map without explicitly constructing $\bar{\boldsymbol{D}}$ as a training target.

Following the continuous-time consistency formulation in Section 2.3, we specialize the continuous-time training rule to a fixed destination s and our bridge-structured flow map. For a fixed destination time s, as the distance between adjacent source times approaches zero, the gradient of the discrete consistency objective with respect to θ converges to

$$\nabla_\theta \mathbb{E}\left[w(t,s) f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)^\top \frac{\mathrm{d}}{\mathrm{d}t} f_{\theta^-}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)\right], \tag{8}$$

where $w(t,s)$ is a weighting function, and θ^- denotes a stop-gradient copy of θ. Here, $\mathrm{d}f_{\theta^-}/\mathrm{d}t$ is the total derivative along the Bridge PF-ODE trajectory. We refer to it as the *bridge flow tangent*:

$$\frac{\mathrm{d}}{\mathrm{d}t} f_{\theta^-}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) = \nabla_{\boldsymbol{x}_t} f_{\theta^-} \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \frac{\partial f_{\theta^-}}{\partial t}, \tag{9}$$

which we evaluate using a Jacobian–vector product (JVP). The JVP differentiates through all t-dependent inputs; the full expansion is provided in the Appendix.

3.4 Stabilizing Continuous-Time Consistency

While the continuous-time training rule in Equation (8) applies to arbitrary destination times s, we find that using this rule to train the bridge-structured flow map f_θ can be unstable. Recall that the network output F_θ is multiplied by the time-dependent gain $G(t,s) = o(t,s)c_{\mathrm{out}}(t)$ in the bridge-structured flow map f_θ. Consequently, computing the bridge flow tangent $\mathrm{d}f_{\theta^-}/\mathrm{d}t$ includes the derivative $G'(t,s) = \partial G(t,s)/\partial t$.

A local analysis shows that $G(t,s)$ can be non-monotonic with respect to the source time t, causing $G'(t,s)$ to cross zero and change sign. The sign change is important because it reverses the sign of the $G'(t,s)F_\theta$ term in the bridge flow tangent. However, the parameter update to θ induced by the continuous-time training rule does not compensate for this change in sign. Consequently, the resulting update to θ can move the bridge flow tangent toward zero when $G'(t,s) > 0$, but away from zero when $G'(t,s) < 0$, leading to severe degradation. We provide the full derivation and analysis of the sign-change region in the Appendix.

To address this issue, we multiply the bridge flow tangent by $\mathrm{sign}(G'(t,s))$ before forming the parameter update to θ. This leaves the update unchanged when $G'(t,s) > 0$, and reverses its direction when $G'(t,s) < 0$, compensating for the sign change. We additionally apply tangent normalization (Lu & Song, 2024) to control the scale of the resulting update.

With tangent normalization and our proposed sign correction, the final pseudo-objective is

$$\mathcal{L}_{\mathrm{BFM}} = \mathbb{E}\left\| F_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) - F_{\theta^-}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) + \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_{\mathrm{RMS}} + c} \right\|^2, \text{ with } c = 0.1, \tag{10}$$

$$\boldsymbol{g} = \underbrace{\mathrm{sign}\left(G'(t,s)\right)}_{\text{sign correction}} \cdot \underbrace{w(t,s)\, G(t,s)}_{\text{effective weight}} \cdot \frac{\mathrm{d}}{\mathrm{d}t} f_{\theta^-}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T),$$

where we set the effective weight $w(t,s)\, G(t,s) = c_{\mathrm{skip}}(t)$. The vector $\boldsymbol{g}$ is a stop-gradient quantity. During training, we sample both t and s ($s < t$), allowing a single f_θ to learn the entire bridge flow.

3.5 Training and Sampling BFMs

Training. We initialize our bridge flow map models from pretrained teacher DBMs and modify the network to accept and condition on the destination time s. At each iteration, we sample a paired observation $(\boldsymbol{x}_0, \boldsymbol{x}_T)$ along with the source and destination times $s < t$. After constructing the intermediate state $\boldsymbol{x}_t$, we use the paired endpoints to compute an analytic unbiased estimator of its marginal Bridge PF-ODE velocity $\mathrm{d}\boldsymbol{x}_t/\mathrm{d}t$ and finetune f_θ using our continuous-time pseudo-objective in Equation (10). We evaluate the bridge flow tangent using a JVP, as described in Section 3.3. The complete training procedure is summarized in Algorithm 1, with the full derivation and implementation details provided in the Appendix.

Sampling. We use a time schedule $0 = t_0 < t_1 < \cdots < t_K = T$ and sample backward from the conditioning endpoint $\boldsymbol{x}_T$ at t_K to $t_0 = 0$. Since the deterministic Bridge PF-ODE is singular at $t = T$ (Venugopal et al., 2026; Zheng et al., 2024), we do not integrate it directly from this boundary. Instead, the first model evaluation predicts $\hat{\boldsymbol{x}}_0$ from the conditioning endpoint $\boldsymbol{x}_T$, which we then re-noise through the forward bridge to sample the first interior state $\boldsymbol{x}_{t_{K-1}}$ at $t_{K-1} < T$. The subsequent sequence of flow-map transitions starts from $\boldsymbol{x}_{t_{K-1}}$. For each subsequent step from t_k to t_{k-1}, we employ γ-sampling (Kim et al., 2023), where γ controls the amount of stochasticity between successive flow-map evaluations. In particular, for nonterminal transitions, $\gamma = 0$ corresponds to deterministic flow-map sampling, while $\gamma = 1$ corresponds to $\boldsymbol{x}_0$-prediction followed by full re-noising. The final transition to $t_0 = 0$ is always deterministic. The complete sampling procedure is provided in Algorithm 2.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and Tasks. We evaluate our models on several I2I translation tasks across different resolutions, spanning both latent- and pixel-space settings. Our main experiments consider $4\times$ super-resolution and Gaussian deblurring on ImageNet (512×512) (Deng et al., 2009; Karras et al., 2024). We also evaluate normal-to-image translation on DIODE (256×256) (Vasiljevic et al., 2019) and edges-to-handbags translation on Edges2Handbags (64×64) (Isola et al., 2017).

Teacher DBMs. Since pretrained DBMs were not available for ImageNet (512×512), we finetune the publicly available pretrained EDM2 checkpoints into DBMs via denoising bridge score matching (Zhou et al., 2023). We modify the models' input layer to additionally accept the endpoint conditioning $\boldsymbol{x}_T$. We use the XS variant for Super-Resolution and the S variant for Deblurring. For each ImageNet task, all reproduced methods use the same task-specific EDM2 (Karras et al., 2024)-based teacher DBM. For E2H and DIODE, we instead train DBMs from scratch using the EDM2 UNet, which we similarly modify to accept $\boldsymbol{x}_T$ as input. As several published E2H and DIODE baselines use ADM-based UNets while our models use EDM2-based UNets, we additionally report sampling results from the matched EDM2-based DBM teachers in Section I.1 to contextualize the backbone difference and measure BFM's acceleration relative to its own teacher.

Evaluation. We evaluate the models using the sampling procedure described in Section 3.5 and report performance at different numbers of function evaluations (NFEs). For ImageNet restoration, we report FID-VAE and top-1 ImageNet classifier accuracy on the fixed 10k-image subset of the ImageNet validation set used by I^2SB (Liu et al., 2023). FID-VAE compares generated images against VAE-reconstructed clean targets using standard Inception-v3 preprocessing. For E2H and DIODE, we follow the legacy training-split evaluation protocol used by prior diffusion-bridge and bridge-distillation methods (Zhou et al., 2023; He et al., 2024; Gushchin et al., 2025), and report FID, Inception Score, LPIPS, and MSE. These results enable direct comparison with published work and are not intended as held-out evaluation; the ImageNet restoration experiments provide our primary held-out quantitative results. Complete evaluation details and a teacher-relative analysis on the small E2H and DIODE test splits are provided in Sections G.5 and I.2.

4.2 FEW-STEP IMAGE RESTORATION

We evaluate few-step image restoration on ImageNet-512 for SR$4\times$ and Gaussian deblurring. Table 1 reports FID-VAE and classifier accuracy. On super-resolution, BFM improves over our reproduced CDBM baseline at every reported NFE budget. On deblurring, CDBM retains better FID-VAE at the extreme 2-NFE budget, whereas BFM overtakes it by 4 NFEs. We hypothesize this is because learning a flow map is more difficult compared to a consistency model (Sabour et al., 2025). At 8 NFEs on both tasks, BFM reaches FID-VAE comparable to DBMSolver using 20–30 NFEs. At 2 NFEs, BFM also improves over a matched IBMD reproduction on both tasks, reducing FID-VAE from 5.26 to 2.72 on SR$4\times$ and from 4.73 to 4.06 on deblurring.

4.3 APPLICABILITY TO OTHER I2I TASKS

To evaluate whether the benefits of our bridge flow-map model extend beyond high-resolution image restoration, we evaluate edges-to-handbag translation on Edges2Handbags (64×64) and surface-normal-to-RGB translation on DIODE (256×256). We compare BFM against prior diffusion-bridge, consistency-based, and inverse bridge matching methods, and report the results in Table 2.

Table 2 extends the comparison to E2H and DIODE under the legacy training-split protocol. At 2 NFEs, BFM matches 30-NFE DBMSolver on E2H and outperforms it on DIODE, while improving further with additional evaluations. It also outperforms the reported two-step endpoint-mapping baselines on both tasks. Representative 2-NFE generated samples on both tasks are provided in Figure 5, showing the conditioning endpoint $\boldsymbol{x}_T$, source data $\boldsymbol{x}_0$, and BFM's generated sample.

Table 1: ImageNet (512×512) restoration results. We report FID-VAE (↓), top-1 ImageNet classifier accuracy (CA; ↑) and the NFEs (↓) for $4\times$ super-resolution $(128 \rightarrow 512$; left) and Gaussian deblurring (right). The endpoint-directed control concentrates training on $s = 0$, whereas full-flow training supervises arbitrary $s < t$; both use the same architecture and training budget. Within each method family, the best value of each metric is bolded.

4×SR Method	NFE ↓	FID-VAE ↓	CA (%) ↑
DBMs & Fast Samplers			
Hybrid Heun (Zhou et al., 2023)	119	1.97	73.82
DBIM-3 (Zheng et al., 2024)	10	2.53	72.91
	20	1.96	74.52
	30	1.83	74.63
DBMSolver	10	**1.71**	**75.13**
(Venugopal et al., 2026)	20	1.74	74.80
	30	1.78	74.70
Endpoint-Mapping Models			
CDBM (He et al., 2024)(reprod)	2	3.00	73.96
	4	**2.96**	**73.98**
	8	3.01	73.90
IBMD (Gushchin et al., 2025) (reprod)	2	5.26	73.76
Flow-Map Models			
BFM (endpoint-directed)	2	3.34	74.55
	4	2.10	**75.19**
	8	1.88	74.55
BFM (full-flow, ours)	2	2.72	75.09
	4	**1.79**	74.88
	8	1.82	75.09

Deblurring Method	NFE ↓	FID-VAE ↓	CA (%) ↑
DBMs & Fast Samplers			
Hybrid Heun (Zhou et al., 2023)	119	2.15	74.07
DBIM-3 (Zheng et al., 2024)	10	2.87	72.59
	20	2.14	74.32
	30	2.02	74.44
DBMSolver	10	2.00	**75.04**
(Venugopal et al., 2026)	20	**1.95**	74.52
	30	1.98	74.55
Endpoint-Mapping Models			
CDBM (He et al., 2024) (reprod)	2	2.90	74.49
	4	**2.85**	**74.27**
	8	2.87	73.95
IBMD (Gushchin et al., 2025) (reprod)	2	4.73	74.15
Flow-Map Models			
BFM (endpoint-directed)	2	5.84	72.30
	4	2.97	74.92
	8	2.16	75.04
BFM (full-flow, ours)	2	4.06	74.48
	4	2.19	75.04
	8	**1.99**	**75.16**

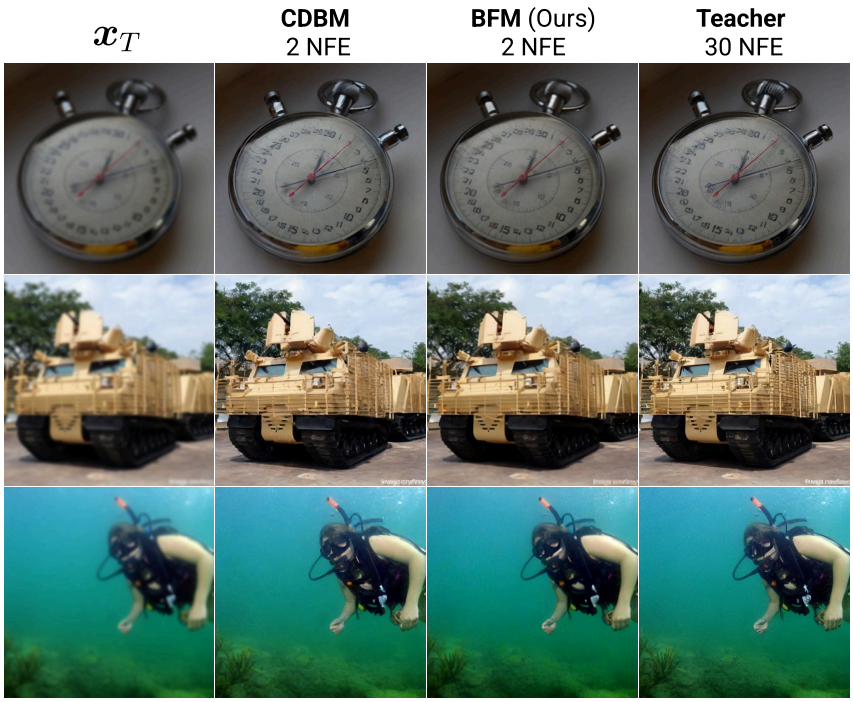

Figure 3: Qualitative comparison on ImageNet-512 Gaussian deblurring at matched 2 NFEs. From left to right: the blurred conditioning input x_T, CDBM (He et al., 2024) (2 NFE), BFM (2 NFE, ours), and a DBMSolver (Venugopal et al., 2026) reference sampled with 30 NFEs. At the same 2-NFE budget, BFM recovers finer details than CDBM: the stopwatch's dial numerals (e.g., "10" and "30") and both hands stay sharp and legible, where CDBM blurs them into indistinct marks (top); the straight guard-rail bars around the military vehicle remain distinct rather than merging into a blurred band (middle); and the diver's individual fingers are resolved rather than fused together (bottom).

Table 2: Quantitative results on the Edges2Handbags (E2H) and DIODE training splits, following the legacy evaluation protocol used by prior diffusion-bridge work. We report FID (↓), IS (↑), LPIPS (↓), and MSE (↓) against the NFE (↓). IBMD does not report LPIPS or MSE.

Method	NFE ↓	E2H (64×64)				DIODE (256×256)			
		FID ↓	IS ↑	LPIPS ↓	MSE ↓	FID ↓	IS ↑	LPIPS ↓	MSE ↓
DDIB (Su et al., 2022)	≥ 40	186.84	2.04	0.869	1.050	242.3	4.22	0.798	0.794
SDEdit (Meng et al., 2021)	≥ 40	26.5	3.58	0.271	0.510	31.14	5.70	0.714	0.534
I^2SB (Liu et al., 2023)	≥ 40	7.43	3.40	0.244	0.191	9.34	5.77	0.373	0.145
Hybrid Heun (Zhou et al., 2023)	119	1.83	3.73	0.142	0.040	4.43	**6.21**	0.244	0.084
DBIM-3 (Zheng et al., 2024)	30	1.21	3.60	0.099	0.005	3.50	6.07	0.200	0.018
DBMSolver (Venugopal et al., 2026)	30	0.52	3.63	0.100	0.005	2.08	6.03	0.199	0.018
CBD (He et al., 2024)	2	1.30	3.62	0.128	0.012	3.66	6.02	0.224	0.022
CBT (He et al., 2024)	2	0.80	3.65	0.106	0.007	**2.93**	**6.06**	**0.205**	**0.018**
IBMD (Gushchin et al., 2025)	2	**0.67**	**3.69**	–	–	3.12	5.92	–	–
BFM (Ours)	2	0.52	**3.68**	0.040	0.0005	1.69	6.11	0.119	0.004
	4	0.48	3.67	0.040	0.0004	1.60	6.12	**0.111**	0.004
	8	**0.47**	3.67	**0.040**	**0.0004**	**1.38**	**6.16**	0.112	**0.004**

4.4 Benefit of Full-Flow Supervision

We isolate the benefit of full-flow supervision using a matched ImageNet (512×512) Gaussian-deblurring ablation whose variants differ only in their destination-time distribution during training. At inference, both variants use the same sampling schedule, with every flow-map evaluation directed to $s = 0$ ($\gamma = 1$). Nevertheless, full-flow supervision improves endpoint-generation quality at every reported NFE. It also improves intermediate-destination accuracy and reduces discrepancy between direct and composed flow-map transitions. Complete results are provided in Sec. H.1 and H.2.

4.5 Analyzing the Bridge-Structured Design

Bridge-Structured Parameterization. Our experiments show that the bridge-structured parameterization is important in practice. A generic average-velocity parameterization produces severely degraded endpoint predictions and 2-NFE samples under the same architecture, training budget, and sampling schedule. This behavior is consistent with poor long-interval conditioning: near the sampling boundary, small velocity errors are strongly amplified by the source–destination interval. Section F provides the conditioning analysis, endpoint diagnostics, and controlled comparisons.

Stability Correction. In this controlled bridge setting, the sign correction is essential in practice. Removing the sign correction in a matched ImageNet (512×512) Gaussian-deblurring ablation degrades 2-NFE FID-VAE from 4.64 to 165.42 and classifier accuracy from 73.05% to 16.32%. Although the sign-uncorrected run retains finite gradient norms throughout training, its severe degradation is consistent with the local sign reversal analyzed in Section E. The complete controlled comparison is provided in Section E.1.

5 Conclusion

We introduced BFM, a continuous-time consistency model that learns the full conditional flow of a diffusion bridge between arbitrary source and destination times. Our bridge-structured parameterization retains the known finite-time contributions of the current state and conditioning endpoint, while the neural network represents the remaining interval-dependent denoising quantity. We further identified a local sign reversal in the continuous-time update and derived a correction for it.

Across ImageNet (512×512) restoration, E2H, and DIODE, BFM produces high-quality translations in 2–8 NFEs and approaches strong DBM solvers that require substantially more evaluations. Controlled experiments show the importance of the bridge-structured parameterization and sign correction, while full-flow supervision improves endpoint sampling, intermediate-destination accuracy, and approximate flow composition. Together, these results show that learning bridge-structured finite-time flow maps provides an effective approach to fast and flexible I2I translation.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, research planning, mathematical presentation, code and figure development, experimental diagnostics, and interpretation of results. The authors reviewed and verified all AI-assisted derivations, code, experiments, interpretations, and writing, and take full responsibility for the content of this work.

ETHICS STATEMENT

This work studies few-step image-to-image translation using existing research datasets; we collect no new data and conduct no human-subject experiments. Like other generative image models, diffusion-bridge models may reproduce biases present in their training data or generate plausible details that are absent from the conditioning input. Thus their outputs should not be treated as faithful recovery of missing information in high-stakes medical, forensic, or evidentiary applications.

REPRODUCIBILITY STATEMENT

The main paper specifies the bridge flow-map parameterization, continuous-time training objective, stability correction, and sampling procedure. The Appendix provides the complete derivations, training and sampling algorithms, architecture details, optimization configurations, checkpoint selections, sampling schedules, evaluation protocols, and controlled ablations. We report the dataset splits and evaluation-set sizes used for every experiment and distinguish the held-out ImageNet evaluation from the legacy E2H and DIODE training-split protocol. We intend to release the implementation, experiment configurations, model checkpoints, and evaluation scripts upon publication.

REFERENCES

Nicholas Boffi, Michael Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. *Advances in Neural Information Processing Systems*, 38: 33346–33382, 2026.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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*, 2009.

Joseph L Doob. *Classical potential theory and its probabilistic counterpart*. Springer, 1984.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026.

Nikita Gushchin, David Li, Daniil Selikhanovych, Evgeny Burnaev, Dmitry Baranchuk, and Alexander Korotin. Inverse bridge matching distillation. *arXiv preprint arXiv:2502.01362*, 2025.

Guande He, Kaiwen Zheng, Jianfei Chen, Fan Bao, and Jun Zhu. Consistency diffusion bridge models. *Advances in Neural Information Processing Systems*, 2024.

Jinhui Hou, Zhiyu Zhu, and Junhui Hou. Energy-oriented diffusion bridge for image restoration with foundational diffusion models. In *International Conference on Learning Representations*, 2026.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Beomsu Kim, Jaemin Kim, Jeongsol Kim, and Jong Chul Ye. Generalized consistency trajectory models for image manipulation. *arXiv preprint arXiv:2403.12510*, 2024.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2sb: Image-to-image schr\" odinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Alex Morehead. JVP Flash Attention, September 2025. URL `https://github.com/amorehead/jvp_flash_attention`.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. *arXiv preprint arXiv:2506.14603*, 2025.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *arXiv preprint arXiv:2203.08382*, 2022.

Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, 2019.

Sankarshana Venugopal, Mohammad Mostafavi, and Jonghyun Choi. Dbmsolver: A training-free diffusion bridge sampler for high-quality image-to-image translation. In *arXiv preprint*, 2026.

Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. Alphaflow: Understanding and improving meanflow models. In *International Conference on Learning Representations*, volume 2026, pp. 144056–144094, 2026.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. *arXiv preprint arXiv:2405.15885*, 2024.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.

A Training and Sampling Algorithms

We provide pseudocode for the training and sampling procedures described in Section 3.5.

Algorithm 1 BFM Training

Require: Paired data $(\boldsymbol{x}_0, \boldsymbol{x}_T)$, pretrained DBM D_{pre} with params θ_{pretrain}, flow-map model f_θ

Initialize f_θ from D_{pre}
repeat
  Sample $(\boldsymbol{x}_0, \boldsymbol{x}_T)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and times $s < t$
  Construct the bridge state $\boldsymbol{x}_t = a_t \boldsymbol{x}_T + b_t \boldsymbol{x}_0 + c_t \boldsymbol{\epsilon}$
  Estimate Bridge PF-ODE velocity $\mathrm{d}\boldsymbol{x}_t/\mathrm{d}t$
  Evaluate $F_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)$
  Compute $\mathrm{d}f_{\theta^-}/\mathrm{d}t$ ▷ JVP; see Section D
  $\boldsymbol{g} \leftarrow \mathrm{sign}(G'(t,s))\, c_{\text{skip}}(t)\, \mathrm{d}f_{\theta^-}/\mathrm{d}t$ ▷ Sign correction (Section 3.4)
  $\boldsymbol{g} \leftarrow \boldsymbol{g}/(\|\boldsymbol{g}\|_{\text{RMS}} + c)$
  $\mathcal{L}_{\text{BFM}} \leftarrow \|F_\theta - F_{\theta^-} + \boldsymbol{g}\|_2^2$ ▷ Pseudo-objective (Equation (10))
  Update θ using $\nabla_\theta \mathcal{L}_{\text{BFM}}$
until convergence

Algorithm 2 BFM Sampling

Require: Conditioning $\boldsymbol{x}_T$, time schedule $0 = t_0 < t_1 < \cdots < t_K = T$, flow map f_θ, $\gamma \in [0,1]$

$\boldsymbol{x}_{t_K} \leftarrow \boldsymbol{x}_T$, and sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\hat{\boldsymbol{x}}_0 \leftarrow f_\theta(\boldsymbol{x}_{t_K}, t_K, t_0, \boldsymbol{x}_T)$
$\boldsymbol{x}_{t_{K-1}} \leftarrow a_{t_{K-1}} \boldsymbol{x}_T + b_{t_{K-1}} \hat{\boldsymbol{x}}_0 + c_{t_{K-1}} \boldsymbol{\epsilon}$ ▷ Stochastic initialization
for $k = K-1, \ldots, 1$ **do**
  $u_k \leftarrow \sqrt{1-\gamma^2}\, t_{k-1}$ ▷ γ-sampling
  $\boldsymbol{x}_{u_k} \leftarrow f_\theta(\boldsymbol{x}_{t_k}, t_k, u_k, \boldsymbol{x}_T)$
  if $\gamma = 0$ or $t_{k-1} = 0$ **then**
    $\boldsymbol{x}_{t_{k-1}} \leftarrow \boldsymbol{x}_{u_k}$ ▷ Deterministic flow-map sampling
  else
    Sample $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    $r_k \leftarrow b_{t_{k-1}}/b_{u_k}$
    $\sigma_k \leftarrow \sqrt{c_{t_{k-1}}^2 - r_k^2 c_{u_k}^2}$
    $\boldsymbol{x}_{t_{k-1}} \leftarrow r_k \boldsymbol{x}_{u_k} + (a_{t_{k-1}} - r_k a_{u_k}) \boldsymbol{x}_T + \sigma_k \boldsymbol{z}$ ▷ Exact forward bridge transition
  end if
end for
return $\boldsymbol{x}_{t_0}$

B Exact Finite-Time Bridge Flow

We first derive the explicit finite-time solution of the Bridge PF-ODE. Let t denote the source time and $s < t$ the destination time. Let

$$\lambda_t \coloneqq \log \frac{\alpha_t}{\sigma_t}, \qquad \mathcal{R}_t \coloneqq \exp(2(\lambda_t - \lambda_T)) - 1, \tag{11}$$

where λ_t is half of the log signal-to-noise ratio.

Following DBMSolver (Venugopal et al., 2026), given $0 \le s < t < T$ the exact solution of the Bridge PF-ODE from $\boldsymbol{x}_t$ to $\boldsymbol{x}_s$ can be written as

$$\begin{aligned}\Phi_{t\to s}(\boldsymbol{x}_t;\boldsymbol{x}_T) = \frac{\alpha_s}{\alpha_t}e^{2(\lambda_t-\lambda_s)}\sqrt{\frac{\mathcal{R}_s}{\mathcal{R}_t}}\,\boldsymbol{x}_t \\ + \frac{\alpha_s}{\alpha_T}e^{2(\lambda_T-\lambda_s)}\left(1-\sqrt{\frac{\mathcal{R}_s}{\mathcal{R}_t}}\right)\boldsymbol{x}_T \\ + \alpha_s e^{-2\lambda_s}\sqrt{\mathcal{R}_s}\int_{\lambda_t}^{\lambda_s}\frac{e^{2\lambda}\boldsymbol{D}(\boldsymbol{x}_\lambda,\lambda,\boldsymbol{x}_T)}{\sqrt{\exp(2(\lambda-\lambda_T))-1}}\mathrm{d}\lambda,\end{aligned} \tag{12}$$

where $\boldsymbol{x}_\lambda$ is the intermediate state on the same Bridge PF-ODE trajectory at λ, and $\boldsymbol{D}$ denotes the denoising function learned by a $\boldsymbol{x}_0$-predicting Diffusion Bridge Model.

Next, let's define the positive normalizing factor

$$\begin{aligned}Z(t,s) &\coloneqq \int_{\lambda_t}^{\lambda_s}\frac{e^{2\lambda}}{\sqrt{\exp(2(\lambda-\lambda_T))-1}}\mathrm{d}\lambda \\ &= e^{2\lambda_T}\left(\sqrt{\mathcal{R}_s}-\sqrt{\mathcal{R}_t}\right).\end{aligned} \tag{13}$$

For the schedules considered here, $\lambda_s > \lambda_t$ whenever $s < t$, and hence $Z(t,s) > 0$.

We can then define the *normalized weighted interval average*

$$\bar{\boldsymbol{D}}(\boldsymbol{x}_t,t,s,\boldsymbol{x}_T) \coloneqq \frac{1}{Z(t,s)}\int_{\lambda_t}^{\lambda_s}\frac{e^{2\lambda}\boldsymbol{D}(\boldsymbol{x}_\lambda,\lambda,\boldsymbol{x}_T)}{\sqrt{\exp(2(\lambda-\lambda_T))-1}}\mathrm{d}\lambda. \tag{14}$$

Thus, $\bar{\boldsymbol{D}}$ is not an instantaneous denoising prediction: it is a *trajectory-dependent* weighted average over the interval from t to s.

Next, substituting Equation (14) into Equation (12), we get

$$\Phi_{t\to s}(\boldsymbol{x}_t;\boldsymbol{x}_T) = m(t,s)\boldsymbol{x}_t + n(t,s)\boldsymbol{x}_T + o(t,s)\bar{\boldsymbol{D}}(\boldsymbol{x}_t,t,s,\boldsymbol{x}_T), \tag{15}$$

with the finite-time coefficients

$$m(t,s) = \frac{\alpha_s}{\alpha_t}e^{2(\lambda_t-\lambda_s)}\sqrt{\frac{\mathcal{R}_s}{\mathcal{R}_t}}, \tag{16}$$

$$n(t,s) = \frac{\alpha_s}{\alpha_T}e^{2(\lambda_T-\lambda_s)}\left(1-\sqrt{\frac{\mathcal{R}_s}{\mathcal{R}_t}}\right), \tag{17}$$

$$o(t,s) = \alpha_s\left(1-e^{2(\lambda_T-\lambda_s)}\right)\left(1-\sqrt{\frac{\mathcal{R}_t}{\mathcal{R}_s}}\right). \tag{18}$$

B.1 Boundary Cases

We show that the structured solution in Equation (15) satisfies the two boundary conditions required of a bridge flow map. First, as $s \to t$,

$$(m(t,s), n(t,s), o(t,s)) \longrightarrow (1,0,0),$$

and therefore $\Phi_{t\to t}(\boldsymbol{x}_t;\boldsymbol{x}_T) = \boldsymbol{x}_t$. Note that although $Z(t,s)$ approaches zero in this limit, its normalized average remains bounded and is multiplied by $o(t,s) \to 0$.

Second, under the standard diffusion schedules $\alpha_0 = 1$ and $\sigma_0 = 0$, we have $\lambda_s \to +\infty$ as $s \to 0$. Consequently,

$$(m(t,0), n(t,0), o(t,0)) = (0,0,1),$$

which gives

$$\Phi_{t\to 0}(\boldsymbol{x}_t;\boldsymbol{x}_T) = \bar{\boldsymbol{D}}(\boldsymbol{x}_t,t,0,\boldsymbol{x}_T).$$

Endpoint consistency is therefore recovered as a boundary case of the full bridge flow.

B.2 Behavior at the Conditioning Endpoint

As noted in prior work (Zheng et al., 2024; Venugopal et al., 2026), the deterministic Bridge PF-ODE is singular at $t = T$, since $\mathcal{R}_t \to 0$ as $t \to T$.

Although the PF-ODE derivative is singular at T, the endpoint prediction

$$\hat{\boldsymbol{x}}_0 = f_\theta(\boldsymbol{x}_T, T, 0, \boldsymbol{x}_T)$$

used for stochastic initialization remains well-defined. Following established DBM samplers, we renoise this prediction through the forward bridge to sample the first interior state $\boldsymbol{x}_{t_{K-1}}$ at $t_{K-1} < T$. Thus, neither deterministic integration nor a training tangent is evaluated at T, and all subsequent flow-map evaluations have source times strictly below T.

B.3 Derivatives of the Finite-Time Coefficients

During continuous-time training, the destination time s is held fixed and the coefficients are differentiated with respect to the source time t. Let

$$\lambda'_t = \frac{\alpha'_t}{\alpha_t} - \frac{\sigma'_t}{\sigma_t}, \qquad \mathcal{R}'_t = 2(\mathcal{R}_t + 1)\lambda'_t. \tag{19}$$

For $s > 0$ and $t < T$, differentiating the coefficients in Equations (16) to (18) with respect to t gives

$$\partial_t m(t,s) = m(t,s)\left[-\frac{\alpha'_t}{\alpha_t} + \left(1 - \frac{1}{\mathcal{R}_t}\right)\lambda'_t\right], \tag{20}$$

$$\partial_t n(t,s) = \frac{\alpha_s}{\alpha_T} e^{2(\lambda_T - \lambda_s)} \sqrt{\frac{\mathcal{R}_s}{\mathcal{R}_t}}\, \frac{\mathcal{R}_t + 1}{\mathcal{R}_t}\, \lambda'_t, \tag{21}$$

$$\partial_t o(t,s) = -\alpha_s \left(1 - e^{2(\lambda_T - \lambda_s)}\right) \sqrt{\frac{\mathcal{R}_t}{\mathcal{R}_s}}\, \frac{\mathcal{R}_t + 1}{\mathcal{R}_t}\, \lambda'_t. \tag{22}$$

At the endpoint case of $s = 0$, the coefficients are $(m, n, o) = (0, 0, 1)$ regardless of t, and hence their derivatives with respect to t are zero.

C Bridge Preconditioning and Its Derivatives

We use the generalized bridge preconditioning introduced by DDBM (Zhou et al., 2023). We reproduce its scaling functions here only to establish notation and provide the derivatives required by our continuous-time objective.

Recall that an intermediate bridge state can be expressed as

$$\boldsymbol{x}_t = a_t \boldsymbol{x}_T + b_t \boldsymbol{x}_0 + c_t \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

Following DDBM, let

$$\sigma_d^2 \coloneqq \mathrm{Var}[\boldsymbol{x}_0], \qquad \sigma_e^2 \coloneqq \mathrm{Var}[\boldsymbol{x}_T], \qquad \sigma_{de} \coloneqq \mathrm{Cov}[\boldsymbol{x}_0, \boldsymbol{x}_T],$$

where these quantities denote scalar dataset-level moments. We define

$$V_t \coloneqq a_t^2 \sigma_e^2 + b_t^2 \sigma_d^2 + 2 a_t b_t \sigma_{de} + c_t^2, \tag{23}$$

$$U_t \coloneqq b_t \sigma_d^2 + a_t \sigma_{de}, \tag{24}$$

$$R_t \coloneqq a_t^2 \left(\sigma_e^2 \sigma_d^2 - \sigma_{de}^2\right) + \sigma_d^2 c_t^2. \tag{25}$$

Under our notation, the DDBM preconditioning functions are

$$c_{\mathrm{in}}(t) = \sqrt{\frac{1}{V_t}}, \qquad c_{\mathrm{skip}}(t) = \frac{U_t}{V_t}, \qquad c_{\mathrm{out}}(t) = \sqrt{\frac{R_t}{V_t}}. \tag{26}$$

For $u \in \{t, s\}$, we use the time-conditioning transformation

$$c_{\mathrm{noise}}(u) = \begin{cases} \frac{1}{4}\log(\max(u, \epsilon_{\mathrm{time}}))\,, & \text{for ImageNet } (512 \times 512) \text{ restoration,} \\ u, & \text{for E2H and DIODE,} \end{cases} \qquad \epsilon_{\mathrm{time}} = 10^{-8}. \tag{27}$$

The logarithmic transformation matches the EDM2 time conditioning (Karras et al., 2024), while the identity transformation follows sCM (Lu & Song, 2024). The same transformation is applied to the source and destination times. Clamping the logarithmic input makes the destination embedding well-defined when $s = 0$. Additionally, we normalize the conditioning endpoint $\boldsymbol{x}_T$ using

$$c_{\mathrm{in},T} = \sigma_e^{-1}. \tag{28}$$

We parameterize the learned approximation to the interval average as

$$\bar{\boldsymbol{D}}_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) = c_{\mathrm{skip}}(t)\boldsymbol{x}_t + c_{\mathrm{out}}(t)F_\theta(c_{\mathrm{in}}(t)\boldsymbol{x}_t, c_{\mathrm{noise}}(t), c_{\mathrm{noise}}(s), c_{\mathrm{in},T}\boldsymbol{x}_T)\,. \tag{29}$$

C.1 Preconditioning Derivatives

For completeness, we provide the source-time derivatives used in the bridge flow tangent and JVP computations. Writing $a_t' = \partial_t a_t$, $b_t' = \partial_t b_t$, and $c_t' = \partial_t c_t$, we have

$$V_t' = 2a_t a_t' \sigma_e^2 + 2b_t b_t' \sigma_d^2 + 2(a_t' b_t + a_t b_t')\sigma_{de} + 2c_t c_t', \tag{30}$$

$$U_t' = b_t' \sigma_d^2 + a_t' \sigma_{de}, \tag{31}$$

$$R_t' = 2a_t a_t' \left(\sigma_e^2 \sigma_d^2 - \sigma_{de}^2\right) + 2\sigma_d^2 c_t c_t'. \tag{32}$$

The preconditioning derivatives are therefore

$$\partial_t c_{\mathrm{in}}(t) = -\frac{V_t'}{2V_t^{3/2}}, \tag{33}$$

$$\partial_t c_{\mathrm{skip}}(t) = \frac{U_t'}{V_t} - \frac{U_t V_t'}{V_t^2}, \tag{34}$$

$$\partial_t c_{\mathrm{out}}(t) = \frac{R_t'}{2V_t c_{\mathrm{out}}(t)} - \frac{c_{\mathrm{out}}(t) V_t'}{2V_t}. \tag{35}$$

These expressions hold at interior source times; endpoint values are interpreted through their corresponding limits. Since every sampled source time satisfies $t > \epsilon_{\mathrm{time}}$, the derivative of the source-time embedding is

$$\partial_t c_{\mathrm{noise}}(t) = \begin{cases} \frac{1}{4t}, & \text{for ImageNet } (512 \times 512) \text{ restoration,} \\ 1, & \text{for E2H and DIODE.} \end{cases} \tag{36}$$

The bridge flow tangent differentiates the map with respect to its source time while holding the destination s fixed. Consequently,

$$\partial_t c_{\mathrm{noise}}(s) = 0. \tag{37}$$

The endpoint normalization $c_{\mathrm{in},T}$ is likewise independent of t. These derivatives enter the Jacobian–vector product derived in the following section.

D Continuous-Time Training Derivation for Diffusion-Bridge Flow Maps

D.1 Continuous-Time Limit of Source Consistency

We derive the continuous-time training rule for a fixed destination time s and conditioning endpoint $\boldsymbol{x}_T$. Let $\boldsymbol{x}_t$ evolve under the marginal Bridge PF-ODE

$$\dot{\boldsymbol{x}}_t \coloneqq \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t}. \tag{38}$$

Throughout this section, $\mathrm{d}/\mathrm{d}t$ denotes the total derivative along this marginal Bridge PF-ODE trajectory, while the destination s and conditioning endpoint $\boldsymbol{x}_T$ are held fixed.

For a sufficiently small $\Delta t > 0$, define $t' = t - \Delta t$, where $s \le t' < t$. The corresponding state on the same Bridge PF-ODE trajectory satisfies

$$\boldsymbol{x}_{t'} = \boldsymbol{x}_t - \Delta t \frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} + \mathcal{O}(\Delta t^2). \tag{39}$$

Consider the rescaled discrete consistency bridge objective

$$\mathcal{L}_{\Delta t} = \mathbb{E}\left[\frac{w(t,s)}{2\Delta t}\left\|f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) - f_{\theta^-}(\boldsymbol{x}_{t'}, t', s, \boldsymbol{x}_T)\right\|_2^2\right]. \tag{40}$$

Here, $w(t,s)$ is a scalar weighting function and $\theta^- = \mathrm{stopgrad}(\theta)$ denotes a copy of the current parameters through which gradients are not propagated. A first-order expansion of the stop-gradient target gives

$$f_{\theta^-}(\boldsymbol{x}_{t'}, t', s, \boldsymbol{x}_T) = f_{\theta^-}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) - \Delta t \frac{\mathrm{d}}{\mathrm{d}t} f_{\theta^-}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) + \mathcal{O}(\Delta t^2). \tag{41}$$

Because f_θ and f_{θ^-} have identical values at the same input, but gradients propagate only through f_θ, taking $\Delta t \to 0$ yields

$$\lim_{\Delta t \to 0} \nabla_\theta \mathcal{L}_{\Delta t} = \nabla_\theta \mathbb{E}\left[w(t,s) f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)^\top \frac{\mathrm{d}}{\mathrm{d}t} f_{\theta^-}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)\right]. \tag{42}$$

Paired-endpoint velocity estimator. The continuous-time derivation above uses the marginal Bridge PF-ODE velocity $\mathrm{d}\boldsymbol{x}_t/\mathrm{d}t$ from Equation (1). During finetuning, paired endpoints allow us to evaluate an analytic conditional bridge score instead of the unknown marginal score. Let $q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{x}_T)$ denote the conditional bridge marginal. The conditional and marginal scores satisfy

$$\mathbb{E}[\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{x}_T) \,|\, \boldsymbol{x}_t, \boldsymbol{x}_T] = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t \mid \boldsymbol{x}_T). \tag{43}$$

Consequently, replacing the marginal score in Equation (1) with the analytically known conditional bridge score gives an unbiased stochastic estimator of $\mathrm{d}\boldsymbol{x}_t/\mathrm{d}t$. Because a Jacobian–vector product is linear in its tangent argument, this also gives an unbiased stochastic estimator of the bridge flow tangent used by our finetuning objective.

D.2 Bridge Flow Tangent

To simplify notation, throughout the remainder of this section F_θ denotes the network evaluated at

$$(c_{\mathrm{in}}(t)\boldsymbol{x}_t, c_{\mathrm{noise}}(t), c_{\mathrm{noise}}(s), c_{\mathrm{in},T}\boldsymbol{x}_T)\,.$$

Recall that our bridge flow map can be written as

$$f_\theta = H + GF_\theta, \tag{44}$$

where

$$H(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) = A(t,s)\boldsymbol{x}_t + n(t,s)\boldsymbol{x}_T, \tag{45}$$
$$A(t,s) \coloneqq m(t,s) + o(t,s)c_{\mathrm{skip}}(t), \tag{46}$$
$$G(t,s) \coloneqq o(t,s)c_{\mathrm{out}}(t). \tag{47}$$

Recall that throughout the source-time derivative, s and $\boldsymbol{x}_T$ are held fixed. Therefore, the required coefficient derivatives are

$$\partial_t A(t,s) = \partial_t m(t,s) + \partial_t o(t,s)c_{\mathrm{skip}}(t) + o(t,s)\partial_t c_{\mathrm{skip}}(t), \tag{48}$$
$$\partial_t G(t,s) = \partial_t o(t,s)c_{\mathrm{out}}(t) + o(t,s)\partial_t c_{\mathrm{out}}(t). \tag{49}$$

Applying the product rule gives the total bridge flow tangent:

$$\begin{aligned}\frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t} &= \partial_t A(t,s)\boldsymbol{x}_t + A(t,s)\dot{\boldsymbol{x}}_t + \partial_t n(t,s)\boldsymbol{x}_T \\ &\quad + \partial_t G(t,s)F_{\theta^-} + G(t,s)\frac{\mathrm{d}F_{\theta^-}}{\mathrm{d}t}.\end{aligned} \tag{50}$$

All derivatives of the coefficients and preconditionings in this expression are given in the previous sections.

D.3 Continuous-Time Training Rule for Bridge Flow Maps

The continuous-time gradient in Equation (42) is written in terms of the complete bridge flow map f_θ. In practice, the learnable component of our parameterization is F_θ. Therefore,

$$f_\theta = H + GF_\theta \implies \nabla_\theta f_\theta = G(t,s)\nabla_\theta F_\theta, \tag{51}$$

as neither H nor G contains learnable parameters.

Because the sampling distribution, $w(t,s)$, and the stop-gradient tangent $\frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}$ are independent of θ, we obtain

$$\begin{aligned}
&\nabla_\theta \mathbb{E}\left[w(t,s) f_\theta^\top \frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}\right] \\
&= \mathbb{E}\left[w(t,s)\left(\nabla_\theta f_\theta\right)^\top \frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}\right] \\
&= \mathbb{E}\left[w(t,s)G(t,s)\left(\nabla_\theta F_\theta\right)^\top \frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}\right] \\
&= \nabla_\theta \mathbb{E}\left[w(t,s)G(t,s)F_\theta^\top \frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}\right].
\end{aligned} \tag{52}$$

Equation (52) shows that the bridge flow tangent is paired with $\nabla_\theta F_\theta$ through the scalar factor $w(t,s)G(t,s)$. Following the continuous-time weighting used by sCM (Lu & Song, 2024), we directly choose this product as

$$w(t,s)G(t,s) = c_{\text{skip}}(t). \tag{53}$$

We implement this product directly and do not separately evaluate $w(t,s) = c_{\text{skip}}(t)/G(t,s)$, thereby avoiding division by $G(t,s)$ near boundary configurations where it may approach zero.

Following the pseudo-objective construction of sCM, the gradient in Equation (52) can be implemented using the pseudo-objective

$$\mathcal{L}_{\text{uncorrected}} = \mathbb{E}\left[\frac{1}{2}\left\|F_\theta - F_{\theta^-} + c_{\text{skip}}(t)\frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}\right\|_2^2\right]. \tag{54}$$

Here, F_{θ^-} and $\mathrm{d}f_{\theta^-}/\mathrm{d}t$ are treated as stop-gradient quantities. Because F_θ and F_{θ^-} have the same numerical value, Equation (54) satisfies

$$\nabla_\theta \mathcal{L}_{\text{uncorrected}} = \nabla_\theta \mathbb{E}\left[c_{\text{skip}}(t) F_\theta^\top \frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}\right]. \tag{55}$$

The factor $1/2$ gives the exact gradient identity in Equation (55). This uncorrected pseudo-objective provides the starting point for the stability correction and tangent normalization derived in the following section.

D.4 JVP Rearrangement

The bridge flow tangent in Equation (50) contains the network derivative $G(t,s)\mathrm{d}F_{\theta^-}/\mathrm{d}t$. After applying the weighting in Equation (53), the corresponding contribution to the pseudo-objective is $c_{\text{skip}}(t)G(t,s)\mathrm{d}F_{\theta^-}/\mathrm{d}t$.

A direct implementation would first compute the unweighted JVP $\mathrm{d}F_{\theta^-}/\mathrm{d}t$ and multiply its output by $c_{\text{skip}}(t)G(t,s)$. In mixed-precision training, however, the unweighted tangent may attain numerically extreme values inside the network before the output scaling is applied. Following sCM (Lu & Song, 2024), we instead move this scaling into the input tangent before evaluating the JVP.

Substituting Equation (50) into Equation (54), the tangent used by the pseudo-objective is

$$\begin{aligned}
c_{\text{skip}}(t)\frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t} = c_{\text{skip}}(t)\Bigg[&\partial_t A(t,s)\boldsymbol{x}_t + A(t,s)\dot{\boldsymbol{x}}_t + \partial_t n(t,s)\boldsymbol{x}_T \\
&+ \partial_t G(t,s)F_{\theta^-} + G(t,s)\frac{\mathrm{d}F_{\theta^-}}{\mathrm{d}t}\Bigg].
\end{aligned} \tag{56}$$

Only the final term $\mathrm{d}F_{\theta^-}/\mathrm{d}t$ requires differentiating through the network. The remaining terms use analytic coefficient derivatives together with the ordinary forward output F_{θ^-}. Thus, the quantity that must be evaluated using a JVP is

$$c_{\mathrm{skip}}(t)G(t,s)\frac{\mathrm{d}F_{\theta^-}}{\mathrm{d}t}. \tag{57}$$

The four inputs to F_{θ^-} are

$$\left(c_{\mathrm{in}}(t)\boldsymbol{x}_t, c_{\mathrm{noise}}(t), c_{\mathrm{noise}}(s), c_{\mathrm{in},T}\boldsymbol{x}_T\right). \tag{58}$$

When differentiating with respect to the source time t, their corresponding tangents are

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(c_{\mathrm{in}}(t)\boldsymbol{x}_t\right) = \partial_t c_{\mathrm{in}}(t)\,\boldsymbol{x}_t + c_{\mathrm{in}}(t)\dot{\boldsymbol{x}}_t, \tag{59}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}c_{\mathrm{noise}}(t) = \partial_t c_{\mathrm{noise}}(t), \tag{60}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}c_{\mathrm{noise}}(s) = 0, \tag{61}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(c_{\mathrm{in},T}\boldsymbol{x}_T\right) = \mathbf{0}. \tag{62}$$

The destination-time and endpoint tangents vanish because s and $\boldsymbol{x}_T$ are held fixed.

For each sample, let a be a scalar, J_F the network Jacobian, and $\boldsymbol{v}$ an input tangent. By the linearity of a Jacobian-vector product,

$$aJ_F\boldsymbol{v} = J_F(a\boldsymbol{v}). \tag{63}$$

We can therefore multiply the input tangents by $c_{\mathrm{skip}}(t)G(t,s)$ before propagating them through the network, rather than multiplying the JVP output afterward. Concretely, we evaluate

$$\begin{aligned}
&c_{\mathrm{skip}}(t)\,G(t,s)\,\frac{\mathrm{d}F_{\theta^-}}{\mathrm{d}t} \\
&= \mathrm{JVP}\left[F_{\theta^-};\left(c_{\mathrm{in}}(t)\boldsymbol{x}_t, c_{\mathrm{noise}}(t), c_{\mathrm{noise}}(s), c_{\mathrm{in},T}\boldsymbol{x}_T\right);\right. \\
&\qquad\left.\left(c_{\mathrm{skip}}(t)G(t,s)\left[\partial_t c_{\mathrm{in}}(t)\,\boldsymbol{x}_t + c_{\mathrm{in}}(t)\dot{\boldsymbol{x}}_t\right], c_{\mathrm{skip}}(t)G(t,s)\partial_t c_{\mathrm{noise}}(t), 0, \mathbf{0}\right)\right].
\end{aligned} \tag{64}$$

The first tuple contains the inputs to F_{θ^-}, and the second contains their scaled source-time tangents. In implementation, the fixed destination embedding, conditioning endpoint, and class label are captured as constants. Automatic differentiation, therefore, needs to receive only the scaled bridge state and source-time embedding as differentiable JVP inputs.

Algebraically, the rearranged JVP is identical to computing the unweighted JVP and multiplying its output by $c_{\mathrm{skip}}(t)\,G(t,s)$. This rearrangement avoids the mixed-precision numerical failures observed with the direct implementation.

E Analysis of the Stability Correction

We now analyze a local sign reversal in the uncorrected update and show how the sign correction removes this reversal. For clarity, define the bridge flow tangent

$$\mathcal{T}_\theta \coloneqq \frac{\mathrm{d}f_\theta}{\mathrm{d}t}. \tag{65}$$

An exact bridge flow map is constant along a trajectory with fixed destination s, and therefore satisfies $\mathcal{T}_\theta = \mathbf{0}$. Consequently, continuous-time consistency training seeks to reduce $\mathcal{T}_\theta$'s magnitude.

Why the uncorrected update can have the wrong direction. From Equation (50), the tangent contains the term

$$\partial_t G(t,s)F_\theta, \tag{66}$$

which provides a direct, sign-sensitive dependence of the tangent $\mathcal{T}_\theta$ on the network output F_θ. At a fixed training pair (t, s), a small perturbation δF_θ yields the first-order change

$$\delta\mathcal{T}_\theta \approx \partial_t G(t,s)\,\delta F_\theta, \tag{67}$$

when the remaining terms of the tangent are held fixed locally. This approximation isolates the effect of the $\partial_t G(t,s)F_\theta$ term.

To isolate this direct effect, consider a local gradient step in function-output space, treating F_θ at the sampled input as the optimized quantity and using a step size $\eta > 0$. Because the tangent in Equation (54) is stop-gradient, the resulting local update is

$$\delta F_\theta = -\eta\, c_{\text{skip}}(t)\,\mathcal{T}_{\theta^-}. \tag{68}$$

Substituting Equation (68) into Equation (67) gives

$$\delta\mathcal{T}_\theta \approx -\eta\, c_{\text{skip}}(t)\,\partial_t G(t,s)\,\mathcal{T}_{\theta^-}. \tag{69}$$

Adding this change to the current tangent gives

$$\begin{aligned}\mathcal{T}_{\text{new}} &\approx \mathcal{T}_{\theta^-} + \delta\mathcal{T}_\theta \\ &= [1 - \eta\, c_{\text{skip}}(t)\,\partial_t G(t,s)]\,\mathcal{T}_{\theta^-}.\end{aligned} \tag{70}$$

For all settings considered in this work, $c_{\text{skip}}(t) \geq 0$ over the sampled time domain. For a sufficiently small positive step size:

- If $\partial_t G(t,s) > 0$, the multiplier in Equation (70) lies between zero and one, so the update moves the tangent toward zero.
- If $\partial_t G(t,s) < 0$, the multiplier is larger than one, so the update increases the tangent magnitude.

Thus, the uncorrected rule contracts the tangent where $\partial_t G(t,s) > 0$ but expands it where $\partial_t G(t,s) < 0$. Here, "sufficiently small" means $\eta c_{\text{skip}}(t)|\partial_t G(t,s)| < 1$.

Sign-corrected update. We compensate for this reversal by multiplying the tangent target by $\text{sign}(\partial_t G(t,s))$. The corrected local update is

$$\delta F_\theta = -\eta\, c_{\text{skip}}(t)\ \text{sign}(\partial_t G(t,s))\,\mathcal{T}_{\theta^-}. \tag{71}$$

Its corresponding effect on the bridge flow tangent is

$$\begin{aligned}\delta\mathcal{T}_\theta &\approx \partial_t G(t,s)\,\delta F_\theta \\ &= -\eta\, c_{\text{skip}}(t)\ |\partial_t G(t,s)|\,\mathcal{T}_{\theta^-}.\end{aligned} \tag{72}$$

The tangent after the corrected update is therefore

$$\mathcal{T}_{\text{new}} \approx [1 - \eta\, c_{\text{skip}}(t)\ |\partial_t G(t,s)|]\,\mathcal{T}_{\theta^-}. \tag{73}$$

For a sufficiently small positive step size, this multiplier lies in $(0, 1]$ regardless of the sign of $\partial_t G(t,s)$, with equality only when $\partial_t G(t,s) = 0$. The corrected update therefore moves the tangent $\mathcal{T}_{\text{new}}$ toward zero in both sign regions.

Occurrence of the sign change. Recall that $G(t,s) = o(t,s)c_{\text{out}}(t)$, and therefore

$$\partial_t G(t,s) = \partial_t o(t,s)c_{\text{out}}(t) + o(t,s)\partial_t c_{\text{out}}(t). \tag{74}$$

Although $G(t,s)$ is nonnegative, it need not be monotonic in the source time t. Consequently, $\partial_t G(t,s)$ can cross zero within the sampled source-destination domain.

At $s = 0$, the finite-time coefficients satisfy $o(t,0) = 1$ and $\partial_t o(t,0) = 0$, so

$$\partial_t G(t,0) = \partial_t c_{\text{out}}(t).$$

Thus, any non-monotonicity of the generalized bridge output scaling directly produces a sign change in the endpoint-training case.

Table 3: Controlled ablation of the stability correction on ImageNet (512×512) Gaussian Deblurring at 2 NFEs. Both models are evaluated after 4k training iterations and differ only in whether the correction is enabled.

Sign correction	FID-VAE ↓	CA (%) ↑	LPIPS ↓	PSNR ↑	SSIM ↑
Without	165.415	16.32	0.7362	11.32	0.3659
With (Proposed)	**4.637**	**73.05**	**0.2111**	**26.80**	**0.8375**

Corrected training target. The resulting tangent target is

$$\boldsymbol{g} = \operatorname{sign}(\partial_t G(t,s))\, c_{\text{skip}}(t) \frac{\mathrm{d}f_{\theta^-}}{\mathrm{d}t}. \tag{75}$$

We subsequently apply tangent normalization (Lu & Song, 2024):

$$\hat{\boldsymbol{g}} = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_{\text{RMS}} + c}, \qquad c = 0.1. \tag{76}$$

The RMS norm is computed independently for each sample:

$$\|\boldsymbol{g}\|_{\text{RMS}} \coloneqq \sqrt{\frac{1}{d}\|\boldsymbol{g}\|_2^2}, \tag{77}$$

where d is the number of scalar elements in that sample. The normalization rescales each sample by a positive scalar and therefore does not change the update direction established above. Neither the sign correction nor the normalization changes the desired consistency fixed point: if $\mathrm{d}f_{\theta^-}/\mathrm{d}t = \mathbf{0}$, then the training target vanishes with or without these operations. This local calculation isolates the sign reversal caused by the direct $\partial_t G(t,s) F_\theta$ term. In Section E.1, we test the practical effect predicted by this analysis using a controlled neural-network training experiment.

E.1 Controlled Sign-Correction Ablation

We evaluate the practical effect of the sign correction using ImageNet (512×512) Gaussian Deblurring. This setting has nonzero covariance between the clean and degraded endpoints, causing the bridge output scaling, and hence $G(t,s)$, to become non-monotonic. During training, approximately 2-3% of sampled source-destination pairs satisfy $\partial_t G(t,s) < 0$.

We train two models from the same DBM initialization for 4k iterations, using the same data, time sampler, optimizer, learning rate, batch size, and continuous-time training objective. The only difference is whether the sign correction in Equation (75) is applied. Both models are evaluated at 2 NFEs on the same 10k ImageNet validation images (Liu et al., 2023).

As shown in Table 3, removing the sign correction causes severe degradation across every evaluation metric. Notably, the uncorrected run retains finite gradient norms throughout its training budget. Its failure is therefore not conventional numerical divergence: training remains numerically finite even though the resulting sampling map is severely degraded. This behavior agrees with the local analysis above: when $\partial_t G(t,s) < 0$, the direct $\partial_t G(t,s) F_\theta$ contribution reverses the local effect of the uncorrected update. Our correction removes this sign reversal.

F Analysis of the Bridge-Structured Parameterization

The continuous-time training rule can be used with different parameterizations of the flow map. To examine the importance of the bridge structure, we compare our parameterization with the linear-displacement form used by AYF and the closely related average-velocity parameterization of MeanFlow and AlphaFlow (Sabour et al., 2025; Geng et al., 2026; Zhang et al., 2026). This comparison tests the practical effect of retaining the finite-time bridge structure under a matched architecture and training budget.

F.1 Average-Velocity Parameterization

Consider the flow-map parameterization

$$f_\theta^{\text{AV}}(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) = \boldsymbol{x}_t + (s-t) V_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T), \tag{78}$$

Table 4: Parameterization comparison on ImageNet (512×512) Gaussian deblurring at 2 NFEs. Both methods are evaluated using the same sampling schedule and the same fixed 10k-image validation subset used in our main ImageNet experiments.

Parameterization	FID-VAE ↓	CA (%) ↑	LPIPS ↓	PSNR ↑	SSIM ↑
Average velocity	250.438	0.15	0.7889	6.56	0.0317
Bridge-structured (ours)	**4.081**	**74.22**	**0.2078**	**26.74**	**0.8342**

where V_θ predicts the average velocity over the interval from t to s. The exact average velocity along the corresponding trajectory is

$$V^\star(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) = \frac{\boldsymbol{x}_s - \boldsymbol{x}_t}{s - t}. \tag{79}$$

Therefore, $f_\theta^{\mathrm{AV}} = \boldsymbol{x}_s$ whenever $V_\theta = V^\star$. The parameterization also satisfies the boundary condition $f_\theta^{\mathrm{AV}}(\boldsymbol{x}_t, t, t, \boldsymbol{x}_T) = \boldsymbol{x}_t$.

Sensitivity over long intervals. Suppose that the predicted average velocity has error $\boldsymbol{\varepsilon}_V$:

$$V_\theta = V^\star + \boldsymbol{\varepsilon}_V. \tag{80}$$

The resulting flow-map error is then

$$f_\theta^{\mathrm{AV}} - \boldsymbol{x}_s = (s - t)\boldsymbol{\varepsilon}_V. \tag{81}$$

Thus, the effect of a velocity-prediction error increases with the interval length $|t - s|$. This is particularly demanding near the initial sampling boundary where $t \approx T$, where a one-step prediction must traverse almost the entire bridge.

Our bridge-structured parameterization instead takes the form

$$f_\theta = H(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T) + G(t, s)F_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T), \tag{82}$$

where H incorporates the known linear evolution of the bridge and $G(t, s)$ is determined by its exact finite-time coefficients. If $F_\theta = F^\star + \boldsymbol{\varepsilon}_F$, the corresponding flow-map error is

$$f_\theta - \boldsymbol{x}_s = G(t, s)\boldsymbol{\varepsilon}_F. \tag{83}$$

Hence, the network error enters through the bridge-dependent gain $G(t, s)$ rather than directly through the raw interval length $s - t$.

Both parameterizations can represent the target flow map in principle. We therefore compare their optimization behavior rather than treating the average-velocity form as less expressive. The average-velocity form represents the complete finite-time displacement through a quantity whose prediction error is multiplied by the raw interval length $|t - s|$. In contrast, the bridge-structured form incorporates the known linear bridge evolution analytically, and its remaining network error is scaled by the bridge-dependent gain $G(t, s)$.

F.2 Empirical Comparison

We compare the two parameterizations on ImageNet (512×512) Gaussian deblurring. Both models use the same backbone, adapted EDM2 initialization, training data, source–destination time distribution, optimizer, learning rate, batch size, and 4k-iteration training budget. Each parameterization uses its corresponding output scaling, tangent weighting, and sign convention.

As shown in Table 4, the average-velocity parameterization fails to produce useful two-step predictions in this diffusion-bridge setting, attaining a FID-VAE of 250.438 and a classifier accuracy of only 0.15%. Under the same architecture, training budget, and sampling schedule, the bridge-structured parameterization attains a FID-VAE of 4.081 and a classifier accuracy of 74.22%, while also substantially improving every paired reconstruction metric. Under this matched training setup, these results are consistent with the long-interval conditioning analysis: retaining the finite-time bridge structure leads to substantially better endpoint and few-step prediction quality.

Table 5: Endpoint-prediction MSE at $s = 0$ on 256 held-out latent pairs. "Bridge state" evaluates the model on a marginal state at the indicated source time. "Conditioning endpoint" evaluates the model on $\boldsymbol{x}_T$ at $t = T = 80$, matching the first model input of the two-NFE sampling procedure.

Parameterization	Bridge state $t = 40$	Bridge state $t = 79.5$	Conditioning endpoint $t = T = 80$
Average Velocity	26.089	179.532	175.845
Bridge-Structured (Ours)	**0.134**	**0.727**	**0.727**

The endpoint diagnostic in Table 5 examines how the prediction error changes as the source-destination interval grows. The average-velocity error grows rapidly with the source time and exceeds the bridge-structured error by more than two orders of magnitude near the initial sampling boundary. The same behavior occurs when the input is the conditioning endpoint, showing that the failure directly affects the state from which sampling begins. These results support the long-interval conditioning analysis in Equations (81) and (83).

G Implementation and Experimental Details

We provide the implementation and evaluation details needed to reproduce our experiments. We describe the model architecture and initialization, time sampling, optimization, inference procedure, and task-specific evaluation protocols.

G.1 Architecture and Initialization

For each task, BFM uses the same UNet backbone as its corresponding teacher DBM. To condition the flow map on an arbitrary destination time, we augment the network with a second time-embedding branch for s. The source and destination times use the same time-conditioning transformation defined in Equation (27).

We initialize the interval-denoising component of f_θ from the corresponding pretrained DBM without changing the denoising function it represents. Specifically, the source- and destination-time embeddings are concatenated before the original time-conditioning projection. We copy the DBM projection weights to the source-time portion and initialize the destination-time portion to zero. Consequently, $\bar{\boldsymbol{D}}_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T)$ is initially independent of s and identical to $D_{\mathrm{pre}}(\boldsymbol{x}_t, t, \boldsymbol{x}_T)$. In particular, because the structured flow map reduces to $\bar{\boldsymbol{D}}_\theta$ at $s = 0$, its initialized endpoint map exactly recovers the pretrained DBM.

For ImageNet (512×512) restoration, we use latent-space EDM2-based backbones: EDM2-XS for $4\times$ super-resolution and EDM2-S for Gaussian deblurring. For E2H and DIODE, we use pixel-space EDM2-based UNets. Apart from the additional destination-time conditioning, each BFM model inherits the architecture and initialization of its corresponding DBM.

G.2 Source–Destination Time Sampling

We sample source-destination pairs (t, s) using an interval distribution inspired by AYF (Sabour et al., 2025). Because the diffusion time covers a large dynamic range, intervals are constructed in the transformed coordinate

$$\tau(u) = \arctan(u). \tag{84}$$

Let $\tau_{\min} = \tau(t_{\min})$ and $\tau_{\max} = \tau(t_{\max})$. We first sample a proposed normalized interval length

$$r = \mathrm{sigmoid}(z_\Delta), \qquad z_\Delta \sim \mathcal{N}(\mu_\Delta, \sigma_\Delta^2), \tag{85}$$

and set $\Delta_\tau^{\mathrm{prop}} = r(\tau_{\max} - \tau_{\min})$.

For ImageNet restoration and DIODE, we independently draw the source time from a log-normal distribution,

$$\widetilde{t} = \exp(z_t), \qquad z_t \sim \mathcal{N}(\mu_t, \sigma_t^2), \tag{86}$$

and clip it to $[t_{\min}, t_{\max}]$. The interval is shortened only when it would cross the lower boundary:

$$\Delta_\tau = \min\{\Delta_\tau^{\text{prop}}, \tau(t) - \tau_{\min}\}\,, \qquad s = \tan(\tau(t) - \Delta_\tau)\,. \tag{87}$$

For E2H, we instead place the proposed interval uniformly in its feasible range in the τ coordinate.

We use $(\mu_\Delta, \sigma_\Delta) = (-0.6, 1.6)$ throughout. For the log-normal source-time variant, we use $(\mu_t, \sigma_t) = (-1, 1.6)$. We set $T = 80$, $t_{\min} = 0.002$, and $t_{\max} = 79.5$, keeping the source time below the singular endpoint T.

A continuous distribution assigns zero probability to the boundary cases encountered during sampling. We therefore add discrete atoms at $s = 0$ and $t = t_{\max}$. For the main full-flow models, the direct $s = 0$ atom has probability 0.30. The $t = t_{\max}$ atom has probability 0.02 for ImageNet restoration and 0.15 for E2H and DIODE. Within this source-boundary atom, we additionally sample the longest interval $(t_{\max}, 0)$ with probability 0.25.

For the endpoint-directed ImageNet ablation, the direct $s = 0$ branch has probability 0.98, while the remaining 0.02 is assigned to the $t = t_{\max}$ source-boundary branch. As above, one quarter of the source-boundary branch also uses $s = 0$, giving a total probability $\Pr(s = 0) = 0.985$. This makes the training distribution effectively endpoint-directed while retaining a small number of source-boundary examples with $s > 0$.

G.3 Optimization and Training Configuration

We initialize each BFM model from its corresponding trained DBM and finetune it using the continuous-time objective. At each iteration, we construct the intermediate state $\boldsymbol{x}_t$ from the paired endpoints $(\boldsymbol{x}_0, \boldsymbol{x}_T)$ and use the known conditional bridge dynamics to compute an analytic unbiased estimator of its marginal Bridge PF-ODE velocity $\mathrm{d}\boldsymbol{x}_t/\mathrm{d}t$. Thus, the DBM supplies the initialization of the flow-map model but is not evaluated as a separate teacher during BFM training. We use this same training procedure for all four tasks.

We optimize the BFM pseudo-objective using the task-specific configurations reported in Table 6. We use no dropout or weight decay and keep the learning rate constant throughout training. Exponential moving averages of the model parameters are maintained for evaluation. The ImageNet (512×512) and DIODE models are trained in BF16, whereas the E2H model is trained in FP16. Task-specific learning rates, batch sizes, training lengths, and EMA rates are reported in Table 6.

Unless otherwise stated for the sign-correction ablation, all main models use the sign-corrected tangent target from Equation (75), as described in Section E. We use the FlashAttention-compatible JVP implementation of Morehead (2025) for efficient computation (Dao et al., 2022; Dao, 2024).

Teacher DBMs. For the ImageNet (512×512) tasks, we initialize the teacher models from publicly available pretrained EDM2 diffusion models and modify them to accept the conditioning endpoint $\boldsymbol{x}_T$. We then finetune them as DBMs for 40k iterations using denoising bridge score matching (Zhou et al., 2023). We use the EDM2-XS configuration for $4\times$ super-resolution and the EDM2-S configuration for Gaussian deblurring. For E2H and DIODE, we train the corresponding EDM2-based DBMs from scratch for 200k and 328k iterations, respectively. Each BFM student is subsequently initialized from its corresponding trained DBM.

Matched IBMD reproduction. For the ImageNet restoration comparison, we reproduce IBMD (Gushchin et al., 2025) using the same task-specific EDM2-based teacher DBMs used by CDBM, BFM, and the numerical solvers. Following its multistep training protocol, we initialize the student and auxiliary bridge model from the corresponding teacher DBM and train an IBMD student for the reported 2-NFE budget. For the reported 2-NFE results, we use a global batch size of 256, RAdam with a learning rate of 3×10^{-6}, an EMA decay of 0.99, and 3k student updates, with five auxiliary-bridge updates per student update. All methods are evaluated on the same 10k-image validation subset under the same protocol.

IBMD alternates between optimizing the student and an auxiliary bridge model. Its authors report that student optimization requires backpropagation through the teacher, auxiliary bridge, and generator, resulting in approximately three times the training-memory cost of training the teacher (Gushchin et al., 2025). In contrast, BFM finetunes a single DBM-initialized model, and one checkpoint supports all reported NFE budgets.

Table 6: **Training configurations for the main BFM.** "Checkpoint" denotes the BFM iteration used to report the main results. All BFM models are finetuned from their corresponding trained DBMs using paired endpoints. The ImageNet teacher DBMs are initialized from pretrained EDM2 models, whereas the E2H and DIODE teacher DBMs are trained from scratch.

Setting	$4\times$ Super-Resolution	Gaussian Deblurring	E2H	DIODE
Student backbone	EDM2-XS	EDM2-S	EDM2-based UNet	EDM2-based UNet
Teacher DBM training	Finetuned from EDM2	Finetuned from EDM2	From scratch	From scratch
BFM training mode	Finetuning	Finetuning	Finetuning	Finetuning
Optimizer	RAdam	RAdam	AdamW	RAdam
Learning rate	5.7×10^{-6}	5.7×10^{-6}	1×10^{-6}	1×10^{-6}
Global batch size	1024	1024	64	32
Training precision	BF16	BF16	FP16	BF16
Dropout %	0	0	0	0
Weight decay	0	0	0	0
EMA used for evaluation	0.999	0.999	0.999	0.999
Reported checkpoint	4k	4k	15k	5k

G.4 Sampling Details

We sample along a decreasing schedule

$$T = t_K > t_{K-1} > \cdots > t_1 > t_0 = 0, \tag{88}$$

starting from the conditioning endpoint $\boldsymbol{x}_{t_K} = \boldsymbol{x}_T$. All reported results use one initial stochastic step. Specifically, the first model evaluation predicts the endpoint

$$\hat{\boldsymbol{x}}_0 = f_\theta(\boldsymbol{x}_T, T, 0, \boldsymbol{x}_T),$$

after which we sample the first interior state as

$$\boldsymbol{x}_{t_{K-1}} = a_{t_{K-1}}\boldsymbol{x}_T + b_{t_{K-1}}\hat{\boldsymbol{x}}_0 + c_{t_{K-1}}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{89}$$

This stochastic initialization is necessary because the bridge begins at the fixed endpoint $\boldsymbol{x}_T$ and its PF-ODE is singular at $t = T$, as also observed for diffusion-bridge solvers (Venugopal et al., 2026; Zheng et al., 2024).

We count each evaluation of f_θ as one NFE. The stochastic initialization and every subsequent transition each require exactly one model evaluation; sampling Gaussian noise and applying the analytic forward bridge transition incur no additional NFEs. Consequently, the reported 2-, 4-, and 8-NFE results use exactly 2, 4, and 8 network evaluations, respectively, including the initial stochastic step. The time nodes and γ values used for each task are reported in Table 7.

Following γ-sampling (Kim et al., 2023), after the initial stochastic step, a nonterminal update targeting the next schedule node t_{k-1} first evaluates the flow map at

$$u_k = \sqrt{1-\gamma^2}\, t_{k-1} \tag{90}$$

and then applies the exact forward bridge transition from u_k to t_{k-1}. Thus, $\gamma = 0$ gives a direct deterministic flow-map transition, whereas $\gamma = 1$ predicts the endpoint at $u_k = 0$ before resampling the next bridge state. Intermediate values interpolate between these cases.

At 2 NFEs, γ does not affect the generated sample: the first evaluation performs the stochastic initialization, and the second performs the final deterministic transition to $s = 0$. The final transition to $s = 0$ is always deterministic.

G.5 Datasets, Degradations, and Evaluation Protocols

ImageNet restoration. We evaluate $4\times$ super-resolution and Gaussian deblurring on a fixed 10k-image subset of the ImageNet validation set at (512×512) used by I^{2}SB (Liu et al., 2023). For super-resolution, we bicubically downsample each image from 512×512 to 128×128 with antialiasing. The low-resolution image is bicubically resized back to 512×512 before VAE encoding. For deblurring, we apply a 21×21 Gaussian kernel with standard deviation 3.0 and reflection padding. The clean and degraded images are encoded using the `stabilityai/sd-vae-ft-mse` VAE, and generated latents are decoded using the same VAE.

Table 7: **Sampling configurations used for the main results.** $\mathrm{Karras}_7(t_1)$ denotes a Karras schedule with $\rho = 7$ whose last nonzero node is fixed to the anchor t_1, followed by the terminal node $t_0 = 0$. For E2H, we instead list the complete schedule explicitly. All configurations use one initial stochastic step.

Task	NFE	Schedule	γ
$4\times$ Super-Resolution	2	$\mathrm{Karras}_7(8.0)$	1.0
	4	$\mathrm{Karras}_7(1.0)$	1.0
	8	$\mathrm{Karras}_7(0.75)$	0.9
Gaussian Deblurring	2	$\mathrm{Karras}_7(16.0)$	1.0
	4	$\mathrm{Karras}_7(2.0)$	1.0
	8	$\mathrm{Karras}_7(0.5)$	1.0
E2H	2	$[80,\ 0.30,\ 0]$	0.0
	4	$[80,\ 1.00,\ 0.50,\ 0.30,\ 0]$	0.0
	8	$[80,\ 8,\ 6,\ 4,\ 2,\ 1,\ 0.5,\ 0.3,\ 0]$	0.0
DIODE	2	$\mathrm{Karras}_7(0.15)$	1.0
	4	$\mathrm{Karras}_7(0.15)$	1.0
	8	$\mathrm{Karras}_7(0.15)$	1.0

For all ImageNet comparisons, we evaluate every method on the same image identifiers. FID-VAE is computed between generated images and the corresponding VAE-reconstructed clean images. Before extracting Inception-v3 features, both sets of images are resized to 299×299 using the `legacy_pytorch` preprocessing of clean-fid. Because its reference distribution consists of VAE reconstructions, FID-VAE is directly comparable only among methods evaluated under this same protocol. We additionally report top-1 classification accuracy using a pretrained ImageNet ResNet-50 classifier. For the controlled ablations in the Appendix, we also report LPIPS, PSNR, and SSIM against the aligned VAE-reconstructed clean targets at their native resolution.

E2H and DIODE. Following the evaluation protocol used by prior diffusion-bridge methods (Zhou et al., 2023; Zheng et al., 2024; Venugopal et al., 2026; He et al., 2024), we evaluate E2H and DIODE on their training splits. The E2H evaluation contains 138,567 paired edge-image examples at (64×64), while the DIODE evaluation contains 16,502 paired surface-normal-RGB examples at (256×256). We report FID and Inception Score using the same evaluation implementation as the prior methods. LPIPS and MSE are computed between each generated image and its aligned ground-truth target. Because this legacy protocol evaluates the training splits, we separately provide a teacher-relative analysis on the held-out test splits in Section I.2.

H Additional Ablations and Analyses

H.1 Full-Flow versus Endpoint-Directed Training

We isolate the effect of learning arbitrary destination times (*i.e.*, the full flow) using a matched ablation on ImageNet (512×512) Gaussian deblurring. Both variants use the same architecture, are initialized from the same trained DBM checkpoint, and use the same optimizer, learning rate, batch size, random seed, continuous-time finetuning objective, and 4k-iteration training budget. The only difference is the destination-time distribution.

The full-flow model uses the mixed destination sampler described in Section G.2, which assigns a direct probability mass of 0.30 to $s = 0$ and otherwise supervises intermediate destinations $0 < s < t$. For the endpoint-directed control, the direct $s = 0$ branch has probability 0.98, while the remaining 0.02 is assigned to the $t = t_{\max}$ source-boundary branch. One quarter of that boundary branch also sets $s = 0$, giving $\Pr(s = 0) = 0.985$; the remaining 1.5% of samples use $t = t_{\max}$ with $s > 0$.

At evaluation time, both variants use identical sampling schedules, stochasticity settings, and the same 10k-image ImageNet I^{2}SB validation subset. All configurations in this ablation use $\gamma = 1$.

Table 8: Matched finetuning comparison of full-flow and endpoint-directed training on ImageNet (512×512) Gaussian deblurring. The variants differ only in their destination-time distribution. All metrics are computed on the same 10k validation images.

Training	NFE	FID-VAE ↓	CA (%) ↑	LPIPS ↓	PSNR ↑	SSIM ↑
Endpoint-directed	2	5.840	72.30	0.2290	26.03	0.8134
	4	2.972	74.92	0.2041	26.73	0.8246
	8	2.163	75.04	0.1997	26.26	0.8108
Full flow	2	**4.056**	**74.48**	**0.2078**	**26.73**	**0.8341**
	4	**2.187**	**75.04**	**0.1927**	**26.76**	**0.8255**
	8	**1.991**	**75.16**	**0.1955**	26.26	**0.8117**

Consequently, every neural flow-map evaluation uses destination $s = 0$; the subsequent nonzero bridge state is constructed analytically through the stochastic forward bridge transition. Thus, the full-flow model is never queried at an intermediate destination $s > 0$ during this comparison.

Across the three NFE budgets in Table 8, full-flow training reduces FID-VAE by 30.5%, 26.4%, and 7.9% at 2, 4, and 8 NFEs, respectively. It consistently improves LPIPS and SSIM, while PSNR is improved or comparable. CA improves at 2 NFEs and remains comparable at 4 and 8 NFEs. Because inference never queries an intermediate destination $s > 0$ in this ablation, these improvements indicate that intermediate-destination supervision also improves the learned endpoint map.

H.2 Intermediate-Destination Accuracy and Flow Composition

The preceding ablation examines whether full-flow supervision improves endpoint sampling. We next evaluate the capability that this supervision is intended to learn: mapping a bridge state $\boldsymbol{x}_t$ to arbitrary destination times $s < t$. The exact bridge flow satisfies the composition property

$$\Phi_{s\to r}(\Phi_{t\to s}(\boldsymbol{x}_t; \boldsymbol{x}_T); \boldsymbol{x}_T) = \Phi_{t\to r}(\boldsymbol{x}_t; \boldsymbol{x}_T), \qquad r \le s \le t. \tag{91}$$

We therefore also examine whether composing two learned bridge-flow transitions yields approximately the same result as one direct transition.

For each triplet $t > s > r$, we sample $\boldsymbol{x}_t$ from the bridge marginal and integrate the teacher Bridge PF-ODE from t to s and r, obtaining the reference states $\boldsymbol{x}_s^\star$ and $\boldsymbol{x}_r^\star$. We then compute

$$\begin{aligned} \hat{\boldsymbol{x}}_s &= f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T), \\ \hat{\boldsymbol{x}}_{r,\mathrm{direct}} &= f_\theta(\boldsymbol{x}_t, t, r, \boldsymbol{x}_T), \\ \hat{\boldsymbol{x}}_{r,\mathrm{composed}} &= f_\theta(\hat{\boldsymbol{x}}_s, s, r, \boldsymbol{x}_T). \end{aligned}$$

These predictions support two complementary evaluations. First, we measure intermediate-destination accuracy by comparing $\hat{\boldsymbol{x}}_s$ and $\hat{\boldsymbol{x}}_{r,\mathrm{direct}}$ with the corresponding teacher states $\boldsymbol{x}_s^\star$ and $\boldsymbol{x}_r^\star$. Second, we measure flow-composition consistency by comparing $\hat{\boldsymbol{x}}_{r,\mathrm{composed}}$ with $\hat{\boldsymbol{x}}_{r,\mathrm{direct}}$. We also compare $\hat{\boldsymbol{x}}_{r,\mathrm{composed}}$ with $\boldsymbol{x}_r^\star$ to determine whether the composed transition remains accurate with respect to the teacher trajectory. The direct-versus-composed comparison tests the approximate composition property

$$f_\theta(\boldsymbol{x}_t, t, r, \boldsymbol{x}_T) \approx f_\theta(f_\theta(\boldsymbol{x}_t, t, s, \boldsymbol{x}_T), s, r, \boldsymbol{x}_T)\,. \tag{92}$$

We perform this diagnostic on held-out ImageNet (512×512) Gaussian deblurring latent pairs using the time triplets $(t, s, r) = (20, 2, 0.2)$ and $(8, 1, 0.1)$. Teacher references are obtained using 32-step Heun integration on a logarithmic time grid. We additionally compare 16- and 32-step teacher trajectories to verify that numerical integration error is substantially smaller than the measured student errors. All comparisons are measured using per-element MSE in latent space.

The 16- and 32-step teacher trajectories differ by at most 1.18×10^{-5} MSE across the evaluated destinations, which is more than two orders of magnitude below the student errors in Table 9. The measurements are therefore not explained by numerical error in the teacher reference.

Table 9: Intermediate-destination accuracy and flow-composition consistency on 256 held-out ImageNet (512×512) Gaussian-deblurring latent pairs. The $t \to s$ and $t \to r$ columns compare direct predictions with the corresponding teacher states. The $t \to s \to r$ column compares the composed prediction with the teacher state at r, while "Direct–composed" compares the direct and composed predictions. All entries are per-element MSE (↓).

Time triplet	Training	$t \to s$	$t \to r$	$t \to s \to r$	Direct–composed
$(20, 2, 0.2)$	Endpoint-directed	0.008629	0.043870	0.033755	0.017785
	Full flow	**0.002209**	**0.017978**	**0.008295**	**0.010172**
$(8, 1, 0.1)$	Endpoint-directed	0.008617	0.010159	0.029054	0.020586
	Full flow	**0.001793**	**0.008984**	**0.005004**	**0.012178**

Table 10: Comparison with matched EDM2-based DBM teachers on E2H and DIODE with the legacy training-split protocol. The teacher DBM is sampled using DBMSolver (Venugopal et al., 2026). Each BFM is finetuned from its corresponding teacher DBM.

Dataset	Method	NFE ↓	FID ↓	IS ↑
Edges2Handbags	EDM2 DBM Teacher	6	0.5715	3.6556
	EDM2 DBM Teacher	10	0.6403	3.6678
	EDM2 DBM Teacher	20	0.3912	3.7000
	EDM2 DBM Teacher	30	0.3856	3.7118
	BFM	2	0.5231	3.6769
	BFM	4	0.4849	3.6769
	BFM	8	0.4772	3.6760
DIODE	EDM2 DBM Teacher	6	3.4688	6.1476
	EDM2 DBM Teacher	10	3.6024	6.1352
	EDM2 DBM Teacher	20	1.2788	6.1249
	EDM2 DBM Teacher	30	1.1100	6.1190
	BFM	2	1.6903	6.1055
	BFM	4	1.6067	6.1221
	BFM	8	1.3878	6.1625

Full-flow supervision reduces the intermediate $t \to s$ error by factors of 3.9 and 4.8 for the two time triplets. It reduces the composed prediction error by factors of 4.1 and 5.8, and reduces the direct–composed discrepancy by approximately $1.7\times$ in both cases. Moreover, for the full-flow model, the composed prediction is closer to the teacher state than the direct $t \to r$ prediction for both triplets. On both evaluated time triplets, full-flow supervision substantially improves intermediate-destination accuracy and reduces the discrepancy between direct and composed transitions.

I Additional Quantitative Results

I.1 Comparison with Matched EDM2-Based DBM Teachers

The published E2H and DIODE baselines in the main comparison primarily use ADM-based UNets, whereas our BFM models use EDM2-based UNets. To contextualize this backbone difference, we compare each BFM with the corresponding EDM2-based DBM from which it was finetuned. We sample the matched DBM teachers using DBMSolver and evaluate all methods under the same legacy training-split protocol. This comparison isolates the acceleration obtained by BFM relative to its matched teacher. We report the results in Table 10.

At 2 NFEs, BFM outperforms the matched teacher sampled with 6 or 10 NFEs in FID on both datasets. Increasing BFM to 8 NFEs further narrows the gap to the 20- to 30-NFE teacher solutions, particularly on DIODE. The high-NFE teachers retain the best converged FID, as expected. These results show that the few-step improvements cannot be attributed solely to using an EDM2-

Table 11: Teacher-relative evaluation on the held-out E2H and DIODE test splits, containing 200 and 446 paired examples, respectively. We compare 2-NFE BFM with its corresponding DBM teacher sampled using 30-NFE DBMSolver. Because the test splits are too small for reliable FID or IS estimation, we report paired metrics only.

Dataset	Method	NFE	LPIPS ↓	MSE ↓	PSNR ↑	SSIM ↑
E2H	DBM teacher	30	0.2025	0.0680	21.218	0.7350
	BFM	2	**0.2006**	**0.0645**	**21.447**	**0.7373**
DIODE	DBM teacher	30	**0.4410**	**0.2378**	**15.780**	0.2308
	BFM	2	0.4464	0.2429	15.689	**0.2338**

based backbone: BFM obtains a substantially better low-NFE quality-cost tradeoff than numerical integration of its matched DBM teacher.

I.2 Held-Out Evaluation on E2H and DIODE

The established E2H and DIODE benchmark evaluates models on their training splits. We additionally examine whether few-step acceleration introduces further degradation on held-out inputs by comparing BFM with the corresponding DBM teachers from which it was finetuned. The E2H and DIODE test splits contain only 200 and 446 paired examples, respectively, making distributional metrics such as FID and IS unreliable at this sample size. We therefore report paired perceptual and reconstruction metrics.

Although both the teachers and their BFM students exhibit a substantial shift relative to the legacy training-split results, 2-NFE BFM remains close to the corresponding 30-NFE teacher. On E2H, BFM matches or slightly improves the teacher across the paired metrics; on DIODE, the remaining differences are small. These results indicate that BFM introduces little additional degradation relative to its matched teacher while reducing inference from 30 to 2 NFEs.

J Additional Qualitative Results

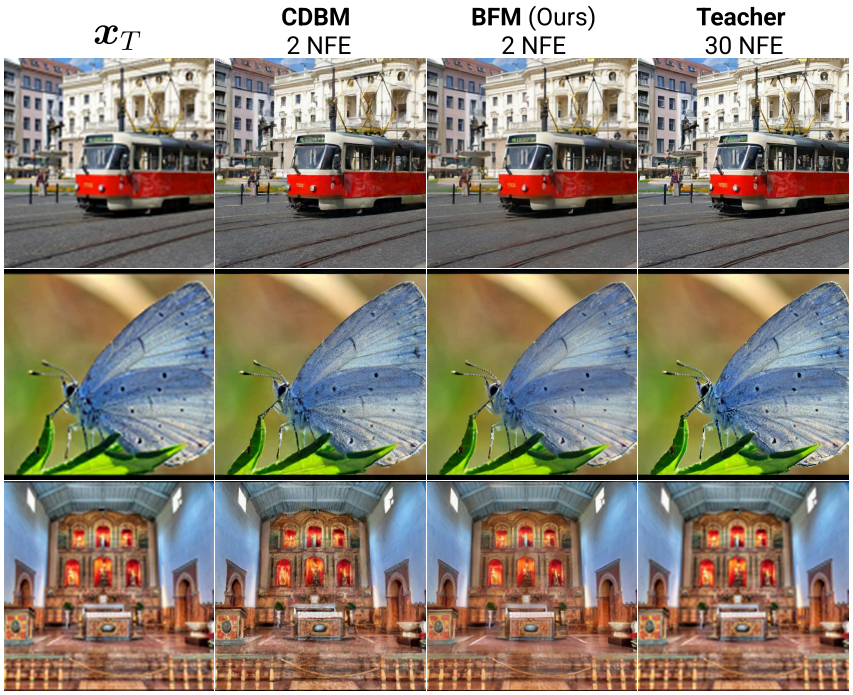

Figure 4: Qualitative comparison on ImageNet 512×512 $4\times$SR at 2 NFEs. From left to right: the low-resolution conditioning input $\boldsymbol{x}_T$, CDBM (He et al., 2024) (2 NFE), BFM (2 NFE, ours), and the matched DBM teacher sampled using DBMSolver (Venugopal et al., 2026) at 30 NFEs. Because $4\times$SR is underdetermined, the methods may recover different high-frequency details. Nevertheless, BFM produces sharp, semantically consistent reconstructions and recovers much of the fine structure produced by the 30-NFE teacher in just 2 NFEs.

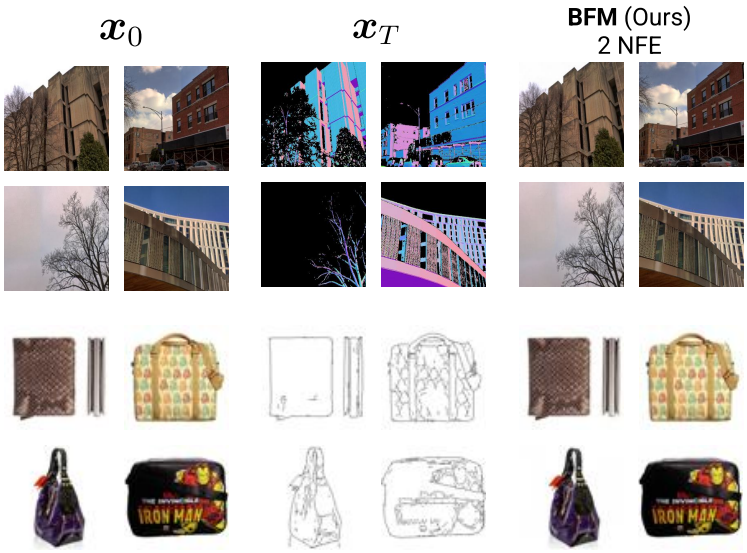

Figure 5: Representative 2-NFE BFM translations on DIODE (top) and Edges2Handbags (bottom). From left to right, each row shows the target image $\boldsymbol{x}_0$, the conditioning input $\boldsymbol{x}_T$, and the BFM output. These examples complement the quantitative results reported in Table 2.