LIGHTSBB-M: BRIDGING SCHRÖDINGER AND BASS FOR GENERATIVE DIFFUSION MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

The Schrödinger Bridge and Bass (SBB) formulation, which jointly controls drift and volatility, is an established extension of the classical Schrödinger Bridge (SB). Building on this framework, we introduce **LightSBB-M**, an algorithm that computes the optimal SBB transport plan in only a few iterations. The method exploits a dual representation of the SBB objective to obtain analytic expressions for the optimal drift and volatility, and it incorporates a tunable parameter $\beta > 0$ that interpolates between pure drift (the Schrödinger Bridge) and pure volatility (Bass martingale transport). On a Gaussian case with a closed-form solution, we verify that the algorithm recovers the optimal joint plan and both optimal controls, not merely the terminal marginal. We show that LightSBB-M achieves the lowest 2-Wasserstein distance on synthetic datasets against state-of-the-art SB and diffusion baselines with an average of 19% improvement. We also illustrate the generative capability of the framework on an unpaired image-to-image translation task (*adult* $\rightarrow$ *child* faces in FFHQ). These findings demonstrate that **LightSBB-M** provides a scalable, high-fidelity SBB solver that outperforms existing SB and diffusion baselines across both synthetic and real-world generative tasks.

1 INTRODUCTION

Stochastic-process-based generative modeling has recently gained significant attention, notably through diffusion models and Schrödinger Bridge (SB) formulations, which construct a controlled stochastic dynamics transporting a source distribution μ_0 to a target distribution μ_T Schrödinger (1932); Léonard (2014); De Bortoli et al. (2021). By minimizing the relative entropy with respect to a Brownian prior, SB establishes deep connections between optimal transport, stochastic control, and score-based learning. However, the classical SB framework relies on a fixed volatility, which constrains the class of dynamics it can represent.

Bass martingale transport provides an alternative formulation by optimizing over volatility while enforcing a zero drift constraint Bass (1983); Backhoff et al. (2025). While this approach allows for greater flexibility on the marginals, it departs from diffusion-based generative mechanisms and lacks an explicit drift control. The Schrödinger-Bass Bridge (SBB) problem Henry-Labordere et al. (2026) unifies these two paradigms by jointly optimizing over drift and volatility, introducing a parameter $\beta > 0$ that interpolates between the classical SB limit ($\beta \to \infty$) and the Bass martingale transport regime ($\beta \to 0$). The practical gain is expressive: the volatility is no longer fixed by the prior but becomes state-dependent, learned jointly with the drift. This is complementary to the recent literature that widens the noise family itself, replacing Gaussian assumptions by α-stable noise or Student-t perturbation kernels Shariatian et al. (2025); Pandey et al. (2025). SBB instead retains a constant-volatility bridge in the transformed Y-space and gains flexibility through a learned transport-induced state-dependent volatility in the original space.

Despite its theoretical appeal, the SBB problem has received limited attention from a computational perspective, especially in high-dimensional generative modeling. Extending existing SB solvers to this setting is challenging due to the presence of stochastic volatility and implicit transport maps.

In this work, we introduce **LightSBB-M**, a simulation-efficient algorithm for solving the SBB problem. Leveraging a dual formulation of SBB and recent advances in Light Schrödinger Bridge matching Gushchin et al. (2024), our approach yields analytic expressions for the optimal drift and volatility and converges in only a few iterations.

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.

Our main contributions are:

- We propose the first practical and scalable algorithm to compute the optimal Schrödinger-Bass Bridge, enabling efficient generative modeling with jointly learned drift and volatility.
- We derive explicit and tractable expressions for the optimal SBB controls, allowing simulation-free sampling and avoiding costly SDE discretization.
- We verify against a closed-form Gaussian solution that the algorithm recovers the optimal joint plan and both optimal controls, not merely the terminal marginal that transport benchmarks usually report.
- We provide extensive numerical experiments on synthetic and high-dimensional datasets, demonstrating improved transport accuracy and generative performance compared to state-of-the-art Schrödinger Bridge and diffusion-based baselines.

This work advances the interface between optimal transport theory and modern generative modeling by making stochastic-volatility transport computationally tractable at scale. Beyond images, the framework opens new directions for generative modeling under distributional constraints and time-series with heteroskedasticity—settings that remain challenging for standard diffusion models.

2 Background

2.1 The Schrödinger Bridge Problem

Let $\Omega = \mathcal{C}([0,T],\mathbb{R}^d)$, $X = (X_t)_{t\in[0,T]}$ the canonical process, and $\mathbb{W}^\varepsilon$ the Wiener measure with variance $\varepsilon > 0$. The Schrödinger Bridge Problem (SBP) Schrödinger (1932) seeks the path measure closest to $\mathbb{W}^\varepsilon$ matching prescribed marginals:

$$\mathbb{P}^* \in \operatorname*{argmin}_{\mathbb{P}} \big\{ \mathrm{KL}(\mathbb{P}\|\mathbb{W}^\varepsilon) : \mathbb{P}_0 = \mu_0,\, \mathbb{P}_T = \mu_T \big\}. \tag{1}$$

When $\mathrm{KL}(\mathbb{P}\|\mathbb{W}^\varepsilon) < \infty$, this is equivalent to the stochastic control problem

$$\alpha^* \in \operatorname*{argmin}_{\alpha} \Big\{ \mathbb{E}_{\mathbb{P}}\big[\int_0^T \|\alpha_t\|^2 \mathrm{d}t \big] : X_0 \sim \mu_0,\, X_T \sim \mu_T \Big\}, \tag{2}$$

with dynamics $\mathrm{d}X_t = \alpha_t \mathrm{d}t + \sqrt{\varepsilon}\mathrm{d}W_t$. The solution $\mathbb{P}^{SB} = \pi^{SB}_{0,T}\mathbb{W}^\varepsilon_{|0,T}$ is characterized via the optimal entropic coupling $\pi^{SB}_{0,T}(x_0,x_T) = \nu_0(x_0)e^{-\frac{|x_0-x_T|^2}{2\varepsilon T}}e^{\psi(x_T)}$, where the potentials $(\nu_0, \psi = \log h_T)$ satisfy the Schrödinger system:

$$\begin{cases} \mu_T = h_T\nu_T, \quad \nu_T = \nu_0 * \mathcal{N}_{\varepsilon T}, \\ \mu_0 = h_0\nu_0, \quad h_0 = h_T * \mathcal{N}_{\varepsilon T}, \end{cases} \tag{3}$$

and the optimal drift is $\alpha^*(t,x) = \varepsilon\nabla_x \log h_t(x)$ with $h_t = h_T * \mathcal{N}_{\varepsilon(T-t)}$.

2.2 Diffusion Schrödinger Bridge Solvers

Several approaches have been proposed to learn $\mathbb{P}^{SB}$ from samples. The **Sinkhorn** algorithm Léonard (2014); Peyré & Cuturi (2019) iteratively enforces the marginal constraints to recover (ν_0, h_T), but relies on repeated Gaussian convolutions that scale poorly in high dimension and do not yield a parametric representation of the drift De Bortoli et al. (2021). **Iterative Markovian Fitting** (IMF) Shi et al. (2023) alternates between Markovian and reciprocal projections to converge to $\mathbb{P}^{SB}$, but is not simulation-free and can accumulate errors across iterations.

We build on **LightSB-M** Gushchin et al. (2024), which solves the SB via a single optimal projection. For any transport plan $\pi \in \Pi(\mu_0,\mu_T)$ with reciprocal process $\mathbb{P}^\pi = \pi\mathbb{W}^\varepsilon_{|0,T}$,

$$\mathbb{P}^{SB} = \operatorname*{argmin}_{\mathbb{Q}\in\mathcal{S}(\mu_0)} \mathrm{KL}(\mathbb{P}^\pi|\mathbb{Q}),$$

where $\mathcal{S}(\mu_0)$ is the set of SB processes starting from μ_0. The optimal plan in $\mathcal{S}(\mu_0)$ takes the disintegrated form

$$\pi_\varphi(x_0, x_T) = \mu_0(x_0) \underbrace{\frac{e^{\langle x_0, x_T\rangle/\varepsilon}\,\varphi(x_T)}{c_\varphi(x_0)}}_{\pi_\varphi(x_T|x_0)}, \tag{4}$$

and minimizing the resulting denoising score matching loss $\mathbb{E}\big[\|\alpha_\varphi(t, X_t) - \frac{X_T - X_t}{T-t}\|^2\big]$ over a Gaussian-mixture parametrization of φ Korotin et al. (2024) yields an analytic drift with no SDE to simulate at inference.

3 Bridging Schrödinger and Bass

The Schrödinger-Bass (SBB) problem Henry-Labordere et al. (2026) extends the classical SB by jointly optimizing over drift *and* volatility. Given $\mu_0, \mu_T \in \mathcal{P}(\mathbb{R}^d)$, it minimizes over all $\mathbb{P}$ with $X_0 \overset{\mathbb{P}}{\sim} \mu_0$, $X_T \overset{\mathbb{P}}{\sim} \mu_T$:

$$J(\mathbb{P}) = \mathbb{E}_{\mathbb{P}}\Big[\int_0^T \underbrace{\|\alpha_t\|^2 + \beta\|\sigma_t - \sqrt{\varepsilon} I_d\|^2}_{H_\beta(\alpha_t, \sigma_t)} \mathrm{d}t\Big], \tag{5}$$

where $\mathrm{d}X_t = \alpha_t \mathrm{d}t + \sigma_t \mathrm{d}W_t$ and $\beta > 0$. As $\beta \to \infty$ the volatility is frozen at $\sqrt{\varepsilon} I_d$, recovering classical SB; dividing by β and letting $\beta \to 0$ forces the drift to vanish, recovering Bass martingale transport. We emphasize that the $\beta \downarrow 0$ limit is understood only as a formal interpolation: a genuine zero-drift limit requires (μ_0, μ_T) to satisfy convex order. The condition $\beta T > 1$ is required for dual attainment Henry-Labordere et al. (2026, Thm. 3.1(ii-c)).

3.1 Dual Representation and Optimal Controls

The primal $\mathrm{SBB}(\mu_0, \mu_T)$ admits a dual representation maximizing the Lagrangian

$$L_{\mu_0,\mu_T}(\psi, v) = \int_{\mathbb{R}^d} \psi(x)\, \mu_T(\mathrm{d}x) - \int_{\mathbb{R}^d} v(0, x)\, \mu_0(\mathrm{d}x)$$

over pairs (v, ψ), where the potential ψ ranges over the continuous functions of quadratic growth $\mathcal{C}_w := \{\psi \in C^0(\mathbb{R}^d, \mathbb{R}) : |\psi/w|_\infty < \infty\}$ with $w := 1 + |\cdot|^2$, and $v : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ is the value function solving the Bellman equation

$$\partial_t v + H^*_\beta(\nabla_x v, D^2_x v) = 0, \quad v(T, \cdot) = \psi, \tag{6}$$

and $H^*_\beta(p, q) = \frac{1}{2}|p|^2 + \frac{\varepsilon\beta}{2} I_d : \big((I_d - \frac{q}{\beta})^{-1} - I_d\big)$ is the Fenchel-Legendre transform of H_β. When $\beta > \frac{1}{T}$ and μ_0, μ_T have finite second moments, the supremum is attained at (v^*, ψ^*), and the optimizers of (5) are Markovian in the state. Writing them as $\alpha^*_t = \mathrm{a}^*(t, X_t)$ and $\sigma^*_t = \vartheta^*(t, X_t)$ defines the feedback controls $\mathrm{a}^* : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ and $\vartheta^* : [0, T] \times \mathbb{R}^d \to \mathcal{S}^{++}_d$, where $\mathcal{S}^{++}_d$ is the cone of symmetric positive definite $d \times d$ matrices. They are given by

$$\mathrm{a}^*(t, x) = \nabla_x v^*(t, x), \qquad \vartheta^*(t, x) = \sqrt{\varepsilon}\Big(I_d - \tfrac{D^2_x v^*(t,x)}{\beta}\Big)^{-1}. \tag{7}$$

The inverse in (7) is well defined because $\beta > 1/T$ makes $I_d - D^2_x v^*/\beta$ positive definite, which is the same convexity that Proposition A.1 exploits. The full duality statement and the Donsker-Varadhan reduction are given in Appendix A.

3.2 SBB System

The dual problem reduces to maximizing over a potential ψ^* (with $\phi^* = \mathcal{T}^-_\beta[\psi^*] = \log h^*_T$), where $\mathcal{T}^\pm_\beta$ are the quadratic inf/sup-convolution (Moreau) operators, $\mathcal{T}^+_\beta[\phi](x) = \inf_{y \in \mathbb{R}^d}\{\phi(y) + \frac{\beta}{2}|x - y|^2\}$ and $\mathcal{T}^-_\beta[\psi] := -\mathcal{T}^+_\beta[-\psi]$. Before stating the system we introduce the map it acts through. For $t \in [0, T]$, let

$$\Phi_t(y) = \frac{|y|^2}{2} + \frac{\varepsilon}{\beta} \log h^*_t(y), \qquad \mathscr{Y}_t = (\nabla_y \Phi_t)^{-1}, \tag{8}$$

so that Φ_t is convex and $\mathscr{Y}_t : \mathbb{R}^d \to \mathbb{R}^d$ is a well defined increasing transport map, both established in Proposition A.1. We write $S\#\mu$ for the pushforward of $\mu \in \mathcal{P}(\mathbb{R}^d)$ by a measurable map $S : \mathbb{R}^d \to \mathbb{R}^d$, that is $(S\#\mu)(B) = \mu(S^{-1}(B))$ for every Borel set $B \subseteq \mathbb{R}^d$. The optimal potential then satisfies the *SBB system*:

$$\begin{cases} \mathscr{Y}_T\#\mu_T = h_T^*\nu_T, & \nu_T = \nu_0 * \mathcal{N}_{\varepsilon T}, \\ \mathscr{Y}_0\#\mu_0 = h_0^*\nu_0, & h_0^* = h_T^* * \mathcal{N}_{\varepsilon T}, \end{cases} \tag{9}$$

When $\beta \to \infty$, $\mathscr{Y} = I_d$ and one recovers the Schrödinger system; when $\beta \to 0$, h^* is constant and one recovers the Bass system. The optimal controls are:

$$\alpha_t^* = \varepsilon\nabla_y \log h_t^*(\mathscr{Y}_t(X_t)), \qquad \sigma_t^* = \sqrt{\varepsilon}\, D_y^2\Phi_t(\mathscr{Y}_t(X_t)), \quad t \in [0,T]. \tag{10}$$

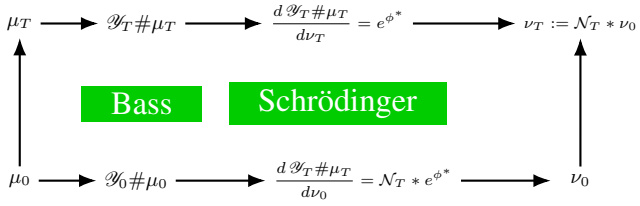
A key structural insight follows from defining $Y_t = \mathscr{Y}_t(X_t)$, i.e.,

$$Y_t = X_t - \tfrac{\varepsilon}{\beta}\nabla_y \log h_t^*(Y_t), \quad t \in [0,T]. \tag{11}$$

Under the change of measure $\frac{\mathrm{d}\mathbb{Q}^*}{\mathrm{d}\mathbb{P}^{\mathrm{SBB}}}|_{\mathcal{F}_t} = \frac{1}{h_t^*(Y_t)}$, the process (Y_t) is a Schrödinger Bridge with constant volatility $\sqrt{\varepsilon}$, while $X_t = Y_t + \frac{\varepsilon}{\beta}\nabla_y \log h_t^*(Y_t)$ is its Bass-stretched version. The joint dynamics under $\mathbb{P}^{\mathrm{SBB}}$ are:

$$\mathrm{d}X_t = D_y^2\Phi_t(Y_t)\,\mathrm{d}Y_t, \qquad \mathrm{d}Y_t = \varepsilon\nabla_y \log h_t^*(Y_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}W_t.$$

In other words, $\mathbb{P}^{\mathrm{SBB}}$ is the *Bass transport of a SB* — a stretched SB. The admissibility and invertibility of $\mathscr{Y}_t$ are established in Appendix A (Proposition A.1), and the SBB system is illustrated below.



4 Generative Modeling with SBB

To generate new samples from μ_T via the learned SBB system, one could directly simulate

$$\mathrm{d}X_t = \varepsilon\nabla_y \log h_t^*(\mathscr{Y}_t(X_t))\,\mathrm{d}t + \sqrt{\varepsilon}\, D_y^2\Phi_t(\mathscr{Y}_t(X_t))\,\mathrm{d}W_t, \; X_0 \sim \mu_0,$$

using an SDE solver (e.g., the Euler-Maruyama scheme) to obtain $X_T \sim \mu_T$. However, this requires computing the inverse of a Hessian matrix, which can be challenging in high dimensions. To overcome this, one can instead generate the process $Y = \mathscr{Y}(X)$ as a DSB:

$$\mathrm{d}Y_t = \varepsilon\nabla_y \log h_t^*(Y_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}W_t,$$

with $Y_0 \sim \mathscr{Y}_0\#\mu_0$ and $Y_T \sim \mathscr{Y}_T\#\mu_T$, and score drift $s_t^*(y) = \varepsilon\nabla_y \log h_t^*(y)$. Then, X_T can be recovered via $X_T = \mathscr{Y}_T^{-1}(Y_T) = Y_T + \frac{1}{\beta}s_T^*(Y_T) \sim \mu_T$.

4.1 Training

As Section 2.2 provides an efficient way to solve the SB between $\mathscr{Y}_0\#\mu_0$ and $\mathscr{Y}_T\#\mu_T$, it remains to learn the transport map $\mathscr{Y}$. Exploiting that $\mathscr{Y}_t = \mathscr{X}_t^{-1}$ with $\mathscr{X}_t(y) = y + \frac{1}{\beta}s_t^*(y)$, we learn the inverse of $\mathscr{X}$ using a neural network $\mathcal{Z}_{\tilde{\theta}}$ to obtain $\mathscr{Y}$, avoiding the need to solve the fixed point (11). Importantly, LightSBB-M does *not* parameterize the SBB volatility σ_t^* directly: the full matrix-valued volatility in X-space is recovered implicitly through the learned map $\mathscr{Y}_t$, while the transformed process Y has constant volatility $\sqrt{\varepsilon}I_d$. This can be viewed as a multivariate, time-dependent Lamperti-type reparameterization: the change of variables converts a state-dependent diffusion in X-space into a constant-volatility process in Y-space, which succeeds because of the convex/gradient

structure of the optimal map (Proposition A.1). This is also computationally advantageous, since it avoids directly learning or evaluating a $d \times d$ volatility field. The numerically delicate regime is when the inverse map becomes poorly conditioned, in particular for very small β. We retain the simpler MLP parameterization; see Appendix C.1 for a discussion of structure-preserving alternatives.

We then introduce LightSBB-M (Algorithm 1) to efficiently solve the SBB problem as follows.

Let $\theta = \{\alpha_j, \mu_j, \Sigma_j\}_{j=1}^J$ denote the parameters of a Gaussian-mixture potential v_θ, and $\tilde{\theta}$ the parameters of the neural network $Z_{\tilde{\theta}}$. Initialize $\phi^0 = 0$, hence $\mathscr{Y}^0 = \mathcal{Z}_{\tilde{\theta}}^0 = I_d$. Then, for $0 \le k \le K$:

1. **Endpoint sampling.** Draw endpoint pairs (y_0, y_T) from a coupling of (p_0^k, p_T^k), where $p_t^k = \mathrm{d}(\mathscr{Y}_t^k \# \mu_t)/\mathrm{d}y$, using
$$\begin{cases} \mathscr{Y}_0^k(x_0) = y_0 = \mathcal{Z}_{\tilde{\theta}}^k(0, x_0), \quad x_0 \sim \mu_0, \\ \mathscr{Y}_T^k(x_T) = y_T = \mathcal{Z}_{\tilde{\theta}}^k(T, x_T), \quad x_T \sim \mu_T. \end{cases}$$
2. **Bridge sampling.** For $t \sim \mathcal{U}[0, T)$ and $Z \sim \mathcal{N}(0, I_d)$, sample $y_t \sim \mathbb{W}^\varepsilon_{|y_0, y_T}$ as an intermediate point from the Brownian bridge:
$$y_t = \frac{T-t}{T} y_0 + \frac{t}{T} y_T + \gamma_t \sqrt{\varepsilon} Z, \quad \gamma_t^2 = \frac{t(T-t)}{T}. \tag{12}$$
3. **Regression step on θ.** Update θ^k by minimizing the bridge-matching loss
$$\mathcal{L}(\theta^k) = \mathbb{E}_{t \sim \mathcal{U}([0,T))} \, \mathbb{E}_{y_T \sim p_T, \, y_t \sim W^\varepsilon_{|y_0, y_T}} \left[\left\| s_\theta^k(t, y_t) - \frac{y_T - y_t}{T - t} \right\|_2^2 \right], \tag{13}$$
which is the KL projection onto the set of SBs, ensuring that the learned drift coincides with the true SB drift.
4. **Regression step on $\tilde{\theta}$.** Once θ^{k+1} is obtained, update $\tilde{\theta}^k$ by minimizing
$$\mathcal{L}(\tilde{\theta}^k) = \mathbb{E}_{x_0 \sim \mu_0, \, x_T \sim \mu_T} \left[\| \mathcal{Z}_{\tilde{\theta}}^k(0, \mathscr{X}_0(x_0)) - x_0 \|^2 + \| \mathcal{Z}_{\tilde{\theta}}^k(T, \mathscr{X}_T(x_T)) - x_T \|^2 \right]. \tag{14}$$

Moreover, using a Gaussian-mixture parametrization of v from LightSB-M, we can derive a closed-form expression for the drift $s_\theta \triangleq s_{v_\theta}$ Gushchin et al. (2024) as follows:

$$s_\theta(t, y) = \varepsilon \nabla_y \log \left(\mathcal{N}(y \,|\, 0, \, \varepsilon(T-t) I_d) \times \sum_{j=1}^J \alpha_j \, \mathcal{N}(r_j \,|\, 0, \, \varepsilon \Sigma_j) \, \mathcal{N}\big(h_j(t, y) \,\big|\, 0, \, A_j^t\big) \right), \tag{15}$$

where $A_j^t \triangleq \frac{t}{\varepsilon(T-t)} I_d + \frac{1}{\varepsilon} \Sigma_j^{-1}$ and $h_j(t, y) \triangleq \frac{y}{\varepsilon(T-t)} + \frac{1}{\varepsilon} \Sigma_j^{-1} r_j$.

We also provide in Appendix B alternative algorithms, including a simplification of Algorithm 1 when β is large and a Sinkhorn-based solver for the SBB problem.

Note that the regression loss (13) rules out arbitrarily small T as the target would explode. On the other hand, an excessively large T drives the noisy marginal μ_T to become almost indistinguishable from μ_0, forcing the reverse dynamics to undo an overwhelming amount of noise; this dramatically inflates the variance of the optimal control and deteriorates sample quality. Hence, in practice we choose T away from extremal values and hence not too small β.

4.2 Inference

Once the drift s_θ^K and the transport map $\mathcal{Z}_{\tilde{\theta}}^K$ are trained, one can generate new samples from μ_T by first computing $Y_0 = \mathcal{Z}_{\tilde{\theta}}^K(X_0) \sim \mathscr{Y}_0 \# \mu_0$, where $X_0 \sim \mu_0$ is an out-of-sample point. Then, sample $Y_T \sim \mathscr{Y}_T \# \mu_T$ according to the learned coupling $\pi_{v_\theta}(Y_T \,|\, Y_0)$ given by (4), and recover $X_T = Y_T + \frac{1}{\beta} s_\theta^K(T, Y_T) \sim \mu_T$. Alternatively, one could simulate the SDE $\mathrm{d}Y_t = s_\theta^K(t, Y_t) \, \mathrm{d}t + \sqrt{\varepsilon} \, \mathrm{d}W_t$, using a numerical SDE solver (e.g., the Euler-Maruyama scheme), but this approach is generally more time-consuming and introduces additional discretization errors.

Note that the drift s_θ defined in (15) is not well-defined at $t = T$. Nevertheless, by continuity of $\phi = \log h$ with respect to time, we can instead approximate $X_{\tilde{T}} = Y_{\tilde{T}} + \frac{1}{\beta} s_\theta^K(\tilde{T}, Y_{\tilde{T}})$, where $\tilde{T} = T - \delta$ for some small $\delta > 0$.

Algorithm 1 LightSBB-M Training Algorithm

Input: Samples $(x_0^m, x_T^m)_{m\leq M} \sim (\mu_0, \mu_T)$, $\theta = \{\alpha_j, \mu_j, \Sigma_j\}_{j\leq J}$, $\tilde{\theta}$, $\beta > 0$, $K > 0$
Initialization: Start with $\mathscr{Y}^0 = I_d$ and $\mathscr{Z}_{\tilde{\theta}}^0 = I_d$
for $k = 0, \cdots, K-1$ **do**
  repeat
    Draw sample batch of pairs $(x_0^n, x_T^n)_{n\leq N}$
    Compute $\mathscr{Y}_0^k(x_0^n) = y_0^n = \mathscr{Z}_{\tilde{\theta}}^k(0, x_0^n)$ and $\mathscr{Y}_T^k(x_T^n) = y_T^n = \mathscr{Z}_{\tilde{\theta}}^k(T, x_T^n)$
    Sample batch $(y_t^n)_{n\leq N} \sim \mathbb{W}^{\varepsilon}_{|y_0,y_1}$ using (12)
    Compute the drift s_θ^k using (15) and update θ^k by minimizing (13)
  until convergence
  $\theta^{k+1} \leftarrow \theta^k$
  repeat
    Draw sample batch of pairs $(x_0^n, x_T^n)_{n\leq N}$
    Compute $\mathscr{X}_0(x_0^n) = x_0^n + \frac{1}{\beta} s_\theta^{k+1}(0, x_0^n)$ and $\mathscr{X}_T(x_T^n) = x_T^n + \frac{1}{\beta} s_\theta^{k+1}(T, x_T^n)$
    Update $\tilde{\theta}^k$ by minimizing (14)
  until convergence
  $\tilde{\theta}^{k+1} \leftarrow \tilde{\theta}^k$
end for
Return θ^K, $\tilde{\theta}^K$

5 Numerical Experiments

In this section, we present numerical experiments to evaluate the proposed algorithm on both univariate and multivariate datasets. We also provide a comparative analysis against state-of-the-art (SOTA) generative models. In all our experiments, we have used $T = 1$.

5.1 Recovering the Optimal Plan and Controls

Most of the SB literature compare the terminal marginal μ_T. This leaves the central claim untested, since the SBB problem is defined by its joint plan $\pi^*(x_0, x_T)$ and by the pair of optimal controls, not by the law it induces at time T. In the Gaussian case, both are known in closed form (Hasenbichler et al., 2026), which gives an exact reference against which the learned solution can be graded.

We take $\mu_0 = \mathcal{N}(0, I_2)$ and $\mu_T = \mathcal{N}(0, \mathrm{diag}(10, 0.1))$ with $T = \varepsilon = 1$, so that one coordinate must spread out while the other is squeezed and both controls matter. Since the model exposes only the endpoints (Y_0, Y_T), we reconstruct the intermediate states by Brownian bridging Y, map them through $X_t = Y_t + \frac{\varepsilon}{\beta} \nabla_y \log h_t^*(Y_t)$, and recover both controls from that map by Itô's formula. Six quantities are then compared against the exact solution. Three assess the coupling, namely the sliced Wasserstein distance between the terminal marginals and between the joint plans, together with the relative error on $\mathrm{Cov}(X_0, X_T)$. The other three grade the solution itself, through the relative gap on the drift component of the SBB objective and the recovery errors E_a and E_b on the optimal drift and volatility. Their definitions and the reconstruction are detailed in Appendix C.3.

Table 1 reports the comparison with LightSB-M over five seeds. Both methods reach the same terminal marginal, which is precisely the point: it does not separate them, whereas the drift does. LightSBB-M halves E_a at $\beta = 10$ and improves it threefold at $\beta = 100$, with seed dispersion an order of magnitude below the gap in both cases; the drift gap is likewise clearly separated at $\beta = 10$, while at $\beta = 100$ both methods are within seed noise of the optimum. On E_b, LightSB-M is structurally pinned to a constant volatility and cannot go below 0.021, whereas LightSBB-M reaches a value compatible with zero. The aggregate cross-covariance separates the two at neither β. The coupling confirms this: the exact SB assigns a covariance of 0.092 to the squeezed coordinate against 0.140 for the SBB optimum at $\beta = 10$, and LightSBB-M attains 0.102, clearing the SB value across every seed.

Table 1: Recovery of the Gaussian SBB solution, mean ± standard deviation over five seeds. Lower is better throughout; the best value of each β block is in **bold**. The first three rows compare marginals and plans, the last three the objective and the optimal controls. The entry $< 10^{-5}$ is below numerical resolution rather than a measured value.

	$\beta = 10$		$\beta = 100$	
Metric	LightSB-M	**SBB (ours)**	LightSB-M	**SBB (ours)**
terminal SW_2	$\mathbf{0.038 \pm 0.008}$	$\mathbf{0.038 \pm 0.010}$	0.038 ± 0.008	$\mathbf{0.026 \pm 0.005}$
plan SW_2	0.032 ± 0.005	$\mathbf{0.026 \pm 0.005}$	0.026 ± 0.005	$\mathbf{0.019 \pm 0.003}$
cross-cov.	$\mathbf{0.016 \pm 0.001}$	0.017 ± 0.006	0.008 ± 0.004	$\mathbf{0.007 \pm 0.004}$
drift gap	0.200 ± 0.010	$\mathbf{0.050 \pm 0.006}$	0.026 ± 0.009	$\mathbf{0.0009 \pm 0.0050}$
E_a	0.021 ± 0.001	$\mathbf{0.010 \pm 0.001}$	0.0016 ± 0.0002	$\mathbf{0.00047 \pm 0.00007}$
E_b	0.021 ± 0.000	$\mathbf{0.0014 \pm 0.0021}$	0.00041 ± 0.00000	$\mathbf{< 10^{-5}}$

Table 2: 2-Wasserstein distances ($\mathcal{W}_2$) on synthetic datasets (lower is better). The best results are highlighted in **bold**. *Indicates results taken from Tong et al. (2024b).

	$\mathcal{W}_2$ (↓)		
Algorithm	$\mathcal{N} \to$ 8gaussians	moons $\to$ 8gaussians	$\mathcal{N} \to$ moons
$[\text{SF}]^2$M-Exact*	0.275±0.058	0.726±0.137	0.124±0.023
$[\text{SF}]^2$M-I*	0.393±0.054	1.482±0.151	0.185±0.028
DSBM-IPF*	0.315±0.079	0.812±0.092	0.140±0.006
DSBM-IMF*	0.338±0.091	0.838±0.098	0.144±0.024
DSB*	0.411±0.084	0.987±0.324	0.190±0.049
LightSB-M	0.339±0.099	0.295±0.051	0.201±0.042
SBB (ours)	**0.241±0.083**	**0.201±0.034**	**0.109±0.014**
OT-CFM*	0.303±0.043	0.601±0.027	0.130±0.016
SB-CFM*	2.314±2.112	0.843±0.079	0.434±0.594
RF*	0.421±0.071	1.525±0.330	0.283±0.045
I-CFM*	0.373±0.103	1.557±0.407	0.178±0.014
FM*	0.343±0.058	—	0.209±0.055

5.2 Quantitative Evaluation on Low-Dimensional Datasets

We provide a quantitative evaluation of the proposed SBB method on low-dimensional datasets, namely **8gaussians** and **moons** ($d = 2$), and compare it against several SOTA baselines. These include alternative SB solvers Gushchin et al. (2024); Shi et al. (2023); Tong et al. (2024b); De Bortoli et al. (2021) and flow-based generative approaches (ODE) Tong et al. (2024a); Lipman et al. (2023); Liu (2022). Each model transports $10{,}000$ samples from the source to the target distribution, and performance is assessed using the 2-Wasserstein distance between real and generated samples, averaged over five seeds. As shown in Table 2, the proposed SBB framework consistently achieves the lowest $\mathcal{W}_2$ distances across all benchmark tasks. It outperforms both SB-based methods (e.g., LightSB-M) and diffusion/flow-based baselines, yielding on average a $\sim 19\%$ improvement in transport accuracy while reducing the variance of the results, including on multimodal targets such as the *moons* $\to$ 8-*gaussians* task. Note that FM is incompatible with the latter as it requires a Gaussian source distribution. We further analyze the influence of β and the convergence behavior of the outer loop in Appendix C.5, including practical guidelines for selecting β.

6 Evaluation on Unpaired Image-to-Image Translation

To evaluate the generative capabilities of our model we consider the task of unpaired image-to-image translation Zhu et al. (2017) on subsets of the FFHQ dataset at a resolution of 1024×1024 pixels Karras et al. (2019). The source distribution μ_0 comprises adult faces, while the target distribution μ_T consists of child faces. Since the two domains are not paired, we adopt a cycle-consistent framework in the latent space of an Adversarial Latent AutoEncoder (ALAE) Pidhorskyi et al. (2020). Each image is first encoded by the *pretrained* ALAE encoder $E : \mathbb{R}^{3\times1024\times1024} \to \mathbb{R}^{512}$, we train our

Table 3: Quantitative evaluation of adult → child translation at $\beta = 1$, $\varepsilon = 0.1$. Age estimated by InsightFace, identity similarity by cosine similarity, FID against 1000 decoded held-out child faces, averaged over five random seeds.

Metric	LightSBB-M	LightSB-M	OT-CFM	NOT	$[\text{SF}]^2\text{M}$
Avg. age ↓	$\mathbf{12.5 \pm 8.3}$	27.2 ± 13.0	24.9 ± 14.8	23.4 ± 14.7	24.4 ± 14.5
Age ≤ 18 (%) ↑	$\mathbf{76.0}$	19.3	30.4	34.1	30.1
Identity sim. ↑	0.81 ± 0.05	0.85 ± 0.05	0.89 ± 0.05	$\mathbf{0.90 \pm 0.05}$	0.89 ± 0.05
FID ↓	$\mathbf{20.9}$	21.6	24.5	24.7	24.4

SBB framework in that 512-dimensional latent space, and the final latent output is mapped back through the pretrained decoder D. This setup assesses the fidelity of the generated child faces and the preservation of identity-related attributes without requiring paired supervision.

Qualitative comparison. Figure 1 compares LightSBB at $\beta = 1$ and $\varepsilon = 0.1$ against LightSB-M, the current state of the art on this task, and the three baselines described below. The outputs of LightSBB are visibly more child-like than those of the other methods, at a comparable perceptual quality. This figure is illustrative, and we defer the comparison between the methods to the quantitative evaluation below. The effect of β and ε is studied separately in Figure 8 of Appendix C.6.

Quantitative evaluation. To complement the qualitative comparison, we evaluate age translation quality and identity preservation quantitatively. Age is estimated using an InsightFace pre-trained model, identity similarity is measured by cosine similarity between input and output latent codes, and perceptual quality by the *Fréchet Inception Distance* (FID) Heusel et al. (2017) against held-out real child faces passed through the same ALAE decoder. Beyond LightSB-M, we compare against three baselines chosen to be of *different natures*: OT-CFM Tong et al. (2024a), NOT Korotin et al. (2023) and $[\text{SF}]^2\text{M}$ Tong et al. (2024b). Together with the bridge-based methods, this covers stochastic bridges, deterministic flows, adversarial maps and neural SDEs approaches. We additionally note that LightSB-M already provides an extensive comparison against further baselines such as DSBM Shi et al. (2023) in its original paper Gushchin et al. (2024). Results are reported in Table 3.

LightSBB-M substantially improves age translation quality. The fraction of outputs estimated as under 18 years old reaches 76.0%, roughly quadruple the 19.3% of LightSB-M and more than twice the best baseline, with a consistently lower average estimated age. The modest decrease in identity similarity is the expected trade-off, since the competing methods retain more adult facial structure and therefore stay closer to the input embedding, while LightSBB-M produces genuinely more child-like faces. A method that perfectly preserves identity would by definition fail at the translation task, and the baselines illustrate exactly that: OT-CFM, NOT and $[\text{SF}]^2\text{M}$ all score higher on identity than either bridge method while transporting far fewer faces across the age boundary. Perceptual quality is unaffected, LightSBB-M attaining the best FID of the five methods.

Finally, Figure 2 depicts the complete pipeline from the source distribution $X_0 \sim \mu_0$ to the target distribution $X_1 \sim \mu_1$ together with the underlying Y-process defined by our framework. The intermediate states Y_t are obtained via bridge matching Equation (12) between the endpoints Y_0 and Y_T. Note that this bridging occurs on the Y process, not on X, since Y is a SB.



Figure 2: Trajectory from $X_0 \sim \mu_0$ to $X_1 \sim \mu_1$ with the underlying Y process.

7 Conclusion

Potential impact. We derive an efficient procedure to compute the optimal SBB transport plan $\mathbb{P}^{SBB}$, and verify against a closed-form Gaussian solution that the algorithm recovers that plan and both

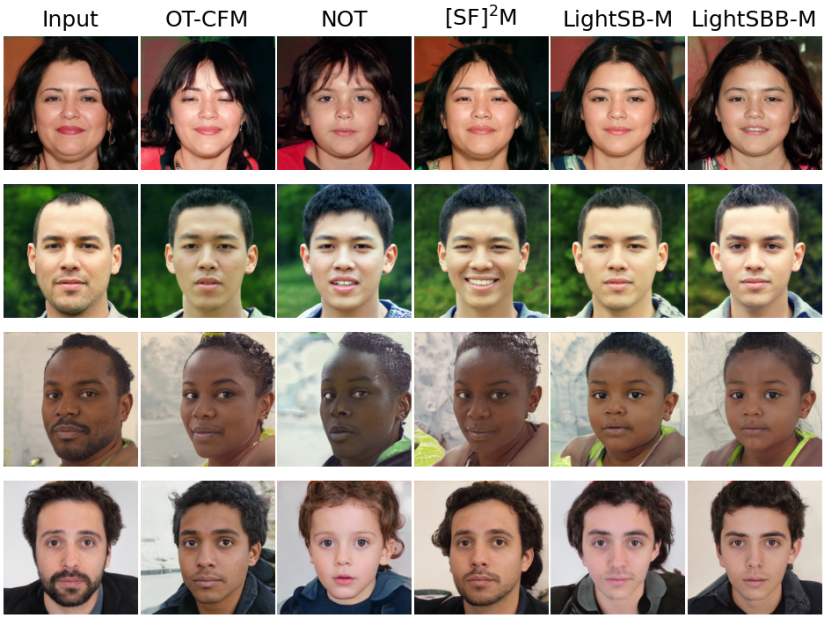


Figure 1: Adult → child translation at $\beta = 1$, $\varepsilon = 0.1$, on four held-out faces. The leftmost column shows the input, and every method is evaluated on the same inputs with a shared decoder.

optimal controls, where a Schrödinger Bridge solver reaching the same terminal marginal does not. A tunable parameter β balances drift and volatility, and the resulting stochastic volatility gives the method a richer class of dynamics than the standard SB.

Limitations and future work. The exact validation is confined to the Gaussian case, the only one admitting a closed-form reference. Computing $\mathbb{P}^{SBB}$ also requires iterating over the transport map, which is computationally demanding, and while the plan recovered at $K = 5$ is the optimal one (in the Gaussian case), the convergence of the outer loop still deserves a proof. Time-series data, already addressed by the SB Hamdouche et al. (2023); Alouadi et al. (2025), is a natural extension.

Use of Generative AI

We used generative AI tools for two purposes. They assisted in drafting and rephrasing parts of the manuscript, and they were used to implement the code for some of the experiments, namely the baselines of Section 6 and the ground-truth validation of Section 5.1. The theoretical results, including all mathematical claims and proofs, were developed without AI assistance. We have reviewed all AI-assisted work and take responsibility for the final content of this paper.

References

Alexandre Alouadi, Baptiste Barreau, Laurent Carlier, and Huyên Pham. Robust time series generation via schrödinger bridge: a comprehensive evaluation. In *International Conference on AI in Finance*, 2025.

Julio Backhoff, Gregoire Loeper, and Jan Obloj. Geometric martingale benamou–brenier transport and geometric bass martingales. *Proceedings of the American Mathematical Society*, 2025.

Richard F Bass. Skorokhod imbedding via stochastic integrals. *Séminaire de Probabilités*, 1983.

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

Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schrödinger bridge matching. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Mohamed Hamdouche, Pierre Henry-Labordere, and Huyên Pham. Generative modeling for time series via schrödinger bridge, 2023.

Manuel Hasenbichler, Gudmund Pammer, and Stefan Thonhauser. A weak transport approach to the schrödinger-bass bridge, 2026. URL https://arxiv.org/abs/2604.02312.

Pierre Henry-Labordere, Grégoire Loeper, Othmane Mazhar, Huyên Pham, and Nizar Touzi. Bridging schrödinger and bass: A semimartingale optimal transport problem with diffusion control, 2026.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Mengjian Hua, Mathieu Laurière, and Eric Vanden-Eijnden. An efficient on-policy deep learning framework for stochastic optimal control. In *ICLR 2025 Workshop on Frontiers in Probabilistic Inference (FPI)*, 2025.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019.

Yeongjong Kim, Namkyeong Cho, Minseok Kim, and Yeoneung Kim. Physics-informed approach for exploratory Hamilton–Jacobi–Bellman equations via policy iterations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22609–22616, 2026.

Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. In *International Conference on Learning Representations*, 2023.

Alexander Korotin, Nikita Gushchin, and Evgeny Burnaev. Light schrödinger bridge. In *International Conference on Learning Representations*, 2024.

Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems-A*, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Jin Ma, Ying Tan, and Renyuan Xu. Schrödinger bridge for generative AI: Soft-constrained formulation and convergence analysis. *arXiv preprint arXiv:2510.11829*, 2025.

Ashok Vardhan Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason D. Lee. Optimal transport mapping via input convex neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6672–6681, 2020.

Kushagra Pandey, Jaideep Pathak, Yilun Xu, Stephan Mandt, Michael Pritchard, Arash Vahdat, and Morteza Mardani. Heavy-tailed diffusion models. In *International Conference on Learning Representations*, 2025.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 2019.

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14104–14113, 2020.

E. Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'Institut Henri Poincaré*, 1932.

Dario Shariatian, Umut Simsekli, and Alain Oliviero Durmus. Denoising levy probabilistic models. In *International Conference on Learning Representations*, 2025.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. In *Advances in Neural Information Processing Systems*, 2023.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019.

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

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a.

Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024b.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

A Dual Representation of the SBB Problem

This section provides a self-contained and detailed account of the theoretical results from Henry-Labordere et al. (2026) that underpin our algorithm. We refer the reader to that work for full proofs and reproduce the key statements here for completeness. The exception is Proposition A.1 and its proof which are original contributions of the present paper.

A.1 Full Duality Statement

The primal problem $\mathrm{SBB}(\mu_0, \mu_T)$ admits a dual representation, which consists of maximizing over a suitable class of functions (v, ψ) the Lagrangian functional

$$L_{\mu_0,\mu_T}(\psi, v) = \int_{\mathbb{R}^d} \psi(x)\, \mu_T(\mathrm{d}x) - \int_{\mathbb{R}^d} v(0,x)\, \mu_0(\mathrm{d}x),$$

where v is the value function of the unconstrained stochastic control problem with Bellman equation:

$$\begin{cases} \partial_t v + H^*_\beta(\nabla_x v, D^2_x v) = 0, & \text{on } [0,T) \times \mathbb{R}^d, \\ v(T,\cdot) = \psi, & \text{on } \mathbb{R}^d, \end{cases}$$

with H^*_β denoting the Fenchel-Legendre transform of H_β, explicitly given by

$$H^*_\beta(p,q) = \frac{1}{2}|p|^2 + \frac{\varepsilon\beta}{2} I_d : \Big((I_d - \frac{q}{\beta})^{-1} - I_d \Big),$$

for $(p,q) \in \mathbb{R}^d \times \mathbb{S}^d_+$ such that $q < \beta I_d$. Assuming $\mathrm{SBB}(\mu_0, \mu_T) < \infty$, we have:

- **Attainment of the primal problem**: there exists $(\alpha^*, \sigma^*) \leftrightarrow \mathbb{P}^{\mathrm{SBB}}$ attaining the infimum in $\mathrm{SBB}(\mu_0, \mu_T)$, in feedback form: $\alpha^*_t = \mathrm{a}^*(t, X_t)$, $\sigma^*_t = \vartheta^*(t, X_t)$.
- **Duality relation**: we have

$$\mathrm{SBB}(\mu_0, \mu_T) = \sup_{\substack{\psi \in C^2 \cap C^\infty_b \cap L^1(\mu_T),\ v \in C^{1,2}_b, \\ D^2_x v < \beta I_d,\ \psi = v(T,\cdot), \\ \partial_t v + H^*_\beta(\nabla_x v, D^2_x v) = 0}} L_{\mu_0,\mu_T}(\psi, v).$$

- **Duality on the control**: When μ_0 and μ_T have finite second moment and $\beta > \frac{1}{T}$, the supremum is attained at (v^*, ψ^*) with optimal feedback policies

$$\begin{cases} \mathrm{a}^*(t,x) = \nabla_x v^*(t,x), & (t,x) \in [0,T) \times \mathbb{R}^d, \\ \vartheta^*(t,x) = \sqrt{\varepsilon} \Big(I_d - \frac{D^2_x v^*(t,x)}{\beta} \Big)^{-1}. \end{cases}$$

A.2 Donsker-Varadhan Reduction and the $\mathcal{T}^{\pm}_\beta$ Operators

One can exploit the quadratic form of the SBB criterion to reduce the dual problem to the maximization over the potential $\psi \in C^2 \cap C^\infty_b \cap L^1(\mu_T)$, subject to $D^2_x \psi < \beta I_d$, of the Donsker-Varadhan type functional:

$$\int \psi \mathrm{d}\mu_T - \int \mathcal{T}^+_\beta \Big[\varepsilon \log \int e^{\mathcal{T}^-_\beta[\psi](\cdot + z)} \mathcal{N}_{\varepsilon T}(\mathrm{d}z) \Big](x)\, \mu_0(\mathrm{d}x),$$

where the quadratic inf/sup-convolution operators are defined as

$$\begin{cases} \mathcal{T}^+_\beta[\phi](x) := \inf_{y \in \mathbb{R}^d} \Big[\phi(y) + \frac{\beta}{2}|x-y|^2 \Big], & x \in \mathbb{R}^d, \\ \mathcal{T}^-_\beta[\psi](y) := \sup_{x \in \mathbb{R}^d} \Big[\psi(x) - \frac{\beta}{2}|x-y|^2 \Big], & y \in \mathbb{R}^d. \end{cases}$$

A.3 Envelope Projection and Admissible Map

The Hessian constraint $D_x^2 v \prec \beta I_d$ in the dual problem is not merely a regularity condition: it underpins both the invertibility of $\mathscr{Y}_t$ and the well-posedness of the volatility formula. The following proposition shows that this constraint can always be enforced without loss of generality, by replacing any candidate potential with its β-convex envelope.

Throughout this section we use two objects from the reduced dual formulation of Henry-Labordere et al. (2026, Sec. 2.4), which we recall for completeness. First, the *backward heat potential* indexed by heat time s,

$$u_s^\phi(y) := \log\big(\mathcal{N}_s * e^\phi\big)(y), \qquad (s,y) \in \overline{Q_T}, \qquad \mathcal{N}_0 * e^\phi := e^\phi, \tag{16}$$

which is the same object as the potential h^* of Section 3.2 read in reverse time: $u_{T-t}^\phi = \log h_t^\phi$ with $h_t^\phi = \mathcal{N}_{T-t} * e^\phi$, so that $\nabla u_{T-t}^\phi = \nabla \log h_t^\phi$ is the score appearing in the optimal controls (7). Second, the *reduced dual functional*

$$J_{\mathrm{red}}(\phi) := \mu_T\big(\mathcal{T}_\beta^+[\phi]\big) - \mu_0\big(\mathcal{T}_\beta^+[u_T^\phi]\big), \tag{17}$$

whose supremum over $\phi \in C_w^{\mathrm{conv}}$ equals $\mathrm{SBB}(\mu_0,\mu_T)$. We write J_{red} rather than J to avoid any confusion with the primal objective $J(\mathbb{P})$ of (5). The statement below is written for $\varepsilon = 1$, as in Henry-Labordere et al. (2026); the general case follows by the time–space rescaling that maps $\mathbb{W}^\varepsilon$ to $\mathbb{W}^1$.

Proposition A.1 (Envelope projection and admissible map)**.** *Assume $\beta T > 1$. For any $\phi \in C_w := \{\psi \in C^0(\mathbb{R}^d,\mathbb{R}) : |\psi/w|_\infty < \infty\}$, define its β-convex envelope $\tilde\phi := \mathcal{T}_\beta^- \circ \mathcal{T}_\beta^+[\phi]$. Then:*

1. *$\tilde\phi \in C_w^{\mathrm{conv}}$ and $J_{\mathrm{red}}(\tilde\phi) \geq J_{\mathrm{red}}(\phi)$;*
2. *the associated exact value function $v_t^{\tilde\phi}(x) := \mathcal{T}_\beta^+[u_{T-t}^{\tilde\phi}](x)$ satisfies $D_x^2 v_t^{\tilde\phi}(x) \prec \beta I_d$ for $t < T$;*
3. *the induced map $\mathscr{Y}_t^{\tilde\phi}(x) := x - \frac{1}{\beta} D v_t^{\tilde\phi}(x)$ is well-defined and one-to-one on $\mathbb{R}^d$ for every $t < T$, with $D\mathscr{Y}_t^{\tilde\phi}(x) \succ 0$. It coincides with the map $\mathscr{Y}_t = (\nabla_y \Phi_t)^{-1}$ of (8), since $\nabla_y \Phi_t(y) = y + \frac{1}{\beta}\nabla_y \log h_t^{\tilde\phi}(y)$;*
4. *if the envelope is taken with parameter $\beta - \delta$ for some $\delta \in (0,\beta)$, the associated exact volatility obeys the uniform bound $|\vartheta^*(t,x)|_{\mathrm{op}} \leq \sqrt{\varepsilon}\,\beta/\delta$.*

Proof. 1. Follows directly from Henry-Labordere et al. (2026, Remark 2.3): for every $\phi \in C_w$ one has $\tilde\phi \in C_w^{\mathrm{conv}} := \mathcal{C}_w \cap \mathcal{C}^{\mathrm{conv}}$, where $\mathcal{C}^{\mathrm{conv}} := \{\phi \in C^0(\mathbb{R}^d) : \phi \text{ is } \beta\text{-convex}\}$, together with $\tilde\phi \leq \phi$ and $\mathcal{T}_\beta^+[\tilde\phi] = \mathcal{T}_\beta^+[\phi]$; hence $u_T^{\tilde\phi} \leq u_T^\phi$ and $J_{\mathrm{red}}(\tilde\phi) \geq J_{\mathrm{red}}(\phi)$.

2. Let $v_t^{\tilde\phi}(x) := \mathcal{T}_\beta^+[u_{T-t}^{\tilde\phi}](x)$. Since $\tilde\phi \in C_w^{\mathrm{conv}}$, Henry-Labordere et al. (2026, Lemma 5.1(3)) yields $v^{\tilde\phi} \in C^{1,2}(Q_T)$, uniqueness of the minimiser in the Moreau envelope, and

$$D^2 v_t^{\tilde\phi}(x) - \beta I_d = -\beta^2\Big(\beta I_d + D^2 u_{T-t}^{\tilde\phi}\big(\mathscr{Y}_t^{\tilde\phi}(x)\big)\Big)^{-1}.$$

The same lemma gives the lower Hessian bound $D^2 u_{T-t}^{\tilde\phi}(y) + \kappa(t) I_d \succeq 0$ with $\kappa(t) = \frac{\beta}{1+\beta(T-t)}$, so that $\beta I_d + D^2 u_{T-t}^{\tilde\phi}(y) \succeq (\beta - \kappa(t)) I_d \succ 0$ for $t < T$. The right-hand side above is therefore negative definite, giving $D^2 v_t^{\tilde\phi}(x) \prec \beta I_d$.

3. Well-posedness and injectivity follow from Henry-Labordere et al. (2026, Theorem 3.1(ii-c)) under $\beta T > 1$. The first-order condition for the Moreau envelope gives $D\mathscr{Y}_t^{\tilde\phi}(x) = \beta\big(\beta I_d + D^2 u_{T-t}^{\tilde\phi}(\mathscr{Y}_t^{\tilde\phi}(x))\big)^{-1}$, which is positive definite by the bound established in item 2. The identification with $(\nabla_y \Phi_t)^{-1}$ follows from $\nabla v_t^{\tilde\phi}(x) = \nabla u_{T-t}^{\tilde\phi}(\mathscr{Y}_t^{\tilde\phi}(x))$ (Henry-Labordere et al., 2026, Lemma 5.1(3-a)), so that $x = \mathscr{Y}_t^{\tilde\phi}(x) + \frac{1}{\beta}\nabla \log h_t^{\tilde\phi}(\mathscr{Y}_t^{\tilde\phi}(x)) = \nabla_y \Phi_t(\mathscr{Y}_t^{\tilde\phi}(x))$.

4. Applying item 2 with $\beta-\delta$ in place of β gives $D^2 v_t \preceq (\beta-\delta) I_d$, hence $I_d - \frac{1}{\beta} D^2 v_t \succeq \frac{\delta}{\beta} I_d$. The exact control formula $\vartheta^*(t,x) = \sqrt{\varepsilon}\,(I_d - \frac{1}{\beta} D^2 v_t(x))^{-1}$ then yields $|\vartheta^*(t,x)|_{\mathrm{op}} \leq \sqrt{\varepsilon}\,\beta/\delta$.

□

B Exploring Alternative Algorithms

In this section, we explore other potential algorithms to solve the SBB problem.

B.1 LightSBB-M for β Large

We recall that the objective of Algorithm 1 is to estimate both the drift of the process Y and the transport map $\mathscr{Y}_t(X_t) = X_t - \frac{1}{\beta}\varepsilon \nabla_y \log h_t^*(\mathscr{Y}_t(X_t))$ for $t \in [0,T]$. One may notice that for large β, we can make the following approximation of the transport map:

$$\mathscr{Y}_t(X_t) = X_t - \frac{1}{\beta}\varepsilon \nabla_y \log h_t^*(\mathscr{Y}_t(X_t)) \simeq X_t - \frac{1}{\beta}\varepsilon \nabla_y \log h_t^*(X_t) \tag{18}$$

Approximation 18 hence provides an explicit expression, given the drift, of the transport map and avoids computing the inverse of $\mathscr{X}$. One can simply sample from $\mathscr{Y}_t \# \mu_t$, at iteration k, using

$$\begin{cases} \mathscr{Y}_0^k(x_0) = y_0 = x_0 - \frac{1}{\beta} s_\theta^k(0, x_0), \\ \mathscr{Y}_T^k(x_T) = y_T = x_T - \frac{1}{\beta} s_\theta^k(T, x_T) \end{cases} \tag{19}$$

Then, Algorithm 1 simply becomes as described in Algorithm 2.

Algorithm 2 LightSBB-M for β large

Input: Samples $(x_0^m, x_T^m)_{m \leq M} \sim (\mu_0, \mu_T)$, $\theta = \{\alpha_j, \mu_j, \Sigma_j\}_{j \leq J}$, $\beta > 0$ large, $K > 0$
Initialization: Start with $\mathscr{Y}^0 = I_d$
for $k = 0, \cdots, K-1$ **do**
  repeat
    Draw sample batch of pairs $(x_0^n, x_T^n)_{n \leq N}$
    Compute $\mathscr{Y}_0^k(x_0^n) = y_0^n$ and $\mathscr{Y}_T^k(x_T^n) = y_T^n$ using (19)
    Sample batch $(y_t^n)_{n \leq N} \sim \mathbb{W}_{|y_0, y_1}$ using (12)
    Compute the drift s_θ^k using (15) and update θ^k by minimizing (13)
  until convergence
  $\theta^{k+1} \leftarrow \theta^k$
end for
Return θ^K

Once s_θ^K is trained, one can sample similarly as before. First, compute $Y_0 = X_0 - \frac{1}{\beta} s_\theta^K(0, X_0) \sim \mathscr{Y}_0 \# \mu_0$ where $X_0 \sim \mu_0$ is an out-of-sample point. Then, sample $Y_T \sim \mathscr{Y}_T \# \mu_T$ according to the learned coupling $\pi_{v_\theta}(Y_T \,|\, Y_0)$ given by (4), and recover $X_T = Y_T + \frac{1}{\beta}\, s_\theta^K(T, Y_T) \sim \mu_T$.

This approximation eliminates the need to learn an explicit transport map, which reduces computational cost and mitigates the accumulation of approximation errors during training.

B.2 Sinkhorn-based Algorithm

We introduce a Sinkhorn-based algorithm that iteratively updates the potential function ϕ, setting $\varepsilon = 1$ for simplicity of notation. The algorithm proceeds as follows, starting from $\phi^0 = 0$, for all $0 \leq k \leq K$:

1. Compute $h_0^k(y) = \mathbb{E}_{\mathcal{N}(0,T)}(e^{\phi^k(y+Z)}) \simeq \frac{1}{N}\sum_{i=1}^N e^{\phi^k(y+Z_i)}$ with $(Z_i)_{i \leq N} \sim \mathcal{N}(0,T)$ i.i.d.

2. Compute $\mathscr{Y}_0^k(x) = \arg\min_{y\in\mathbb{R}^d}\left[\log h_0^k(y) + \frac{\beta}{2}\|x-y\|^2\right]$ for $x \sim \mu_0$. This can be achieved either via grid search, or approximately (for large β) using $\mathscr{Y}_0^k(x) \simeq x - \frac{1}{\beta}\nabla_y \log h_0^k(x)$, where $\nabla_y \log h_0^k(x)$ is obtained by interpolating $\log h_0^k$ and estimating its gradient via finite differences for all $x \sim \mu_0$.
3. Compute $\mathscr{Y}_T^k(x) = \arg\min_{y\in\mathbb{R}^d}\left[\phi^k(y) + \frac{\beta}{2}\|x-y\|^2\right]$ for $x \sim \mu_T$. Similarly, this can be done either through grid search or, when β is large, by approximating $\mathscr{Y}_T^k(x) \simeq x - \frac{1}{\beta}\nabla_y \phi^k(x)$, where the gradient is computed from an interpolated version of ϕ^k using finite differences.
4. Compute $h_T^{k+1}(y) = e^{\phi^{k+1}(y)} = \dfrac{\mathbb{E}_{\mu_T}\left(\delta_{\mathscr{Y}_T^k(X_T)}(y)\right)}{\mathbb{E}_{\mu_0}\left(\frac{\mathcal{N}_{\mathscr{Y}_0^k(X_0),T}(y)}{h_0^k(\mathscr{Y}_0^k(X_0))}\right)}$ via Monte Carlo estimation using samples from μ_T and μ_0.
5. Update $\phi^{k+1} = \log h_T^{k+1}$.

After convergence, $\phi^\star \simeq \phi^K$, one can simulate the diffusion $\mathrm{d}Y_t = \nabla_y \log h_t^K(Y_t)\mathrm{d}t + \mathrm{d}W_t$ starting from $Y_0 \sim \mathscr{Y}_0^K \# \mu_0$ obtained in step 2, with $h_t^K = h_T^K * \mathcal{N}_{T-t}$ computed on a grid and interpolated to estimate its score via finite differences. Note that this approach is computationally demanding and does not scale well to high-dimensional settings due to: (i) the need for interpolation, (ii) the grid search procedure, and (iii) the estimation of the density of $\mathscr{Y}_T^k \# \mu_T$ in the numerator of step 4, which can be performed using kernel regression methods.

Alternatively, one may directly simulate the process X as follows:

- **Compute** $h_t^K(\cdot)$ as in step 4, using $\mathscr{Y}_T^K(\cdot)$ and $\mathscr{Y}_0^K(\cdot)$.
- **Compute** $\mathscr{Y}_t^K(\cdot)$ as in step 3.
- **Compute the optimal drift and volatility controls.** The expressions of the optimal drift α^K and volatility σ^K at time t are given by:

$$\begin{cases} \alpha^K(t,x) = \nabla_y \log h_t^K(\mathscr{Y}_t^K(x)) = \frac{\nabla_y {h_t}^K(\mathscr{Y}_t^K(x))}{h_t^K(\mathscr{Y}_t^K(x))}, \\ \sigma^K(t,x) = I_d + \frac{1}{\beta}\nabla_y^2 \log h_t^K(\mathscr{Y}_t^K(x)) = I_d + \frac{1}{\beta}\left(\frac{\nabla_y^2 h_t^K(\mathscr{Y}_t^K(x))}{h_t^K(\mathscr{Y}_t^K(x))} - \frac{\nabla_y h_t^K(\mathscr{Y}_t^K(x))\nabla_y h_t^K(\mathscr{Y}_t^K(x))^T}{h_t^K(\mathscr{Y}_t^K(x))^2}\right) \end{cases}$$

  where the derivatives are obtained from the interpolated version of $h_t^K(\cdot)$ using finite differences.

We propose to evaluate this approach by transporting between measures μ_0 and μ_T for which the optimal drift and volatility are known. As an illustrative example, consider $\mu_0 = \delta_0$ and $\mu_T = \mathcal{N}(0,T)$. In this case, the optimal drift and volatility are given by $\alpha_0^* = 0$ and $\sigma_0^* = I_d$, since $X_T = \alpha_0^* T + \sigma_0^* W_T = 0 + I_d \times \mathcal{N}(0,T) \sim \mathcal{N}(0,T)$. We perform the experiment using 2000 real samples drawn from $\mathcal{N}(0,T)$ with parameters $T = 1$, $\beta = 1$, $K = 20$, $N^\pi = 1$, $N = 1$, and initialization $\phi^0(y) = \frac{y^2}{4}$. The estimated values are $\hat{\alpha}_0 = 0.002$ and $\hat{\sigma}_0 = 1.001$. Figure 3 shows the estimated drift and volatility obtained with SBB across all time steps $t \in [0,T]$, along with the empirical cumulative distribution function (ECDF) computed from both real and generated samples.

We also present in Figure 4 illustrative toy examples in one dimension, demonstrating the behavior of the alternative Sinkhorn algorithm for transporting between different probability measures. In these experiments, we set $\beta = 10$, $K = 20$, and $N^\pi = 40$, corresponding to the number of time steps in the Euler scheme. For the density estimation in step 4, we employ the following kernel function:

$$K_\lambda(x) = \frac{1}{\lambda^d}\left(1 - \left\|\tfrac{x}{\lambda}\right\|^2\right)^2 \mathbb{1}_{\{\|x\|<\lambda\}}, \quad x \in \mathbb{R}^d,$$

where $\lambda = 0.3$ denotes the kernel bandwidth.

C Experimental Setup

All experiments were conducted using a single NVIDIA A100 SXM4 40 GB GPU. The parameters used throughout this study, unless otherwise stated, are summarized in Table 4.

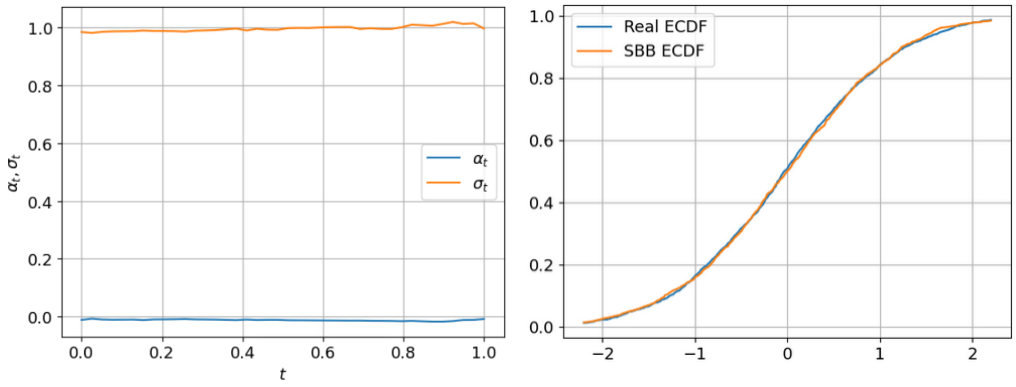


Figure 3: Estimated optimal drift/volatility over time (*left*) and Empirical CDF comparison between real and SBB-generated samples (*right*)

Table 4: Table of parameters and their values.

K	T	$\tilde{T}$	n_{epoch}	Batch Size	lr
5	1	0.99	15000	512	10^{-3}

C.1 Implementation Remark: MLP vs. Structure-Preserving Architectures

By Proposition A.1, one may enforce the admissible-map structure at the dual level by replacing any candidate potential by its β-convex envelope. In additional experiments, we tested structure-preserving neural architectures enforcing this monotonicity structure more directly. They did not yield a meaningful improvement in transport accuracy while increasing training time, so we retain the simpler MLP parameterization. This is consistent with recent stochastic-control and bridge-based numerical practice, where exact structural conditions are often handled at the PDE/control level while neural approximations remain unconstrained or softly constrained for scalability Kim et al. (2026); Hua et al. (2025); Ma et al. (2025); by contrast, some static OT methods enforce convexity explicitly through ICNN architectures when Brenier structure is the primary object Makkuva et al. (2020).

C.2 Runtime and Efficiency

We now present in Table 5 a comprehensive analysis of both training and inference costs for our algorithm, compared to the LightSB-M baseline. The additional computational cost of LightSBB-M relative to LightSB-M arises from learning the implicit inverse map $\mathscr{Y}_t$, defined as the inverse of $\mathscr{X}_t(y) = y + \frac{1}{\beta} s_\theta(t, y)$, via the alternating outer loop. In practice this loop requires only $K = 5$ outer iterations, and we decrease the number of epochs as $k \leq K$ increases. Wall-clock times are reported on a single NVIDIA A100 GPU; times for LightSBB-M are averaged over all outer iterations k.

Three observations are in order. First, **inference cost is essentially identical** across all settings: the additional network $\mathcal{Z}_{\tilde{\theta}}$ requires only a single forward pass through a lightweight MLP at inference time. Second, for large β (e.g. $\beta = 100$), the inverse map reduces to the explicit first-order approximation $\mathscr{Y}_t(x) \simeq x - \frac{1}{\beta} s_\theta(t, x)$ (Algorithm 2), so no inverse network is needed; the moderate overhead ($\sim$4–5$\times$) comes purely from the $K = 5$ outer iterations. Third, for small β (e.g. $\beta = 1$) the inverse map must be learned explicitly, adding the "Inverse/epoch" cost. This regime is also where LightSBB-M provides the largest transport quality gains, so the training overhead is well-justified by the additional expressivity.

C.3 Recovering the Optimal Plan and Controls

This section details the setup behind Section 5.1. The reference solution is the Gaussian SBB of Hasenbichler et al. (2026), which is available in closed form. With $\mu_0 = \mathcal{N}(0, I_2)$ and $\mu_T =$

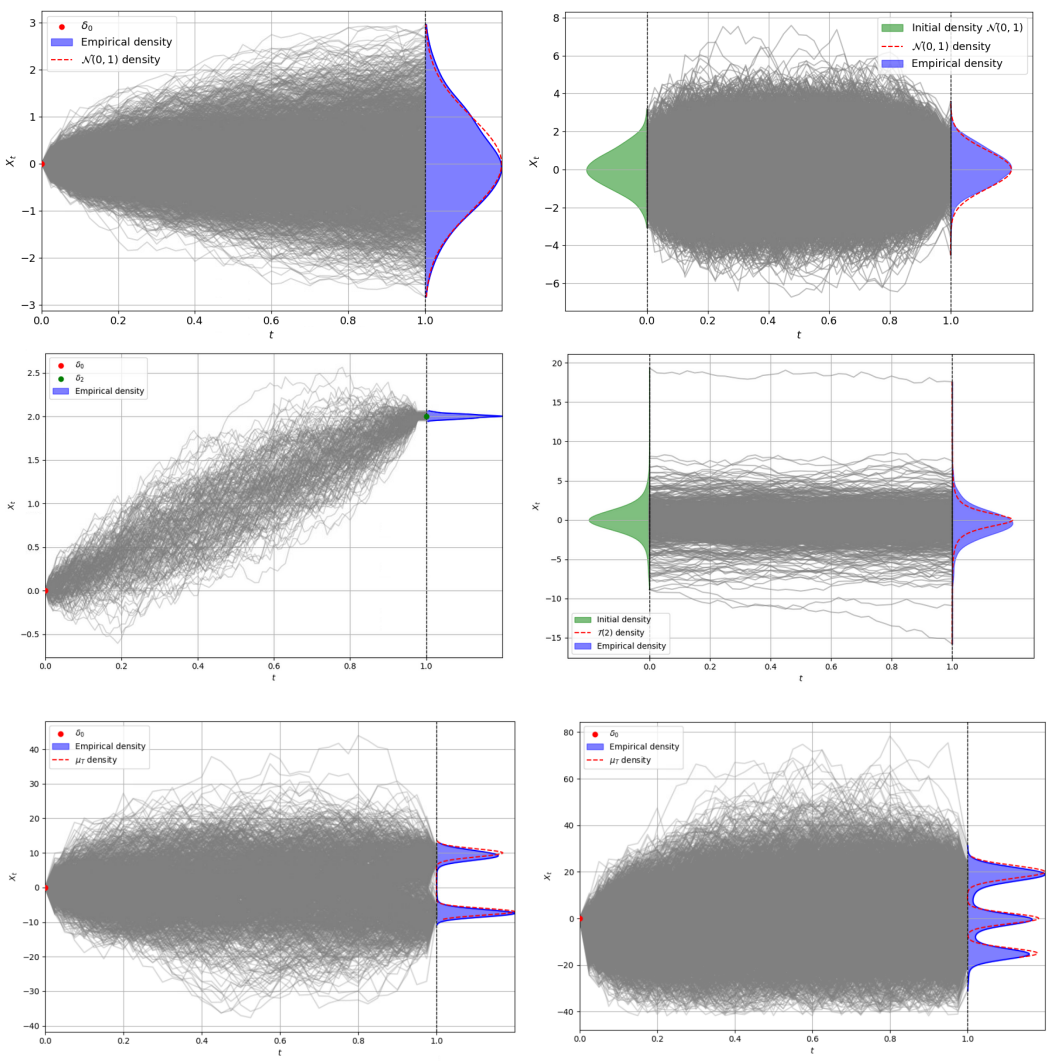


Figure 4: Transport interpolation examples. (*Top*) $\delta_0 \to \mathcal{N}(0,1)$ and $\mathcal{N}(0,1) \to \mathcal{N}(0,1)$. (*Middle*) $\delta_0 \to \delta_2$ and $\mathcal{T}(2) \to \mathcal{T}(2)$. (*Bottom*) $\delta_0 \to$ multimodal distribution.

$\mathcal{N}(0, \mathrm{diag}(10, 0.1))$ the problem decouples across coordinates, and the optimizer of coordinate i solves $\mathrm{d}X_t^i = a_i^\beta(t) X_t^i \,\mathrm{d}t + b_i^\beta(t)\,\mathrm{d}B_t^i$, with $\mathrm{Cov}(X_0^i, X_T^i) = 1/(1-r_i^\beta)$ where r_i^β is the admissible root of the associated quadratic. The optimal cost J^* follows by quadrature along the resulting linear dynamics. We use $K = 5$ outer iterations, 20,000 epochs and five seeds, with the hyperparameters of Table 4, and evaluate on 50,000 sampled pairs.

The model parameterizes neither the intermediate states nor the volatility, so both have to be reconstructed. Since Y is a Schrödinger Bridge, we sample Y_t by Brownian bridging between the learned endpoints (Y_0, Y_T) through Equation (12), and obtain the states as $X_t = Y_t + \frac{\varepsilon}{\beta}\nabla_y \log h_t^*(Y_t)$. Applying Itô's formula to that map gives the two learned controls,

$$\hat{\vartheta}(t) = \sqrt{\varepsilon}\Big(I_d + \tfrac{1}{\beta} D_y s_\theta(t, Y_t)\Big), \qquad \hat{\mathrm{a}}(t) = s_\theta + \tfrac{1}{\beta}\Big(\partial_t s_\theta + D_y s_\theta \, s_\theta + \tfrac{\varepsilon}{2}\Delta s_\theta\Big),$$

where $s_\theta := \varepsilon \nabla_y \log h_t^\theta$ is the learned score. Both are evaluated at Y_t and indexed by X_t, so the map $\mathscr{Y}_t$ never has to be inverted.

Writing π^* for the exact joint plan and $\hat{\pi}$ for the learned one, the reported quantities are the sliced Wasserstein distances $SW_2(\hat{\mu}_T, \mu_T)$ and $SW_2(\hat{\pi}, \pi^*)$, the relative Frobenius error on the cross-

Table 5: Wall-clock times on a single NVIDIA A100. "Inverse/epoch" refers to training the map $\mathcal{Z}_{\hat{\theta}}$; "—" indicates Algorithm 2 is used (no inverse network needed). Inference is evaluated on 10,000 samples for 2D tasks and 50 samples for the 512-dim latent space.

Model	Total	Drift/epoch	Inverse/epoch	Inference
2D $\mathcal{N} \to$ moons				
LightSB-M	38s	38s	—	8s
LightSBB-M ($\beta = 100$)	155s	31s	—	8s
LightSBB-M ($\beta = 1$)	322s	31s	33s	8s
ALAE 512-dim latent space				
LightSB-M	38s	35s	—	16s
LightSBB-M ($\beta = 100$)	178s	35s	—	19s
LightSBB-M ($\beta = 1$)	340s	31s	53s	19s

covariance $\mathrm{Cov}(X_0, X_T)$, and the relative objective gap $(\hat{J} - J^*)/|J^*|$. Because LightSB-M pays no volatility cost while J^* charges the optimum for one, its raw gap is measured against an inflated reference and can even come out negative without its drift being any better. Table 1 therefore reports a drift gap that removes the term $\beta\|\sigma_t - \sqrt{\varepsilon} I_d\|^2$ from both sides, which is the only comparison of the two objectives that is like for like. On the full objective, LightSBB-M attains 0.068 at $\beta = 10$ against 0.106 for LightSB-M, and 8.3×10^{-4} at $\beta = 100$ against 0.011. At $\beta = 100$ the dispersion of that gap across seeds is several times its mean for both methods, so the defensible reading is that LightSBB-M is indistinguishable from the optimum there rather than within any particular fraction of it.

The control recovery errors are

$$E_a = \frac{\int_0^T \mathbb{E}\|\hat{\mathrm{a}}(t) - \mathrm{a}^*(t, X_t^*)\|^2 \mathrm{d}t}{\int_0^T \mathbb{E}\|\mathrm{a}^*(t, X_t^*)\|^2 \mathrm{d}t}, \qquad E_b = \frac{\int_0^T \|\hat{\vartheta}(t) - \vartheta^*(t)\|_F^2 \mathrm{d}t}{\int_0^T \|\vartheta^*(t)\|_F^2 \mathrm{d}t}.$$

The numerator of E_a is evaluated at the states the model itself reconstructs, while both denominators use the exact solution. A misplaced map is thus penalized through the numerator rather than excused by a matching denominator. Both integrals are computed on a time grid truncated away from $t = T$ by the same margin for the learned and the exact quantities, so that the truncation does not bias the comparison.

C.4 Model Architecture

The transport map $\mathcal{Z}_\theta$ is defined by a simple multilayer perceptron (MLP). The network takes as input the time $t \in \mathbb{R}$ and the state $x \in \mathbb{R}^d$. Each input is first processed by a feed-forward network (FFN) comprising a linear layer, a normalization layer, a GELU activation function, and a final linear layer. This projects the inputs into latent spaces of dimension t_{model} and d_{model}, respectively. The resulting embeddings are then concatenated and passed through a subsequent FFN with a similar structure, which maps the combined representation back to the original space $\mathbb{R}^d$.

C.5 Quantitative Evaluation and Hyperparameter Analysis

For these experiments, we set the model hyperparameters to $t_{\text{model}} = 8$ and $d_{\text{model}} = 32$ with $J = 50$ potentials in the Gaussian mixture. To remain consistent with the setup in Tong et al. (2024b), we use $\varepsilon = 1$ for all datasets, except for *moons*$\to$8 *gaussians*, where we set $\varepsilon = 5$. Table 6 reports the 2-Wasserstein distances ($\mathcal{W}_2$) obtained for varying values of β across all considered datasets.

Table 6: 2-Wasserstein distances ($\mathcal{W}_2$) on synthetic datasets (lower is better) for different β values. The best results are highlighted in **bold**.

	$\mathcal{W}_2$ (↓)		
β	$\mathcal{N} \to$ 8gaussians	moons $\to$ 8gaussians	$\mathcal{N} \to$ moons
1	4.103±1.247	2.312±1.089	0.927±0.743
10	**0.241±0.083**	0.603±0.048	0.372±0.032
50	0.277±0.096	0.243±0.029	0.152±0.024
100	0.330±0.077	**0.201±0.034**	**0.109±0.014**
1000	0.462±0.091	0.232±0.025	0.171±0.037

Influence of β. Figure 5 (left) shows that $\mathcal{W}_2$ distances decrease rapidly as β increases from 10 to 100, except for $\mathcal{N} \to$ 8gaussians which keeps increasing, with optimal performance achieved between $\beta = 10$ and 100 across all datasets. Beyond $\beta = 100$, performance slightly degrades, suggesting an intermediate β value provides the best trade-off between drift and volatility. Figure 6 visualizes the transport results for all β values. In each subplot, the initial distribution μ_0 is shown in *blue*, and the transported distribution μ_T is shown in *orange*. For the smallest tested value, $\beta = 1$, the algorithm fails to converge. This behavior can be attributed to the violation of the required condition $\beta > \frac{1}{T}$, which is not satisfied when $T = 1$.

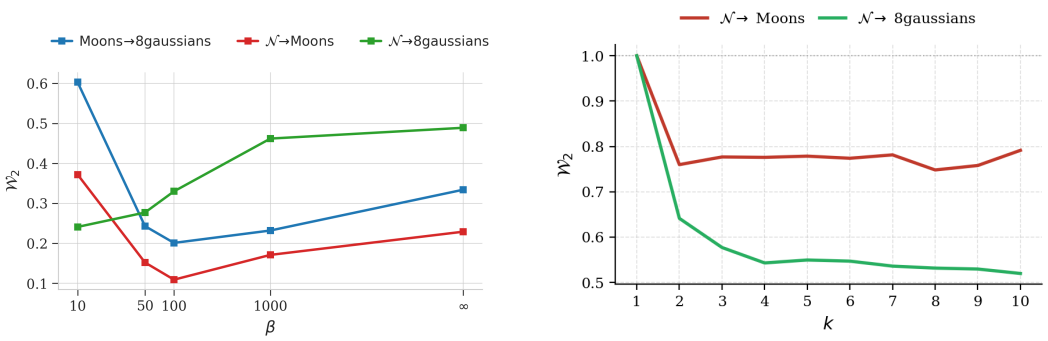
Convergence of the outer loop. Figure 5 (right) shows the normalized 2-Wasserstein distance $\mathcal{W}_2(k)/\mathcal{W}_2(1)$ as a function of the number of outer iterations K for two synthetic benchmarks with both $\beta = 100$ and $\beta = 10$, in order to illustrate the two regimes. Values are normalized by the first-iteration distance to allow comparison across tasks with different absolute scales, and averaged across multiple seeds. Both tasks exhibit rapid convergence within the first 3–5 iterations, after which the $\mathcal{W}_2$ distance plateaus. The $\mathcal{N} \to$ 8gaussians task converges monotonically, while $\mathcal{N} \to$ Moons stabilizes with minor fluctuations but shows no divergence. This empirically justifies the choice of $K = 5$ used throughout our experiments and suggests that the alternating procedure converges in practice, despite the absence of a formal convergence proof for the outer loop.

Practical selection of β. The hyperparameter β controls the trade-off between drift and volatility: larger β recovers the classical SB, while smaller β increases the role of stochastic volatility. In practice, one can choose β via a coarse grid search (e.g., $\beta \in \{1, 10, 50, 100\}$) using a held-out metric such as the 2-Wasserstein distance. On the synthetic benchmarks, the optimal β consistently lies in $[10, 100]$ (Table 6), and performance is not sensitive to the precise value within this range. The boundary value $\beta T = 1$ exhibits setting-dependent behavior: it fails on the 2D benchmarks (Table 6), but yields the best age translation quality on the ALAE latent space (Table 3). By Proposition A.1, the map $\mathscr{Y}_t$ remains well-defined only while $D^2\psi^*$ stays strictly below βI_d, but as $\beta T \to 1$, this margin vanishes. The 2D targets are sharply multimodal or geometrically non-convex, inducing high-curvature potentials that exhaust this margin, whereas the ALAE latent space is *regularized* by the pretrained autoencoder toward a smooth geometry Pidhorskyi et al. (2020) that remains within the admissibility bound. When βT falls well below 1 (typically from values around $\beta = 0.6, T = 1$), the algorithm fails across all settings, confirming that $\beta T > 1$ remains necessary in practice.

C.6 Unpaired Image-to-Image Translation Setup

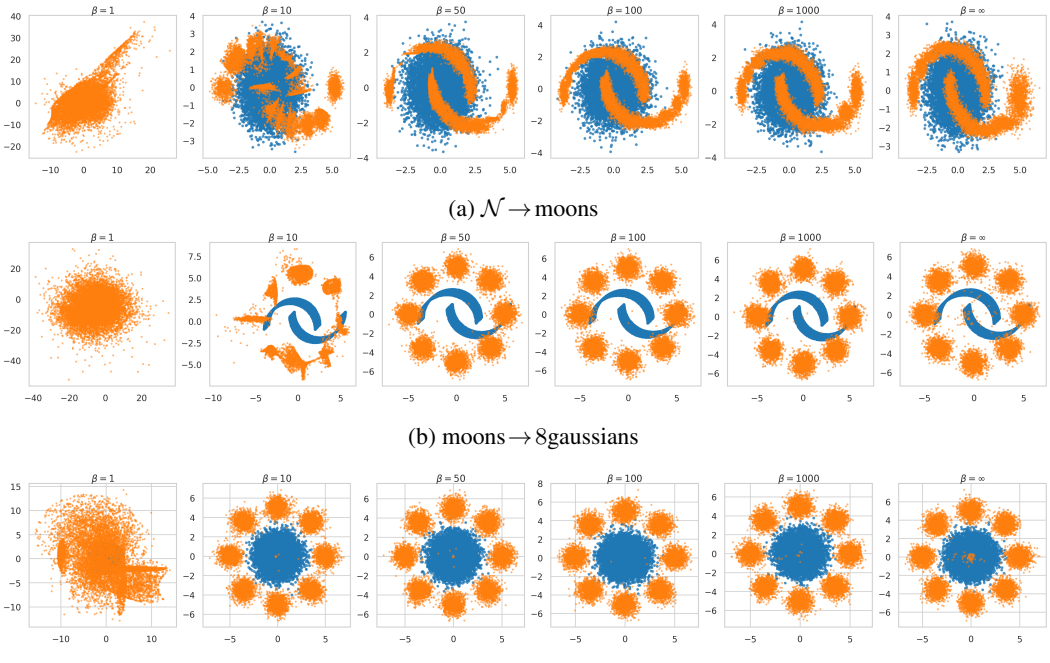
For unpaired image-to-image translation, we closely follow the experimental setup in Gushchin et al. (2024). The dataset consists of 70,000 labeled images, of which 60,000 were used for training and 10,000 for testing. The model hyperparameters were set to $t_{\text{model}} = 32$ and $d_{\text{model}} = 128$, with $J = 10$ potentials in the Gaussian mixture. This results in a drift model with approximately 2.6 million parameters, and a transport map estimator with 100,000 parameters. Training took under 5 minutes, and inference on 10 input images required less than 1 minute.

Our implementation builds upon the code provided by Gushchin et al. (2024), available at: `https://github.com/SKholkin/LightSB-Matching/tree/main`.



(a) $\mathcal{W}_2$ vs. β (b) $\mathcal{W}_2$ vs. K (normalized)

Figure 5: (a) Evolution of the 2-Wasserstein distance for different values of β. (b) Normalized 2-Wasserstein distance $\mathcal{W}_2(k)/\mathcal{W}_2(1)$ as a function of outer iterations K, showing convergence within 3–5 iterations.



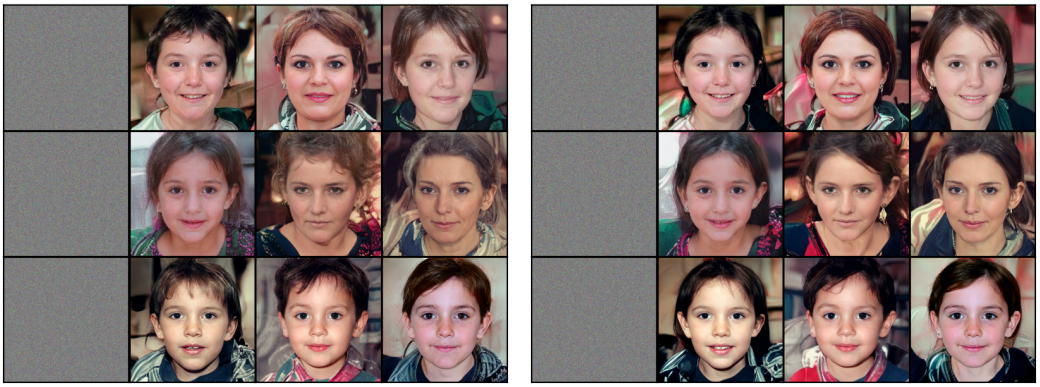
(a) $\mathcal{N} \to$ moons

(b) moons $\to$ 8gaussians

(c) $\mathcal{N} \to$ 8gaussians

Figure 6: Transport results across datasets for different β values. Initial distributions μ_0 are shown in *blue*, and transported distributions μ_T in *orange*.

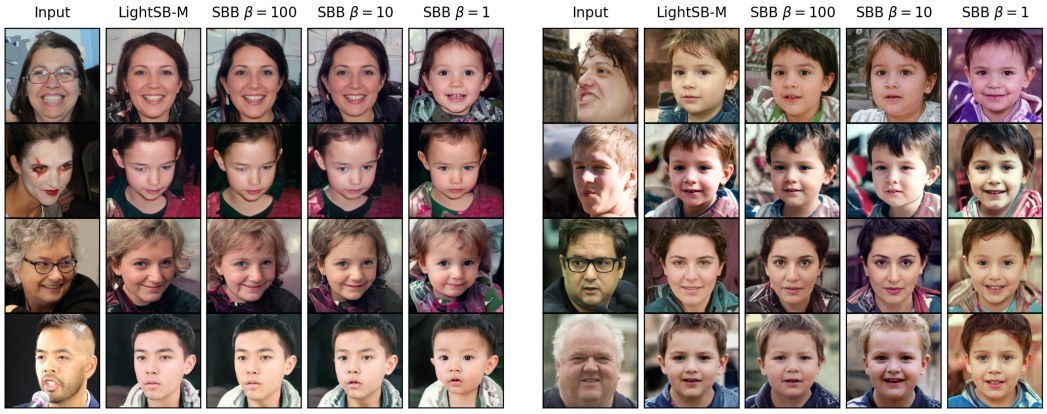
Additional result We propose the experiment illustrated in Figure 7, which generates a child image directly from noise. Unlike conventional diffusion models that rely on a backward-forward sampling Song & Ermon (2019); Song et al. (2021) scheme, our approach proceeds in a single forward pass, thereby avoiding the costly reverse diffusion step and error accumulation. We also present the comparison between the Y and X process, where we observe that the inverse sample X_T significantly improves sample quality and corrects potential errors of Y_T. This improvement is evident both in terms of visual fidelity and distributional alignment, as Y_T does not always give child images.



Y_T X_T

Figure 7: Comparison between Y_T and X_T (*right*) with $\beta = 5$. The left column shows the input image, and three representative samples are reported for each method.

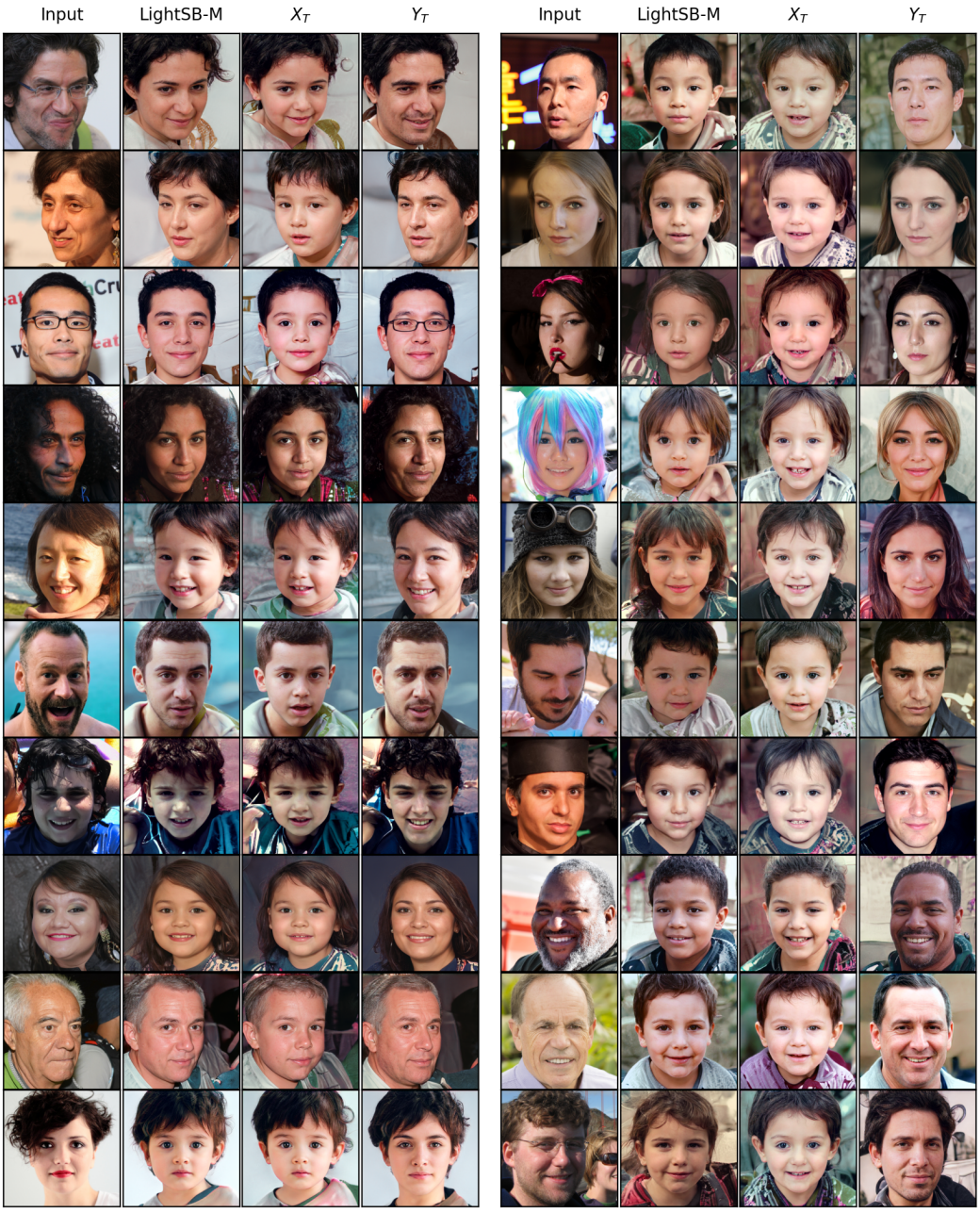
Effect of β and ε. Figure 8 varies both parameters, with LightSB-M as the reference. At $\varepsilon = 0.1$ small values of β give higher visual quality and better fidelity than LightSB-M, several of whose outputs are clearly not children. Raising the noise to $\varepsilon = 1$, where the classical SB has high variance, small β instead yields greater diversity among the generated outputs. As β grows the outputs approach those of the standard SB, which is what the theory predicts and what motivates the choice $\beta = 1$ used in the main text.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 1$

Figure 8: Comparison between our framework SBB and the benchmark LightSB-M across several (β, ε) settings. The left column shows the input image.

Finally, Figure 9 provides additional examples of our method, showing both X_T and Y_T for $\beta = 1$ and $\varepsilon \in \{0.1, 1\}$, with LightSB-M used as the baseline for comparison.



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 1$

Figure 9: Additional examples for unpaired image-to-image translation using LightSB-M as the baseline. Results show both X_T and Y_T from SBB for $\beta = 1$ and $\varepsilon \in \{0.1, 1\}$.