
Go With the Flow: Fast Diffusion for Gaussian Mixture Models

George Rapakoulias

Ali Reza Pedram
Georgia Institute of Technology

Panagiotis Tsiotras

Abstract

Schrödinger Bridges (SB) are diffusion processes that steer, in finite time, a given initial distribution to another final one while minimizing a suitable cost functional. Although various methods for computing SBs have recently been proposed in the literature, most of these approaches require computationally expensive training schemes, even for solving low-dimensional problems. In this work, we propose an analytic parametrization of a set of feasible policies for steering the distribution of a dynamical system from one Gaussian Mixture Model (GMM) to another. Instead of relying on standard non-convex optimization techniques, the optimal policy within the set can be approximated as the solution of a low-dimensional linear program whose dimension scales linearly with the number of components in each mixture. Furthermore, our method generalizes naturally to more general classes of dynamical systems such as controllable Linear Time-Varying systems that cannot currently be solved using traditional neural SB approaches. We showcase the potential of this approach in low-to-moderate dimensional problems such as image-to-image translation in the latent space of an autoencoder, and various other examples. We also benchmark our approach on an Entropic Optimal Transport (EOT) problem and show that it outperforms state-of-the-art methods in cases where the boundary distributions are mixture models while requiring virtually no training.

1 Introduction and Background

The problem of finding mappings between distributions of data, originally known as the Optimal Transport (OT) problem in mathematics, has received significant attention in recent years in multiple research fields, due to its application in problems such as generative AI (Ruthotto & Haber, 2021; Arjovsky et al., 2017), biology (Bunne et al., 2023b; Bunne & Rätsch, 2023; Tong et al., 2020), population dynamics, (Bunne et al., 2022) and control theory (Chen et al., 2015a,b; Rapakoulias & Tsiotras, 2024) among many others. Despite appearing static in nature, reformulating OT in the context of dynamical systems imbues it with further structure and unlocks tools from the literature on dynamical systems that can be employed for its efficient solution (Benamou & Brenier, 2000).

To set the stage, consider two distributions ρ_i, ρ_f , supported in the d -dimensional Euclidean space, denoted by \mathbb{R}^d , and consider the regularized version of the static OT optimization problem, known as Entropic Optimal Transport (EOT) (Peyré & Cuturi, 2019),

$$\min_{\pi \in \Pi(\rho_i, \rho_f)} \int \frac{1}{2} \|x_0 - x_1\|^2 d\pi(x_0, x_1) - \epsilon h(\pi), \quad (1)$$

where $\pi(x_0, x_1)$ is the transport plan (also referred to as coupling) between ρ_i, ρ_f , $\Pi(\rho_i, \rho_f)$ is the set of all joint distributions with marginals ρ_i, ρ_f , and h is the differential entropy, defined by

$$h(\rho) \triangleq - \int \rho(x) \log \rho(x) dx. \quad (2)$$

The corresponding dynamic formulation, known as the Schrödinger Bridge Problem (SBP) (Léonard, 2014; Chen et al., 2021), when formulated as a stochastic optimal control problem, is given by

$$\min_{u \in \mathcal{U}} J_{\text{SB}} \triangleq \mathbb{E}_{x_t \sim \rho_t} \left[\int_0^1 \frac{1}{2\epsilon} \|u_t(x_t)\|^2 dt \right], \quad (3a)$$

$$dx = u_t(x) dt + \sqrt{\epsilon} dw, \quad (3b)$$

$$x_0 \sim \rho_i, \quad x_1 \sim \rho_f, \quad (3c)$$

where the objective is to find an optimal drift function $u_t(x)$, also referred to as the control policy in the

context of control applications, belonging to a set of adapted finite-energy policies \mathcal{U} , such that, when applied to the stochastic dynamical system (3b), guarantees that, for initial conditions sampled at time $t = 0$ from ρ_i , the state at time $t = 1$ will be distributed according to ρ_f , and the cost (3a) will be minimized.

The increased interest in solving the EOT and SB problems, especially in high-dimensional applications where the boundary distributions ρ_i, ρ_f are only available through a finite number of samples, has led to the development of a multitude of algorithms over the recent years. One of the first methods proposed is the Dynamic Iterative Proportional Fitting Method (IPF) method (Chen et al., 2016a; Pavon et al., 2021; De Bortoli et al., 2021), originally conceptualized by Fortet (1940) as a constructive proof for existence and uniqueness of solutions of the SBP problem. Another approach, leveraging forward-backward Stochastic Differential Equations (SDEs) attempts to locally solve the optimality conditions of problem (3), namely, the Hamilton-Jacoby-Bellman (HJB) and Fokker-Plank-Kolmogorov (FPK) Partial Differential Equations (PDEs), using a modified version of the Feynman-Kac lemma (Chen et al., 2022; Liu et al., 2022b). A third technique leveraging a Wasserstein Proximal recursion algorithm is presented in (Caluya & Halder, 2021) and (Bunne et al., 2022), where the solution to problem (3) is successively approximated by solving a series of initial value problems.

Since for general boundary distributions ρ_i, ρ_f the optimal drift $u_t(x)$ does not attain a closed form, it is usually approximated with an appropriate parametric function approximator. In high-dimensional machine learning applications, Neural Networks are usually employed (De Bortoli et al., 2021), while for lower-dimensional problems parameterizations using basis functions have been explored (Pavon et al., 2021). Regardless of the type of parametric approximation, the parameters are usually optimized using some stochastic optimization technique. To this end, a popular approach for categorizing existing approaches is to determine whether or not the objective function in the parameter optimization scheme requires simulating trajectories of the dynamical system (3b), due to the increased memory and computational requirements of the latter. Techniques requiring trajectory simulations can be further categorized as to whether they optimize the objective function (3a) explicitly or implicitly by attempting to solve the first-order optimality conditions of the problem. Popular approaches belonging to the first category are (Tong et al., 2020; Onken et al., 2021; Rapakoulias & Tsotras, 2024; Ruthotto et al., 2020), while (De Bortoli et al., 2021; Chen et al., 2022; Liu et al., 2022b) belong to the later.

Simulation-free approaches leverage properties of problem (3), such as the decomposition of the optimal probability flow into conditional problems that are easier to solve, sometimes even analytically (Chen et al., 2016b; Lipman et al., 2023; Liu et al., 2023). In the latter category, a recent technique known as Diffusion Schrödinger Bridge Matching (DSBM) (Peluchetti, 2023; Shi et al., 2023), or its deterministic counterpart, known as Flow Matching (FM) Lipman et al. (2023) or Rectified Flow (RF) Liu et al. (2023), leverages the decomposition of the optimal probability flow to a mixture of flows conditioned on their respective endpoints and retrieves an approximation of the optimal policy that solves (3) as a mixture of conditional policies that are easy to calculate. Theoretically, one needs to combine an infinite number of conditional flows to retrieve the true flow, due to the continuous support of the boundary distributions. To overcome this issue, a neural network is usually trained to approximate this infinite mixture.

In this paper, we use a similar flow decomposition to solve the problem of finding a policy that can steer the distribution of a dynamical system from a Gaussian Mixture Model (GMM) to another one, using a mixture of conditional policies that can each steer the individual components of the initial mixture to the components of the terminal mixture. We also show that our approach shares similarities with recent lightweight Schrödinger Bridge solvers such as (Korotin et al., 2024; Gushchin et al., 2024), which use Gaussian Mixture Models to parameterize the Schrödinger potentials.

Specifically, we claim the following **main contributions**:

1. We present an efficient, training-free method to solve the Schrödinger Bridge problem in the case where the boundary distributions are Gaussian Mixture Models.
2. Contrary to existing approaches, our method can handle both the stochastic and the deterministic versions of the problem (3). Being based on a control-theoretic formulation, it also naturally generalizes to dynamical systems with a general Linear Time-Varying (LTV) structure, with the control input u and stochastic component dw having different dimensions than the state, which could be of interest in Mean Field Games and multi-agent control applications (Ruthotto et al., 2020; Liu et al., 2022b; Chen, 2024).
3. We demonstrate our algorithm in both low-dimensional toy problems, moderate-dimensional image-to-image translation tasks, and Entropic Optimal Transport benchmark problems, showing

that it outperforms state-of-the-art lightweight methods for solving the SB problem both in terms of training speed and accuracy of the learned boundary distributions, when these are available through samples (40% better FID scores in the image translation task).

2 Preliminaries

2.1 Diffusion Schrödinger Bridge Matching and Flow Matching

The composition of diffusion processes as mixtures of processes conditioned on their endpoints was originally proposed by Peluchetti (2021) as a simulation-free algorithm for generative modeling applications. The concept was later tailored to solve the Schrodinger Bridge problem in the DSBM algorithm (Shi et al., 2023), proposed concurrently by Peluchetti (2023). Similar simulation-free methods have also been proposed to solve variants of the same problem in (Albergo & Vanden-Eijnden, 2023) and in (Liu et al., 2024) for the stochastic bridge setting, as well as in Liu et al. (2023) and Lipman et al. (2023) for the deterministic setting. For a more comprehensive overview and comparison of the available methods, we refer the reader to (Shi et al., 2023, Section 5) and (Peluchetti, 2023, Section 5).

Given problem (3), the main idea is to decompose the problem into a sequence of elementary conditional subproblems that are easier to solve, and then express the solution as a mixture of the solutions of the conditional subproblems. This idea has an intuitive motivation: Informally, finding a policy that transports the state distribution from an initial to a target density can be separated into two problems. First one needs to calculate a transport plan solving the “who goes where” problem and then calculate a point-to-point optimal policy, that solves the “how to get there” problem (Terpin et al., 2024). In many cases, the two are known to be decoupled (Chen et al., 2021, 2016b; Terpin et al., 2024); most importantly, however, the latter can be solved analytically for simple dynamical systems such as (3b).

More rigorously, the optimal probability flow ρ_t of problem (3) is known (Föllmer, 1988; Chen et al., 2021) to admit the decomposition

$$\rho_t^*(x) = \int W_{t|x_0, x_1}(x) d\pi_\epsilon^*(x_0, x_1), \quad (4)$$

where $W_{t|x_0, x_1}(x)$ is the probability density of the unforced dynamics ($u_t \equiv 0$) of (3b), namely the Brownian motion kernel, pinned at x_0 for $t = 0$ and at x_1 for $t = 1$, and $\pi_\epsilon^*(x_0, x_1)$ is the entropic optimal transport plan between ρ_i, ρ_f solving (1). Furthermore, the

calculation of $W_{t|x_0, x_1}(x)$ can be expressed as the solution to the following optimal control problem (Dai Pra, 1991)

$$\min_{u_{t|0,1} \in \mathcal{U}} J_{0,1}(x_0, x_1) \triangleq \mathbb{E} \left[\int_0^1 \|u_{t|0,1}(x)\|^2 dt \right], \quad (5a)$$

$$dx = u_{t|0,1}(x) dt + \sqrt{\epsilon} dw, \quad (5b)$$

$$x_0 \sim \delta_{x_0}, \quad x_1 \sim \delta_{x_1}, \quad (5c)$$

where $\delta_{x_0}, \delta_{x_1}$ are Dirac delta functions centered on x_0 and x_1 respectively. Assuming $\rho_{t|0,1}(x)$ and $u_{t|0,1}(x)$ solve (5), one can construct a feasible solution for the original problem (3) using any transport plan $q(x_0, x_1) \in \Pi(\rho_i, \rho_f)$, i.e., any joint distribution between the desired boundaries ρ_i, ρ_f , using the mixtures

$$\rho_t(x) = \int \rho_{t|0,1}(x) q(x_0, x_1) dx_0 dx_1, \quad (6a)$$

$$u_t(x) = \int u_{t|0,1}(x) \frac{\rho_{t|0,1}(x) q(x_0, x_1)}{\rho_t(x)} dx_0 dx_1. \quad (6b)$$

Showing that the flow (6a) is a feasible solution to (3) for any valid coupling $q(x_0, x_1)$ amounts to observing that, indeed, it satisfies the boundary distributions ρ_i, ρ_f at times $t = 0$ and $t = 1$, respectively. Proving that the policy (6b) produces (6a) amounts to showing that the pair (6a), (6b) satisfies the FPK PDE (Lipman et al., 2023; Liu et al., 2024). When $q(x_0, x_1) = \pi_\epsilon^*(x_0, x_1)$ equations (6a) reduces to (4) and (6b) recovers the optimal policy solving (3) (Shi et al., 2023; Peluchetti, 2023).

2.2 Schrödinger Bridges with Gaussian Marginals

The Schrödinger bridge problem with Gaussian Marginals, henceforth referred to as the Gaussian Schrodinger Bridge (GSB), has been extensively studied in the literature and can be solved either analytically for simple choices of prior dynamics (Bunne et al., 2023a) or as a convex semidefinite optimization problem for general linear dynamical systems both for continuous and discrete time cases (Chen et al., 2015b; Liu et al., 2022a). Because we use the GSB solution as a building block to construct a policy that works with general GMM boundary distributions, we briefly review the available methods for its solution here.

To this end, consider the optimization problem with Gaussian marginals

$$\min_{u \in \mathcal{U}} \mathbb{E} \left[\int_0^1 \|u_t(x)\|^2 dt \right], \quad (7a)$$

$$dx = u_t(x) dt + \sqrt{\epsilon} dw, \quad (7b)$$

$$x_0 \sim \mathcal{N}(\mu_i, \Sigma_i), \quad x_1 \sim \mathcal{N}(\mu_f, \Sigma_f), \quad (7c)$$

where $\mu_i, \Sigma_i, \mu_f, \Sigma_f$ are the means and covariances of the initial and final Gaussian boundary distributions respectively. Since the optimal distribution of the state that solves (7) remains Gaussian, i.e., $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$, problem (7) reduces to that of the control of the mean and the covariance of the state. Controlling the mean and the covariance of the state can be achieved with a drift parametrization of the form

$$u_t(x) = K_t(x - \mu_t) + v_t, \quad (8)$$

where $K_t \in \mathbb{R}^{n \times n}$ and $v_t \in \mathbb{R}^n$ is a feedforward term that controls the dynamics of the mean distributions. It can be shown that the policy (8) is optimal and that, for this policy, the propagation of the mean and the covariance of the state are decoupled. More specifically, when applying (8) to (7b), the mean μ_t and covariance Σ_t propagate according to

$$\dot{\Sigma}_t = K_t \Sigma_t + \Sigma_t K_t^\top + \epsilon I, \quad (9a)$$

$$\dot{\mu}_t = v_t, \quad (9b)$$

which when substituted in place of (7b), turn (7) into an optimization problem in the space of affine feedback control laws of the form (8).

Theorem 1. (Bunne et al., 2023a) *The optimal solution to the problem (7) is given by*

$$\Sigma_t = (1-t)^2 \Sigma_i - t^2 \Sigma_f + (1-t)t(C_\sigma + C_\sigma^\top + \epsilon^2 I) \quad (10a)$$

$$\mu_t = (1-t)\mu_i + t\mu_f, \quad (10b)$$

$$K_t = S_t^\top \Sigma_t^{-1}, \quad (10c)$$

$$v_t = \mu_f - \mu_i, \quad (10d)$$

$$S_t = t(\Sigma_f - C_\sigma^\top) - (1-t)(\Sigma_f - C_\sigma) + \frac{\epsilon}{2}(1-2t)I, \quad (10e)$$

where $C_\sigma = \frac{1}{2}(\Sigma_i^{\frac{1}{2}} D_\sigma \Sigma_i^{-\frac{1}{2}} - \epsilon I)$, $D_\sigma = (4\Sigma_i^{\frac{1}{2}} \Sigma_f \Sigma_i^{\frac{1}{2}} - \epsilon^2 I)^{\frac{1}{2}}$.

The more general case where the dynamical system (7b) has a Linear Time Varying (LTV) structure of the form

$$dx = A_t x_t dt + B_t u_t dt + D_t dw, \quad (11)$$

where $x_t \in \mathbb{R}^d$, $u_t \in \mathbb{R}^m$, $w_t \in \mathbb{R}^q$ has also been extensively studied in the context of control theory and is known as the Covariance Steering Problem (Chen et al., 2015a,b; Bakolas, 2018; Liu et al., 2022a). Although it is a non-convex optimization problem in the space of affine feedback policies, it can be losslessly relaxed to a convex problem for both continuous and discrete-time cases (Chen et al., 2015b; Liu et al., 2022a), and therefore attains an efficient and exact calculation which we will exploit in the sequel.

3 Fast Diffusion for Mixture Models

Equation (6b) expresses the policy of problem (3) as an infinite mixture of conditional, point-to-point policies. In this section, we extend this idea to construct a mixture policy, consisting of conditional policies each solving a Gaussian bridge sub-problem of the form (7), to solve a Bridge problem with Gaussian Mixture Model (GMM) boundary conditions. To this end, consider the problem

$$\min_{u \in \mathcal{U}} J_{\text{GMM}} \triangleq \mathbb{E} \left[\int_0^1 \|u_t(x)\|^2 dt \right], \quad (12a)$$

$$dx = u_t(x) dt + \sqrt{\epsilon} dw, \quad (12b)$$

$$x_0 \sim \text{GMM}(\mu_0^i, \Sigma_0^i, w_0^i) \triangleq \sum_{i=1}^{N_0} \mathcal{N}(\mu_0^i, \Sigma_0^i) w_0^i, \quad (12c)$$

$$x_1 \sim \text{GMM}(\mu_1^j, \Sigma_1^j, w_1^j) \triangleq \sum_{j=1}^{N_1} \mathcal{N}(\mu_1^j, \Sigma_1^j) w_1^j. \quad (12d)$$

The main result is summarized in the following theorem.

Theorem 2. Consider problem (12), with N_0 components in the initial mixture and N_1 components in the terminal mixture. Assume that $u_{t|ij}$ is the conditional policy that solves the (i, j) -GSB problem, that is the bridge from the i -th component of the initial mixture, to the j -th component of the terminal mixture and let the resulting probability flow be $\rho_{t|ij}$. Furthermore, let $\lambda_{ij} \geq 0$ such that, for all $j \in \{1, 2, \dots, N_1\}$, $\sum_i \lambda_{ij} = w^j$ and such that, for all $i \in \{1, 2, \dots, N_0\}$, $\sum_j \lambda_{ij} = w^i$. Then, the policy

$$u_t(x) = \sum_{i,j} u_{t|ij}(x) \frac{\rho_{t|ij}(x) \lambda_{ij}}{\sum_{i,j} \rho_{t|ij}(x) \lambda_{ij}} \quad (13)$$

is a feasible policy for Problem (12), and the corresponding probability flow is

$$\rho_t(x) = \sum_{i,j} \rho_{t|ij}(x) \lambda_{ij}. \quad (14)$$

The mixture policy (13) is a weighted average of conditional policies, weighted according to $\lambda_{ij} \rho_{t|ij}(x)$, while the denominator $\sum_{i,j} \rho_{t|ij}(x) \lambda_{ij}$ is just a normalizing constant. Since $\rho_{t|ij}(x)$ is a Gaussian distribution centered at the mean of the (i, j) -Gaussian bridge at time t , this weighing scheme prioritizes the conditional policies whose mean is closer to the value of x at the time t .

Thus far, we have constructed a set of feasible solutions to Problem (12), for all values of λ_{ij} satisfying the conditions of Theorem 2. To this end, we substitute the expression for the policy (13) in the cost function

(12a), to obtain

$$\begin{aligned} & \mathbb{E}_{x_t \sim \rho_t} \left[\int_0^1 \left\| \sum_{i,j} u_{t|ij}(x_t) \frac{\rho_{t|ij}(x_t) \lambda_{ij}}{\sum_{i,j} \rho_{t|ij}(x_t) \lambda_{ij}} \right\|^2 dt \right] \\ & \leq \sum_{i,j} \lambda_{ij} \mathbb{E}_{x_t \sim \rho_{t|ij}} \left[\int_0^1 \|u_{t|ij}(x_t)\|^2 dt \right] \end{aligned} \quad (15a)$$

$$= \sum_{i,j} \lambda_{ij} J_{ij}, \quad (15b)$$

where the upper bound is due to the discrete version of Jensen's inequality, and the integral in (15a) is the optimal cost of the (i, j) -Gaussian bridge, denoted J_{ij} . Therefore, instead of minimizing (12a) with respect to the parameters λ_{ij} directly, one can minimize the upper bound over their feasible domain, given in Theorem 2. More formally, we have the following theorem.

Theorem 3. *Let J_{ij} be the optimal cost of solving the (i, j) -Gaussian bridge subproblem of the form (7) with marginal distributions the i -th component of the initial and the j -th component of the terminal mixture. Then, the cost function of the linear optimization problem*

$$\min_{\lambda_{ij}} J_{\text{OT}} \triangleq \sum_{i,j} \lambda_{ij} J_{ij} \quad (16a)$$

$$\text{s.t. } \sum_j \lambda_{ij} = w_0^i, \quad i = 1, 2, \dots, N_0, \quad (16b)$$

$$\sum_i \lambda_{ij} = w_1^j, \quad j = 1, 2, \dots, N_1, \quad (16c)$$

$$\lambda_{ij} \geq 0, \quad (16d)$$

provides an upper bound for (12a), that is, $J_{\text{GMM}} \leq J_{\text{OT}}$, for all values of λ_{ij} satisfying (16b)-(16d).

For clarity of exposition, we defer the proofs of Theorems 2, 3 to Appendix A.

4 Related work

Although the idea of creating a mixture policy from elementary point-to-point policies is at the heart of flow-matching, to the best of our knowledge the benefits of the method for solving problems with mixture models with a finite number of components have not been explored.

In an early work, Chen et al. (2016b) developed an upper bound on the Wasserstein-2 distance between Gaussian mixture models, i.e., the deterministic version of problem (12), that matches the upper bound of Theorem 3. The problem of finding a policy that solves the dynamic problem, however, was not explored.

Focusing on works concerning dynamic optimal transport problems, the concept of constructing SDEs as

mixtures of Gaussian probability flows can be traced back to mathematical finance applications (Brigo, 2002; Brigo et al., 2002). More recently, and in the context of generative applications, Albergo & Vanden-Eijnden (2023) developed similar expressions for finding a policy for steering between mixture models using an alternative conditional solution for the Gaussian-to-Gaussian bridge and focused on the case of deterministic fully actuated dynamical systems with no prior dynamics. The discussion is limited to a theoretical derivation of a feasible policy and a fixed transport plan between the components of the mixtures, leaving the potential applications of the methods unexplored. In (Balci & Bakolas, 2023), a similar idea is explored for deterministic LTV systems with the key difference that the proposed policy results from a random selection from a set of conditional policies, instead of the mixture policy (13). The authors show that for deterministic systems, the optimal value within the set can be calculated exactly, and the resulting cost will equal the upper bound from Theorem (3).

Our work reveals some similarities in scope and structure with Light-SB (Korotin et al., 2024; Gushchin et al., 2023). Both works aim at providing numerically inexpensive tools to solve the SB problem in low- to moderate-dimensional scenarios. Furthermore, both works result in feedback policies that are of the form (13), i.e., a mixture of affine feedback terms weighted with exponential kernels; see (Korotin et al., 2024, Proposition 3.3). However, the formulations are otherwise quite distinct. Specifically, Light-SB works by modeling the so-called Schrödinger potentials using a Gaussian Mixture Model. This results in flows with boundary distributions that have a mixture-like structure and are optimal by construction, but whose marginals are neither amenable to exact calculation, nor are Gaussian Mixture models, in general. Finally, to approximate the optimal flows for boundary distributions that are available only through samples, Light-SB works by minimizing a negative log-likelihood objective, which is known to perform sub-optimally when the underlying distributions are Gaussian Mixture models (Bishop & Nasrabadi, 2006), while our approach works by first pre-fitting GMMs to the data using Expectation Maximization (EM) and then computing an optimal policy by solving exactly a linear program.

5 Experiments

5.1 2D problems

We first test the algorithm in 2D toy problems, starting from the case where the boundary mixture models are known explicitly. The resulting flows are illustrated in Figure 2 for a Gaussian-to-Gaussian Mix-

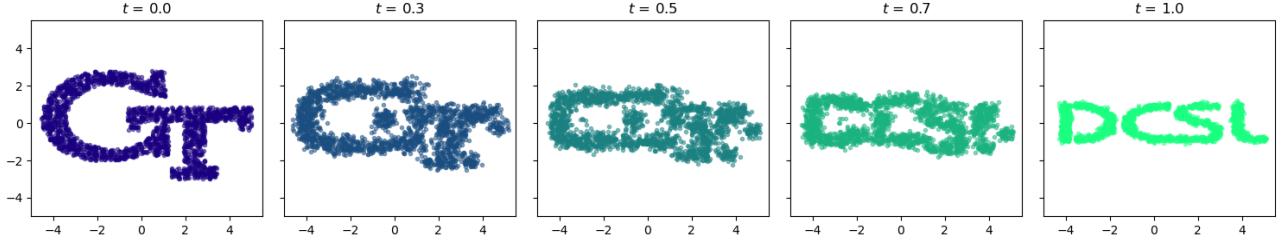


Figure 1: GT to DCSL distribution steering with zero prior dynamics and $\epsilon = 0.1$.

ture problem for various levels of noise. To study the optimality of the proposed approach, we evaluate the resulting transport cost for policy (13) for each noise level and compare it with the upper bound from (16a). Additionally, we benchmark our results against two state-of-the-art methods: Diffusion Schrödinger Bridge (DSB) (De Bortoli et al., 2021) and Diffusion Schrödinger Bridge Matching (DSBM) (Shi et al., 2023). The exact hyper parameters for the experiments are presented in Appendix B. The results are shown in Table 1. For the zero-noise case, DSB and DSBM are not applicable. Instead, we computed the true optimal transport cost using 10,000 samples from the boundary distributions via the POT library (Flamary et al., 2021).

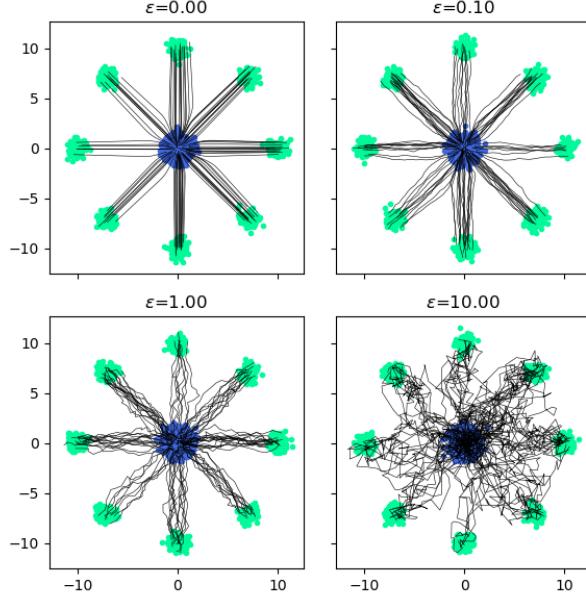


Figure 2: Gaussian to 8-Gaussians with no prior dynamics.

To test the performance on problems with boundary distributions available only through samples, we test the algorithm on the distributions depicted in Figure 1. To use the proposed approach, we pre-fit GMMs with 100 components in the initial and terminal distribu-

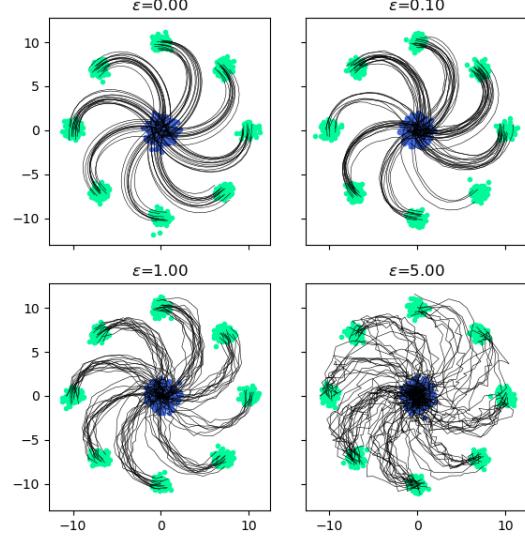


Figure 3: Gaussian to 8-Gaussians with LTI prior dynamics.

tions using the Expectation maximization (EM) algorithm (Pedregosa et al., 2011).

Table 1: Transport cost comparison.

ϵ	J_{GMM} (12a)	J_{OT} (16a)	DSBM	DSB	OT
0	89.45	100.06	-	-	84.87
0.1	89.32	100.15	84.26	98.62	-
1	89.28	102.28	131.50	100.82	-
10	116.60	162.13	133.04	244.31	-

5.2 Problems with LTI prior dynamics

To test the algorithm on more complicated dynamical systems, we use the 4-dimensional Linear Time-Invariant (LTI) system

$$dx_t = Ax_t dt + Bu_t dt + D dw, \quad (17)$$

with

$$A = \begin{bmatrix} 2S & I_2 \\ S & 0_2 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I_2 \end{bmatrix}, \quad D = \epsilon I_4,$$

and boundary distributions

$$\rho_i = \sum_{k=0}^8 \frac{1}{8} \mathcal{N}([10 \cos(k\pi/4); 10 \sin(k\pi/4); 0; 0], 0.4I_4), \quad (18a)$$

$$\rho_f = \mathcal{N}(0_4, 0.4I_4). \quad (18b)$$

We note that solving problem (12) with the dynamical system (17) in place of (12b) is not currently solvable using any mainstream neural SB solvers because the stochastic disturbance dw in (17) does not enter through the same channels as the control signal u_t and the state x_t . The only available method to solve this problem is detailed in (Chen et al., 2016b), which, however, assumes access to the solution of the static EOT problem (1) with boundary distributions (18), and a closed form of the probability density transition kernel included by the dynamical system (17) for $u_t \equiv 0$. The results of our approach are illustrated in Figure 3.

5.3 Performance on EOT benchmarks

To further evaluate the optimality of the proposed approach, we tested the algorithm on the Entropic Optimal Transport benchmark detailed in (Gushchin et al., 2023). The benchmark provides a pair of boundary test distributions ρ_i, ρ_f (see Figure 4), where ρ_i is a scaled Normal distribution and ρ_f is a mixture-like distribution whose density cannot be calculated exactly but is defined implicitly through an optimal conditional transport plan $\pi^*(x_1|x_0)$, which is a Gaussian Mixture Model and is known in closed form.

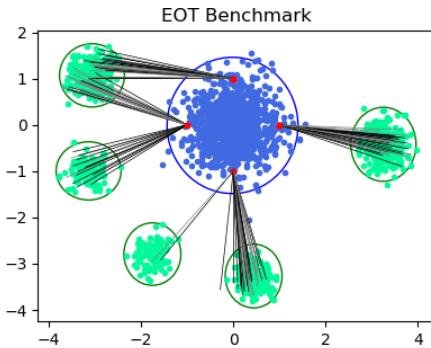


Figure 4: 2D EOT benchmark boundary distributions, from (Gushchin et al., 2023).

For the pair ρ_i, ρ_f , the optimal policy solving (3) can be calculated explicitly, allowing for direct comparisons with our approach. The metric we use to measure the optimality of our approach is the Bures-Wasserstein Unexplained Variance Percentage

Table 2: cBW-UVP metric for EOT benchmarks for various noise levels and problem dimensions.

	$d = 2$	$d = 16$	$d = 64$	$d = 128$
$\epsilon = 0.1$	10.35	14.68	29.20	11.83
$\epsilon = 1.0$	5.78	7.20	6.93	6.38
$\epsilon = 10$	0.16	0.28	1.43	2.77

(cBW-UVP) (Gushchin et al., 2023), defined by

$$\text{cBW-UVP}(\pi, \pi^*) \triangleq \frac{100\%}{\frac{1}{2}\text{Var}(\rho_1)} \int \text{BW}_2^2(\pi(x_1|x_0) \| \pi^*(x_1|x_0)) \rho_0(x_0) dx_0,$$

which measures the distance between conditional transport plans, evaluated using the Bures-Wasserstein metric.

To use the method of Gushchin et al. (2023), we first obtain samples from the two boundary test distributions and then fit mixture models on them using EM. We then deploy policy (13), and report the values of the cBW-UVP index between the known optimal conditional transport plan $\pi^*(x_1|x_0)$, and the conditional transport plan resulting from the integration of the policy (13). The results are reported in Table 2 for problems of various dimensions and noise levels. We note that although the distributions described in Gushchin et al. (2023) are not, in general, mixture models, they can be closely approximated as such. Therefore, the cBW-UVP values reported in Table 2 are partly due to the potential suboptimality of (13), and partly due to the mismatch between the true and the mixture model distributions. Nonetheless, Table 2 places our approach among the best scoring methods for problems with boundary distributions that are mixture modes or can be closely approximated as such (see Table 5 in (Gushchin et al., 2023) for the scores of most widely used Neural Network methods for solving the SB problem) while requiring virtually no training other than solving the linear program (16) and fitting the mixture models to the data if those are not explicitly known beforehand.

5.4 Image-to-Image translation

Although Mixture Models are not designed to capture high-dimensional distributions such as those of images, following (Korotin et al., 2024) we use our algorithm in the latent space of an autoencoder to perform a man-to-woman and adult-to-child image translation task. We use the pre-trained ALAE autoencoder (Pidhorskyi et al., June 2020) trained on the FFHQ (Karras et al., 2019). The latent space of the autoencoder is 512-dimensional. We start by fitting a 10-component mixture model to the embeddings of each image class



Figure 5: Man to woman image translation. The first column contains the original images, the second column contains the translated image using the baseline method [Korotin et al. \(2024\)](#), and the last contains 4 samples from the transformed images using the proposed approach with $\epsilon = 0.1$.



Figure 6: Adult to Children image translation. The first column contains the original images, the second column contains the translated image using the baseline method [Korotin et al. \(2024\)](#), and the last contains 4 samples from the transformed images using the proposed approach with $\epsilon = 0.1$.

using Scikit’s ([Pedregosa et al., 2011](#)) default EM algorithm with diagonal covariance matrices to facilitate matrix inversions in the Gaussian to Gaussian policy calculations (10), and then apply the mixture policy (13). The results are illustrated in Figures 5 and 6. To test how well the generated images match the features of the given target distribution, we calculate the Fréchet inception distance (FID) scores [Heusel et al. \(2017\)](#) between the actual and the generated images of a given class, using 10000 samples from each distribution. The FID scores correspond to the empirical Bure-Wasserstein distance between the images of the two classes, evaluated in the latent space of the Inception network. To further test how close the trans-

formed images are to the target class, we also calculate the empirical Bure-Wasserstein distance between the transformed images and the real images of the target class, directly in the latent space of the ALAE autoencoder, since this is the space our algorithm is trained in. We compare against two state-of-the-art lightweight SB solvers, LSB [Korotin et al. \(2024\)](#) and LSBM [Gushchin et al. \(2024\)](#), and present the results in Tables 3, 4.

Qualitatively, our algorithm performs more aggressive feature changes compared to the baseline method, as illustrated in Figures 5, 6. On a quantitative level, the features of the transformed images better capture the true distribution of the features of a given target

M→W	FID	ALAE-BW ₂
LSB	4.94	28.9
LSBM	4.98	28.3
GMMflow (ours)	3.04	9.3

Table 3: Man to Woman FID comparison

A→C	FID	ALAE-BW ₂
LSB	6.62	31.00
LSBM	6.61	30.99
GMMflow (ours)	3.50	8.54

Table 4: Adult to Child FID comparison

class, given the almost 40% better FID scores and 65% better ALAE-BW₂ scores provided in tables 3, 4. We believe that this significant performance increase is due to the fact that our algorithm leverages the Expectation Maximization algorithm for the GMM pre-fitting, which is less prone to converge to locally optimal values, compared to LightSB’s Maximum likelihood objective, or the LSBM’s bridge matching objective. We further remark that our approach takes 63% less time to train, as noted in Table 5, Appendix B.3.

6 Conclusion

This paper introduces a novel, efficient method for solving the Schrödinger Bridge problem with Gaussian Mixture Model boundary distributions by utilizing a mixture of conditional policies, each solving a Gaussian bridge subproblem. Unlike traditional methods that rely on computationally expensive training of neural networks, which approximately solve high-dimensional non-convex optimization problems, our approach approximates the optimal policy by solving a low-dimensional linear program exactly. Furthermore, the method extends naturally to more general classes of dynamical systems, such as controllable Linear Time-Varying (LTV) systems. Through various experiments, including image-to-image translation tasks, we demonstrate that our outperforms state-of-the-art methods for solving the same task, while requiring minimal computational effort (i.e., no training), highlighting its potential for practical applications in diverse fields, from generative modeling to control theory.

Acknowledgements

Support for this work has been provided by ONR award N00014-18-1-2828 and NASA ULI award #80NSSC20M0163. This article solely reflects the opinions and conclusions of its authors and not of any NASA entity. George Rapakoulias acknowledges financial support from the A. Onassis Foundation Scholarship.

Bibliography

- Michael Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, Kigali Rwanda, May 2023.
- Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, Sydney, Australia, 2017. PMLR.
- Efstathios Bakolas. Finite-horizon covariance control for discrete-time stochastic linear systems subject to input constraints. *Automatica*, 91:61–68, 2018.
- Isin Balci and Efstathios Bakolas. Density steering of gaussian mixture models for discrete-time linear systems. *arXiv preprint arXiv:2311.08500*, 2023.
- Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.
- Alain Bensoussan, KCJ Sung, Sheung Chi Phillip Yam, and Siu-Pang Yung. Linear-quadratic mean field games. *Journal of Optimization Theory and Applications*, 169:496–529, 2016.
- Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- Damiano Brigo. The general mixture diffusion sde and its relationship with an uncertain-volatility option model with volatility-asset decorrelation. Available at SSRN 455060, 2002.
- Damiano Brigo, Fabio Mercurio, and Giulio Saratorelli. Lognormal-mixture dynamics under different means, 2002.
- Charlotte Bunne and Gunnar Rätsch. Neural optimal transport predicts perturbation responses at the single-cell level. *Nature Methods*, 2023.
- Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 6511–6528. PMLR, 28–30 Mar 2022.
- Charlotte Bunne, Ya-Ping Hsieh, Marco Cuturi, and Andreas Krause. The schrödinger bridge between gaussian measures has a closed form. In *International Conference on Artificial Intelligence and Statistics*, pp. 5802–5833. PMLR, 2023a.
- Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause,

- and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023b.
- Kenneth F Caluya and Abhishek Halder. Wasserstein proximal algorithms for the Schrödinger bridge problem: Density control with nonlinear drift. *Transactions on Automatic Control*, 67(3):1163–1178, 2021.
- Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*, held virtually, April 2022.
- Yongxin Chen. Density control of interacting agent systems. *Transactions on Automatic Control*, 69(1):246–260, 2024. doi: 10.1109/TAC.2023.3271226.
- Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal steering of a linear stochastic system to a final probability distribution, part I. *Transactions on Automatic Control*, 61(5):1158–1169, 2015a.
- Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal steering of a linear stochastic system to a final probability distribution, part II. *Transactions on Automatic Control*, 61(5):1170–1180, 2015b.
- Yongxin Chen, Tryphon Georgiou, and Michele Pavon. Entropic and displacement interpolation: a computational approach using the hilbert metric. *SIAM Journal on Applied Mathematics*, 76(6):2375–2396, 2016a.
- Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal transport over a linear dynamical system. *Transactions on Automatic Control*, 62(5):2137–2152, 2016b.
- Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Stochastic control liaisons: Richard Sinkhorn meets Gaspard Monge on a Schrödinger bridge. *SIAM Review*, 63(2):249–313, 2021.
- Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied mathematics and Optimization*, 23(1):313–329, 1991.
- Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.
- Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.
- Hans Föllmer. Random fields and diffusion processes. *Lect. Notes Math*, 1362:101–204, 1988.
- Robert Fortet. Résolution d’un système d’équations de m. schrödinger. *Journal de Mathématiques Pures et Appliquées*, 19(1-4):83–105, 1940.
- Nikita Gushchin, Alexander Kolesov, Petr Mokrov, Polina Karpikova, Andrei Spiridonov, Evgeny Burnaev, and Alexander Korotin. Building the bridge of schrödinger: A continuous entropic optimal transport benchmark. *Advances in Neural Information Processing Systems*, 36:18932–18963, 2023.
- Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schrödinger bridge matching. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17100–17122. PMLR, 21–27 Jul 2024.
- Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- 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. *Advances in neural information processing systems*, 30, 2017.
- 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*, pp. 4401–4410, 2019.
- Alexander Korotin, Nikita Gushchin, and Evgeny Burnaev. Light Schrödinger bridge. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=WhZoCLRWYJ>.
- Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems-A*, 34(4):1533–1574, 2014.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, Kigali Rwanda, May 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- Fengjiao Liu and Panagiotis Tsotras. Reachability and controllability analysis of the state covariance for linear stochastic systems. *arXiv preprint arXiv:2406.14740*, 2024.

- Fengjiao Liu, George Rapakoulias, and Panagiotis Tsiotras. Optimal covariance steering for discrete-time linear stochastic systems. *arXiv preprint arXiv:2211.00618*, 2022a.
- Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou. Deep generalized Schrödinger bridge. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9374–9388, Louisiana, LA, 2022b. Curran Associates, Inc.
- Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos A Theodorou, and Ricky TQ Chen. Generalized Schrödinger bridge matching. In *International Conference on Learning Representations*, Vienna, Austria, 2024.
- Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, Kigali Rwanda, May 2023.
- ApS Mosek. Mosek modeling cookbook, 2020.
- Derek Onken, Samy Wu Fung, Xingjian Li, and Lars Ruthotto. OT-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9223–9232, held virtually, 2021.
- Michele Pavon, Giulio Trigila, and Esteban G Tabak. The data-driven schrödinger bridge. *Communications on Pure and Applied Mathematics*, 74(7):1545–1573, 2021.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- Stefano Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2021.
- Stefano Peluchetti. Diffusion bridge mixture transports, schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023.
- Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport: With Applications to Data Science*. Foundations and Trends in Machine Learning, 2019. doi: 10.1561/2200000073.
- Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, Seattle, Washington, June 2020.
- George Rapakoulias and Panagiotis Tsiotras. Discrete-time optimal covariance steering via semidefinite programming. In *62nd Conference on Decision and Control*, pp. 1802–1807, Singapore, 2023. doi: 10.1109/CDC49753.2023.10384118.
- George Rapakoulias and Panagiotis Tsiotras. Discrete-time maximum likelihood neural distribution steering. *arXiv preprint arXiv:2409.02272*, 2024.
- Lars Ruthotto and Eldad Haber. An introduction to deep generative modeling. *GAMM-Mitteilungen*, 44(2):e202100008, 2021.
- Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.
- Augustinos D Saravacos, Yihui Li, and Evangelos A Theodorou. Distributed hierarchical distribution control for very-large-scale clustered multi-agent systems. In *Robotics: Science and Systems XIX*, Daegu, Republic of Korea, July 2023.
- Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.
- Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2023.
- Antonio Terpin, Nicolas Lanzetti, and Florian Dörfler. Dynamic programming in probability spaces via optimal transport. *SIAM Journal on Control and Optimization*, 62(2):1183–1206, 2024. doi: 10.1137/23M1560902.
- Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. TrajectoryNet: A dynamic optimal transport network for modeling cellular dynamics. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9526–9536. PMLR, 13–18 Jul 2020.
- Richard Lee Wheeden and Antoni Zygmund. *Measure and integral*, volume 26. Dekker New York, 1977.

A Proofs

All proofs are carried out for a general Linear Time-Varying (LTV)

$$dx_t = A_t x_t dt + B_t u(x_t) dt + D_t dw, \quad (\text{A.1})$$

where $x_t \in \mathbb{R}^d$, $A_t \in \mathbb{R}^{d \times d}$, $u_t \in \mathbb{R}^m$, $B_t \in \mathbb{R}^{d \times m}$, $D_t \in \mathbb{R}^{d \times q}$ and dw is the q -dimensional Brownian increment having the properties $\mathbb{E}[dw] = \mathbb{E}[dw dt] = 0$ and $\mathbb{E}[dw^\top dw] = dt$. The dynamical system (12b) is just a special case of (A.1) with $A_t = 0$, $B_t = I$, $D_t = \sqrt{\epsilon} I$.

A.1 The Fokker-Plank-Kolmogorov equation

The equation describing the propagation of the distribution of the state of the dynamical system (A.1), known as the Fokker-Plank-Kolmogorov (FPK) equation (Särkkä & Solin, 2019) is:

$$\frac{\partial \rho_t}{\partial t} + \sum_i \frac{\partial}{\partial x_i} (\rho_t (A_t x_t + B_t u_t(x_t))) - \frac{1}{2} \sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} ([D_t D_t^\top]_{ij} \rho_t(x_t)) = 0. \quad (\text{A.2})$$

This equation can be written more compactly using standard vector notation as follows

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t (A_t x_t + B_t u_t(x_t))) - \frac{1}{2} \text{tr} (D_t D_t^\top \nabla^2(\rho_t)) = 0, \quad (\text{A.3})$$

where $\nabla^2(\rho_t)$ denotes the Hessian of the density with respect to the state x_t at time t . In the specific case where $D_t = \sqrt{\epsilon} I$, equation (A.3) reduces to the well-known equation

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t (A_t + B_t u_t)) - \frac{\epsilon}{2} \Delta \rho_t(x) = 0, \quad (\text{A.4})$$

where Δ denotes the Laplacian operator.

A.2 Proof of Theorem 2

First, notice that the probability flow (14) respects the constraint (12c), (12d) for all feasible values of λ_{ij} , since

$$\rho_0 = \sum_{i,j} \rho_{0|ij} \lambda_{ij} = \sum_{i,j} \mathcal{N}(\mu_0^i, \Sigma_0^i) \lambda_{ij} = \sum_i \mathcal{N}(\mu_0^i, \Sigma_0^i) w_i, \quad (\text{A.5a})$$

$$\rho_1 = \sum_{i,j} \rho_{1|ij} \lambda_{ij} = \sum_{i,j} \mathcal{N}(\mu_1^j, \Sigma_1^i) \lambda_{ij} = \sum_j \mathcal{N}(\mu_1^j, \Sigma_1^i) w_j. \quad (\text{A.5b})$$

Therefore, it suffices to show that the policy (13) produces the probability flow (14). Following the approach of Lipman et al. (2023) and Liu et al. (2024), we show that the pair (13), (14) satisfies the FPK equation. We start from the FPK equation describing a conditional flow and sum over all conditional variables to retrieve the unconditional flow. Specifically, given that the individual policies $u_{t|ij}$ solve the Gaussian Bridge subproblems

(7), the pair $\rho_{t|ij}, u_{t|ij}$ satisfies the FPK equation for the dynamical system (A.1), that is,

$$\frac{\partial \rho_{t|ij}}{\partial t} + \nabla \cdot (\rho_{t|ij}(A_t x_t + B_t u_{t|ij})) - \frac{1}{2} \text{tr}(D_t D_t^\top \nabla^2(\rho_{t|ij})) = 0, \quad (\text{A.6a})$$

Multiplying with λ_{ij} and summing, we obtain

$$\sum_{i,j} \lambda_{ij} \left[\frac{\partial \rho_{t|ij}}{\partial t} + \nabla \cdot (\rho_{t|ij}(A_t x_t + B_t u_{t|ij})) - \frac{1}{2} \text{tr}(D_t D_t^\top \nabla^2(\rho_{t|ij})) \right] = 0, \quad (\text{A.6b})$$

$$\frac{\partial}{\partial t} \left(\sum_{i,j} \rho_{t|ij} \lambda_{ij} \right) + \nabla \cdot \left(A_t x_t \sum_{i,j} \rho_{t|ij} \lambda_{ij} + B_t \sum_{i,j} u_{t|ij} \rho_{t|ij} \lambda_{ij} \right) - \frac{1}{2} \text{tr} \left(D_t D_t^\top \nabla^2 \left(\sum_{i,j} \rho_{t|ij} \lambda_{ij} \right) \right) = 0. \quad (\text{A.6c})$$

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot \left(\rho_t \left(A_t x_t + B_t \sum_{ij} u_{t|ij} \frac{\rho_{t|ij} \lambda_{ij}}{\sum_{i,j} \rho_{t|ij} \lambda_{ij}} \right) \right) - \frac{1}{2} \text{tr}(D_t D_t^\top \nabla^2(\rho_t)) = 0, \quad (\text{A.6d})$$

and finally,

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t(A_t x_t + B_t u_t)) - \frac{1}{2} \text{tr}(D_t D_t^\top \nabla^2(\rho_t)) = 0. \quad (\text{A.6e})$$

A.3 Proof of Theorem 3

It suffices to show that the cost (12a) is upper bounded by the cost of (16a). Substituting policy (13) to the cost (12a) we obtain

$$J_{\text{GMM}} = \mathbb{E}_{x_t \sim \rho_t} \left[\int_0^1 \left\| \sum_{i,j} u_{t|ij}(x) \frac{\rho_{t|ij}(x) \lambda_{ij}}{\sum_{i,j} \rho_{t|ij}(x) \lambda_{ij}} \right\|^2 dt \right] \quad (\text{A.7a})$$

$$= \int_0^1 \int \rho_t \left\| \sum_{i,j} u_{t|ij}(x) \frac{\rho_{t|ij}(x) \lambda_{ij}}{\sum_{i,j} \rho_{t|ij}(x) \lambda_{ij}} \right\|^2 dx dt \quad (\text{A.7b})$$

$$\leq \int_0^1 \int \rho_t \frac{\sum_{i,j} \|u_{t|ij}(x)\|^2 \rho_{t|ij}(x) \lambda_{ij}}{\sum_{i,j} \rho_{t|ij}(x) \lambda_{ij}} x dt \quad (\text{A.7c})$$

$$= \int_0^1 \int \sum_{i,j} \|u_{t|ij}(x)\|^2 \rho_{t|ij}(x) \lambda_{ij} dx dt \quad (\text{A.7d})$$

$$= \sum_{i,j} \lambda_{ij} \mathbb{E}_{x \sim \rho_{t|ij}} \left[\int_0^1 \|u_{t|ij}(x)\|^2 dt \right] \quad (\text{A.7e})$$

$$= \sum_{i,j} \lambda_{ij} J_{ij} = J_{\text{OT}} \quad (\text{A.7f})$$

where (A.7b) is due to Fubini's theorem (Wheeden & Zygmund, 1977, Theorem 6.1) and (A.7c) makes use of the discrete version of Jensen's inequality (Wheeden & Zygmund, 1977, Theorem 7.35).

B Experiment Details

B.1 2D Problems

To compare our approach with state-of-the-art Neural SB Solvers we used the original implementations of the DSB¹ (De Bortoli et al., 2021) and DSBM² (Shi et al., 2023) algorithms and report the results in Table (2). The network architecture used for both algorithms is the fully connected DNN of De Bortoli et al. (2021) with 128-dimensional sinusoidal temporal encodings, 256 neurons in the encoder layer, {256, 256} neurons in the decoder layers and SiLU activation functions (Hendrycks & Gimpel, 2016).

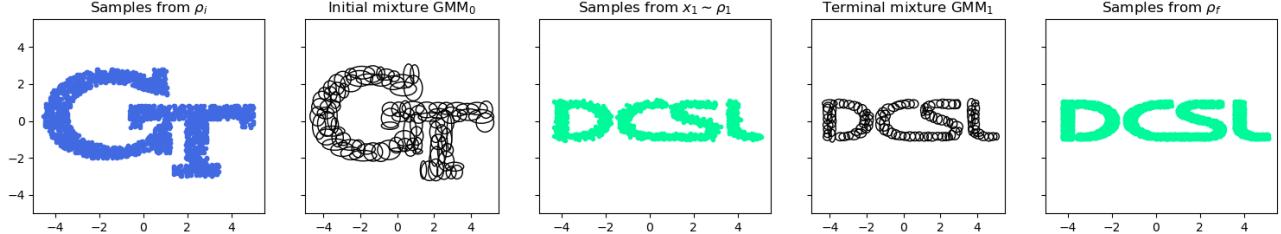


Figure 7: GT to DCSL distribution steering details.

B.2 Problems with LTI prior dynamics

To solve the Gaussian Bridge sub-problems with a dynamical system of the form (17) we use the discrete-time convex formulation of Rapakoulias & Tsiotras (2023). We also include a brief overview of the method in Appendix C. Each continuous-time Gaussian Bridge is discretized (in the temporal dimension) into 101 steps over uniform intervals of size $\Delta t = 0.01$. We used MOSEK (Mosek, 2020) to solve the resulting semidefinite programs.

B.3 Image-to-Image translation

To better evaluate the performance of the proposed approach in the task of Image-to-Image translation, we provide further examples in Figures 8, 9. We also provide approximate training and inference times for our approach for the two examples in subsection 5.4 and compare them with the training and inference times of LightSB in Table 5. For our approach, training time consists of the time required to fit the GMMs in the encodings of the FFHQ dataset for the two boundary distributions, and the solution of the linear program (16). As inference time, we consider the time taken for the integration of the SDE (or ODE for $\epsilon = 0$) (12b) with the mixture policy (13). We observe that while inference time is small for solving the deterministic (optimal transport) problem, i.e. for $\epsilon = 0$, integrating the stochastic dynamical system for positive values of ϵ requires more time due the small time step required for SDE integration. The quality of the produced images was not found to be affected by this parameter, implying that $\epsilon = 0$ could be used for fast, deterministic inference, while a positive value of ϵ will allow for some randomness in the generated images. We also note that the faster training time for our approach is mainly due to the very fast convergence of the EM algorithm, which is also less likely to converge to local minima, compared to the standard maximum likelihood method for fitting distribution to data. All tests were conducted on a desktop computer with RTX 3070 GPU.

	Training [s]	Inference ($\epsilon = 0$) [s],	Inference ($\epsilon = 0.1$) [s]
LightSB ³	57	-	0.02
Ours	21	0.7	6.5

Table 5: Training and inference time comparison with state of the art. Inference time is measured for a batch of 10 images.

¹https://github.com/JTT94/diffusion_schrodinger_bridge

²<https://github.com/yuyang-shi/dsbm-pytorch>

³Korotin et al. (2024)

C Gaussian Bridge for Linear Time-Varying systems

In this section, we briefly review the methods available in the literature to solve the Gaussian Bridge problem with general LTV dynamics of the form (A.1). That is, we consider the problem

$$\min_{u \in \mathcal{U}} \mathbb{E} \left[\int_0^1 \|u_t(x)\|^2 dt \right], \quad (\text{C.1a})$$

$$dx_t = A_t x_t dt + B_t u(x_t) dt + D_t dw, \quad (\text{C.1b})$$

$$x_0 \sim \mathcal{N}(\mu_i, \Sigma_i), \quad x_1 \sim \mathcal{N}(\mu_f, \Sigma_f). \quad (\text{C.1c})$$

The solution of problem (C.1) is used to solve the Gaussian Bridge problem for the example in Section 5.2 and is relevant to applications with prior dynamics of more general structure such as mean field games (Bensoussan et al., 2016) and large multi-agent control applications (Saravacos et al., 2023). The existence and uniqueness of solutions for problem (C.1) are studied in (Chen et al., 2015a; Liu et al., 2022a; Liu & Tsiotras, 2024). Since the state of (C.1b) remains Gaussian throughout the steering horizon, i.e., $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$, the problem simplifies to that of the control of the first two statistical moments of the state, namely the mean μ_t and the covariance Σ_t . Using a control policy parametrization of the form

$$u_t(x) = K_t(x - \mu_t) + v_t, \quad (\text{C.2})$$

allows for the decoupling of the propagation equations for the mean and covariance of the state. More specifically, applying (C.2) to (C.1b), the equations describing the propagation of μ_t and Σ_t yield

$$\dot{\Sigma}_t = (A_t + B_t K_t) \Sigma_t + \Sigma_t (A_t + B_t K_t)^\top + D_t D_t^\top, \quad (\text{C.3a})$$

$$\dot{\mu}_t = A_t \mu_t + B_t v_t. \quad (\text{C.3b})$$

Expanding the expression (C.3a) and performing the change of variables $U_t = \Sigma_t K_t$, we obtain

$$\dot{\Sigma}_t = A_t \Sigma_t A_t^\top + B_t U_t + U_t^\top B_t^\top + D_t D_t^\top, \quad (\text{C.4})$$

which is linear in U_t, Σ_t . Furthermore, substituting (C.2) to the cost function (C.1a), it simplifies to

$$\mathbb{E} \left[\int_0^1 \|u_t(x)\|^2 dt \right] = \int_0^1 v_t^\top v_t + \text{tr}(K_t \Sigma_t K_t^\top) dt = \int_0^1 v_t^\top v_t + \text{tr}(U_t \Sigma_t^{-1} U_t^\top) dt. \quad (\text{C.5})$$

Equations (C.3b), (C.4), (C.5) can be used to reformulate problem (C.1) to a simpler optimization problem in the space of affine feedback policies, parameterized by U_t, v_t . To be more precise, problem (C.1) reduces to

$$\min_{\mu_t, v_t, \Sigma_t, U_t} \int_0^1 v_t^\top v_t + \text{tr}(U_t \Sigma_t^{-1} U_t^\top) dt, \quad (\text{C.6a})$$

$$\dot{\Sigma}_t = A_t \Sigma_t A_t^\top + B_t U_t + U_t^\top B_t^\top + D_t D_t^\top, \quad (\text{C.6b})$$

$$\dot{\mu}_t = A_t \mu_t + B_t v_t, \quad (\text{C.6c})$$

$$\mu_0 = \mu_i, \Sigma_0 = \Sigma_i, \mu_1 = \mu_f, \Sigma_1 = \Sigma_f. \quad (\text{C.6d})$$

which can be further relaxed to a convex semi-definite program using the lossless convex relaxation Chen et al. (2015b)

$$\min_{\mu_t, v_t, \Sigma_t, U_t} \int_0^1 v_t^\top v_t + \text{tr}(Y_t^\top) dt, \quad (\text{C.7a})$$

$$U_t \Sigma_t^{-1} U_t \preceq Y_t, \quad (\text{C.7b})$$

$$\dot{\Sigma}_t = A_t \Sigma_t A_t^\top + B_t U_t + U_t^\top B_t^\top + D_t D_t^\top, \quad (\text{C.7c})$$

$$\dot{\mu}_t = A_t \mu_t + B_t v_t. \quad (\text{C.7d})$$

$$\mu_0 = \mu_i, \Sigma_0 = \Sigma_i, \mu_1 = \mu_f, \Sigma_1 = \Sigma_f, \quad (\text{C.7e})$$

after noting that the constraint (C.7b) can be cast as a Linear Matrix Inequality (LMI) using Schur's complement as

$$\begin{bmatrix} \Sigma_t & U_t^\top \\ U_t & Y_t \end{bmatrix} \succeq 0.$$

Problem (C.7) is still infinite dimensional since the decision variables are functions of time $t \in [0, 1]$, however, it can be discretized, approximately using a first-order approximation of the derivatives in (C.7c), (C.7d) (Chen et al., 2015b) or exactly using a zero-order hold (Liu et al., 2022a; Rapakoulias & Tsiotras, 2023), and solved to global optimality using a semidefinite programming solver such as MOSEK (Mosek, 2020).

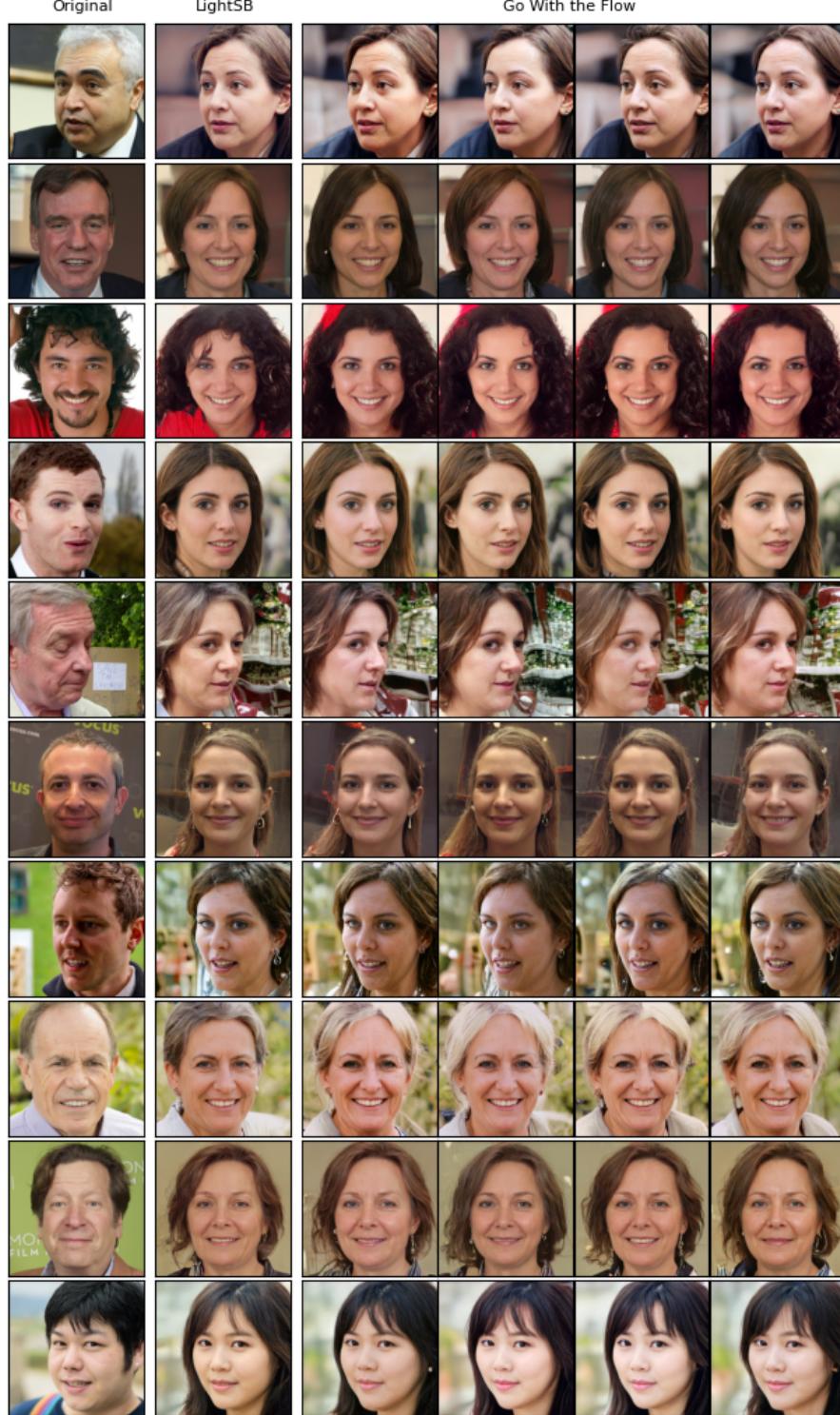


Figure 8: Further examples for the man-to-woman Image-to-Image translation task.

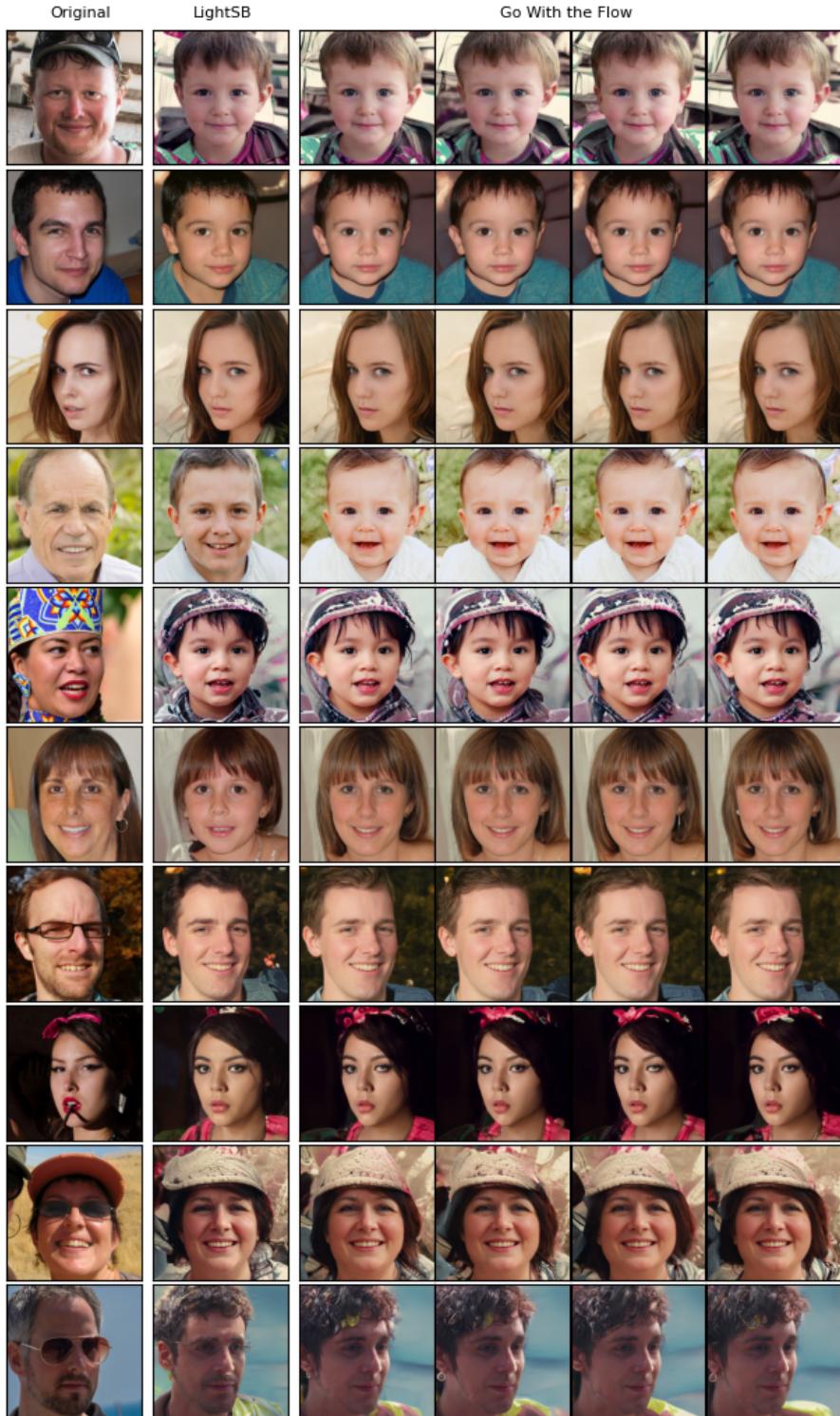


Figure 9: Further examples for the adult-to-child Image-to-Image translation task.