Stochastic Lifting for One-Step Per-Frame Video Generation and Physics Simulations

Jules Berman
Courant Institute of
Mathematical Sciences
New York University
New York, NY 10012
jmb1174@nyu.edu

Tobias Blickhan
Courant Institute of
Mathematical Sciences
New York University
New York, NY 10012
tobias.blickhan@nyu.edu

Benjamin Peherstorfer
Courant Institute of Mathematical Sciences
New York University
New York, NY 10012
pehersto@cims.nyu.edu

Abstract

We introduce Stochastic Lifting, which challenges the prevailing view that accurately generating video and stochastic physics simulations requires some form of dynamic or optimal transport, which is inherently slow. Stochastic Lifting augments training data samples with high-dimensional stochastic labels and then learns a transition map from the current frame and label to the next frame via a simple regression problem. Evaluating the regression map on a newly drawn label generates a sample, which results in one-step per-frame inference. The error of samples generated with Stochastic Lifting can be bounded in the (empirical) Wasserstein-2 metric if the regression map interpolates and is smooth. Stochastic Lifting achieves state-of-the-art accuracy on simulating trajectories of stochastic physics systems and video generation benchmarks among one-step methods. We also demonstrate the scalability and low inference costs by generating 32 frames of 480x480 videos in pixel space in under one second on a single H100 GPU.

1 Introduction

Generative models and rearrangement Many state-of-the-art generative models are based on dynamic transport, which means that they learn a multi-step transformation of a reference distribution (often noise) to data [58, 24, 29, 60, 3, 37]. These models do not require meaningfully paired data from reference and data distributions but learn such a coupling throughout training via a sequence of regression problems over an artificial sampling time. We refer to this as a rearrangement.

After rearrangement, learning a regular map that moves samples from source to target is a regression problem, e.g. in the distillation of a trained multi-step model [55, 4, 14, 65]. In other words, rearrangement turns the unsupervised problem of generating data into a supervised problem of fitting coupled data points between reference and target. Approximately computing a meaningful coupling has been shown to improve the training of multi-step methods [47].

One-step methods Multi-step methods can lead to high inference costs per sample, which quickly accumulate to prohibitive costs when generating sequential data, such as trajectories of physics

Preprint. Under review.

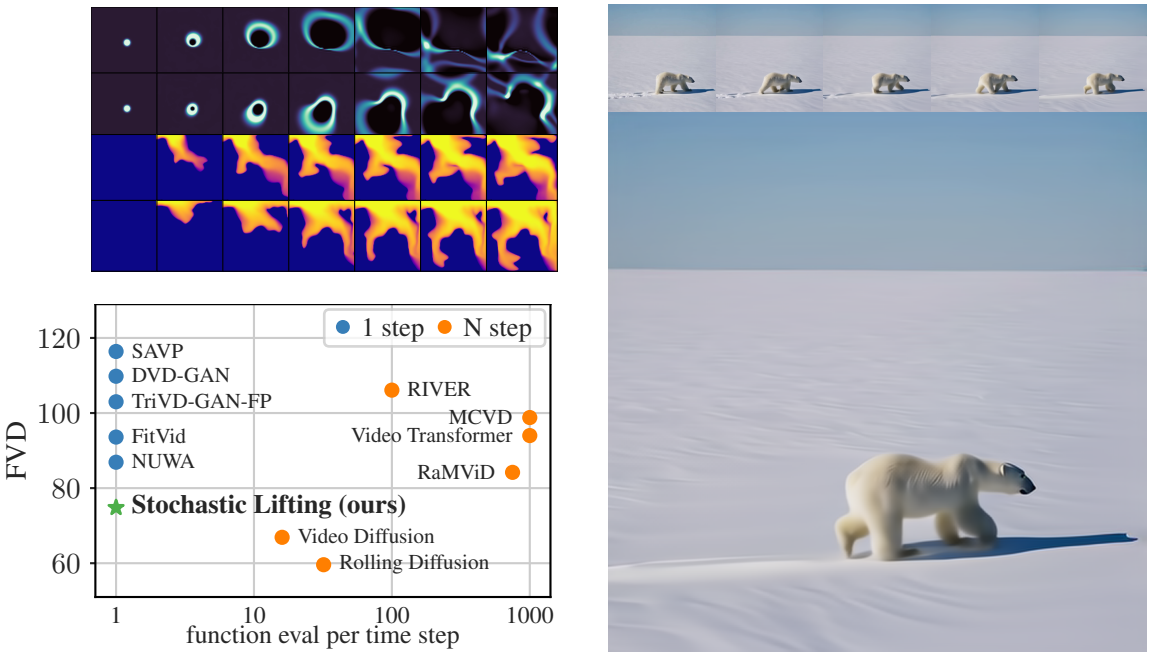


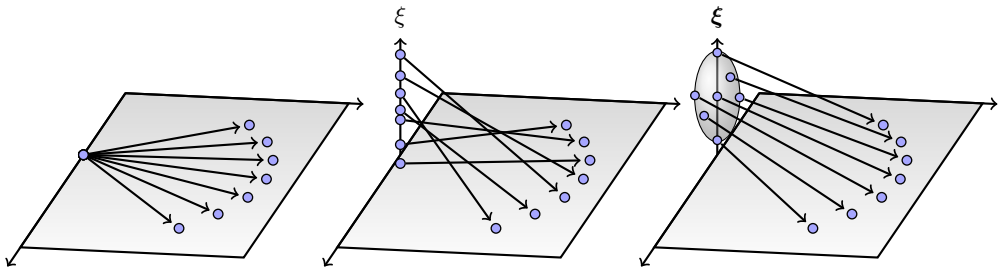
Figure 1: **Left Bottom**: On the BAIR video generation benchmark, Stochastic Lifting is state-of-the-art among one-step methods and competitive with diffusion-based methods that use over one order of magnitude more compute. **Left Top**: Stochastic Lifting produces accurate and diverse samples for stochastic physics simulations. **Right**: Stochastic Lifting generates 32 frames of 480×480 color video directly in pixel space in just 0.96 seconds.

systems and videos. One-step methods have been proposed to reduce inference costs. They have proven to be challenging to train and have been mostly demonstrated on static image generation tasks so far [70, 57, 72, 59, 20, 19, 71].

Stochastic Lifting: regressing on randomly labeled points plus lifting to higher dimensions for smoothness is sufficient for next-frame generation *We show that for the specific task of next-frame sample generation, the conditioning on the previous frame provides a strong enough coupling that no explicit rearrangement is needed to learn a regression map for generating new samples.* In fact, we show that a map that interpolates the training data and that is sufficiently smooth accurately generates new samples with respect to the Wasserstein-2 metric.

By introducing a random label to each pair of current and next frames in the training data, we separate the data points well enough to learn an interpolation function. We show that we have some control over the regularity of this function (in terms of its Lipschitz constant) via the dimension of the labels. Additionally, because subsequent frames are similar, the conditioning on the previous frame already introduces a strong coupling prior. Putting all of this together (lifting, interpolation, conditioning on previous frame) allows Stochastic Lifting to skip the rearrangement step and still find a smooth regression function. Once a regression function has been trained, it can be evaluated at a newly drawn stochastic label to generate a new sample with just one neural-network function evaluation.

We stress again that the conditioning on the previous frame is critical because it means the (source) distribution corresponding to the current frame is close in some meaningful sense to the (target) distribution at the next frame. In particular, our approach fails to generate images from noise samples, but Stochastic Lifting achieves state-of-the-art accuracy on video generation benchmarks such as one-step BAIR and simulating stochastic physical systems. It does so using only a single neural-network function evaluation per frame and without requiring pre-trained generative models for distillation.



(a) one-to-many issue (b) poorly conditioned (c) Stochastic Lifting

Figure 2: **(a)** For data from a stochastic process, (almost) identical frames can have different next frames, which prevents learning a transition map from just trajectories. **(b)** Labeling data points separates them but low-dimensional labels can still lead to poorly conditioned regression problems. **(c)** Stochastic Lifting uses high-dimensional labels, which enables fitting a smooth regression function.

1.1 Setup and problem formulation

Consider a stochastic process $\{\boldsymbol{X}_t\}_t \subset \mathcal{X} = [0,1]^n$ with initial condition $\boldsymbol{X}_0 \sim \rho_0$. We will call a realization $\boldsymbol{x}_t$ of $\boldsymbol{X}_t$ a frame. The frames evolve via a conditional distribution

$$\boldsymbol{X}_{t+1}|\boldsymbol{X}_t = \boldsymbol{x}_t \sim \rho(\,\cdot\,|\boldsymbol{x}_t)\,. \tag{1}$$

To ease exposition, we focus on transitions that condition on the previous frame $\boldsymbol{X}_t = \boldsymbol{x}_t$ only, but all of the following extends to transitions that condition on a history of more than one previous frame. By the law of total probability, we define the time marginals as $\rho_{t+1} = \int_{\mathcal{X}} \rho(\cdot|\boldsymbol{x}_t)\rho_t(\boldsymbol{x}_t)\,\mathrm{d}\boldsymbol{x}_t$ with the assumption that we can directly sample from ρ_0. Notice that the time marginals ρ_t can change with time while the conditional distribution $\rho(\cdot|\boldsymbol{x}_t)$ is independent of time and only depends on the previous frame $\boldsymbol{x}_t$. The coupling between $\boldsymbol{x}_t, \boldsymbol{x}_{t+1}$ is described by the density π_t over $\mathbb{R}^n \times \mathbb{R}^n$ as $\pi_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) = \rho(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)\rho_t(\boldsymbol{x}_t)$. Our data consists of pairs of realizations

$$\mathcal{D} = \bigcup_{i=1}^{M} \bigcup_{t=0}^{T-1} \left\{(\boldsymbol{x}_t^i, \boldsymbol{x}_{t+1}^i)\right\} \tag{2}$$

with $(\boldsymbol{x}_t^i, \boldsymbol{x}_{t+1}^i) \sim \pi_t$ for all $i = 1, \ldots, M$. The goal is to learn a map F that mimics the transition (1). Once we have such a map, we can roll it out autoregressively to rapidly predict new trajectories for a given sample from ρ_t at any time t.

1.2 Challenges of learning one-step transition functions of stochastic processes

Stochastic dynamics lead to one-to-many maps We face challenges that are unique to generating trajectory data (videos, simulations of physics systems) coming from stochastic dynamics. One challenge is that there cannot exist a function $f : \mathcal{X} \to \mathcal{X}, \boldsymbol{x}_t \mapsto \boldsymbol{x}_{t+1}$ that describes the transition (1) of the stochastic process $\{\boldsymbol{X}_t\}_t$ over more than one time step. The reason is that, when $\rho(\cdot|\boldsymbol{x}_t)$ does not collapse to a single point, the transition from $\boldsymbol{x}_t$ to $\boldsymbol{X}_{t+1}|\boldsymbol{X}_t = \boldsymbol{x}_t$ can be interpreted as a one-to-many map, see Figure 2a. There are arbitrarily many different realizations $\boldsymbol{x}_{t+1}^j, j = 1, 2, 3, \ldots$ of $\boldsymbol{X}_{t+1}|\boldsymbol{X}_t = \boldsymbol{x}_t$ at time $t+1$ and a function f can map from $\boldsymbol{x}_t$ to only one of them. Even if we consider two different but very close $\boldsymbol{x}_t$ and $\boldsymbol{x}_t'$, a similar issue can arise when the corresponding $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}_{t+1}'$ are far apart. One can interpret this as a poorly conditioned regression problem, which requires a regression function (and a parametrization that can represent it) that is far from smooth. In particular, the Lipschitz constant of f grows to infeasible values in this setting.

One interpretation of the one-to-many map perspective is that the transition from $\boldsymbol{x}_t$ to a realization of $\boldsymbol{X}_{t+1}\,|\,\boldsymbol{X}_t = \boldsymbol{x}_t$ depends on additional information that is not present in $\boldsymbol{x}_t$. For example, let us consider a transition (1) stemming from a discretized stochastic differential equation $\boldsymbol{X}_{t+1} = b(\boldsymbol{X}_t) + \sigma\boldsymbol{W}_t$. The transition from $\boldsymbol{X}_t$ to $\boldsymbol{X}_{t+1}$ depends on the realization of $\boldsymbol{X}_t$ but additionally on the realization of the noise $\boldsymbol{W}_t$, which is typically independent of $\boldsymbol{X}_t$. Thus, $\boldsymbol{x}_t$ alone is insufficient to describe $\boldsymbol{X}_{t+1}|\boldsymbol{X}_t = \boldsymbol{x}_t$.

Fitting a function to stochastic dynamics collapses to the mean dynamics Simply fitting a function $f : \mathcal{X} \to \mathcal{X}, \boldsymbol{x}_t \mapsto \boldsymbol{x}_{t+1}$ to data (2) from a stochastic process can capture only mean-like

behavior. In fact, training f on data (2) with the mean-squared error loss collapses to the conditional expectation $\boldsymbol{x}_t \mapsto \mathbb{E}[\boldsymbol{X}_{t+1}|\boldsymbol{X}_t = \boldsymbol{x}_t]$ in the limit $M \to \infty$.

If the stochasticity in the data (2) is due to noise and perturbations, then often the goal is recovering the mean behavior, in which case fitting time integrators with the mean-squared loss is reasonable; as in operator learning [36, 39, 33]. However, our goal is generating new sample trajectories where the randomness is not only injected at the initial condition sampled from ρ_0 but also during the roll out, as described by the transition (1) and reflected in the data (2) (recall the example with the stochastic differential equation in the previous paragraph). Thus, capturing mean-like behavior only by applying operator learning and other deterministic transition modeling methods and relying on input randomness alone is insufficient. In particular, we will consider stochastic processes that always start with the same initial condition $\rho_0 = \delta_{\boldsymbol{x}_0}$ (see wave and multi-phase flow in Section 3).

1.3 Literature review

Multi- and single-step generative models There is a range of multi-step methods for generative modeling, which build on diffusion- and flow-based modeling [58, 24, 29, 61, 60, 62, 3, 2, 37, 38]. The focus of this work lies on one-step methods, for which a single neural-network function evaluation per sample is sufficient. Early one-step methods are difficult to train [21, 5] or easily collapse [31]. There are distillation approaches for constructing one-step models out of multi-step models but these require having readily available a multi-step model [70, 57, 72, 59, 20, 19]. More recently, there are methods that aim to circumvent having to train a multi-step model first [59, 71] but they focus on static generation tasks such as noise to image and cannot explicitly use conditioning on previous frames that we rely on for next-frame generation.

Autoregressive generative models for trajectory data generation There is a range of works on autoregressive diffusion models (ARDMs) [26, 2, 48] and also flow-based models [15, 11] that can generate trajectory data; however, these require multiple steps per frame, which can be expensive. There are also time-space approaches [25] with the drawback of having to handle a large time-space tensor. We also mention marginal trajectory matching [44, 9] which also performs only a single neural-network function evaluation per time step but the corresponding models can be challenging to scale to high dimensions. Another line of work imposes constraints on the diffusion trajectories to better align them [54, 53]; these again require multiple steps per frame.

Lifting A key aspect of our approach is lifting data into higher dimensions, which is prevalent in machine learning [13, 50] but also in computational science in general [41, 22, 34, 49, 42]. We mention neural ordinary differential equations [16, 30] that augment the state space to extend the expressivity of neural ordinary differential equations. With lifting we pursue an analogous goal of making a problem better behaved, namely smoother.

1.4 Contributions

(1) **Fast inference** – one forward pass per frame, generating videos with 32 frames of size 480×480 in pixel space in 0.96 seconds on one H100 GPU.

(2) **Simple algorithm** – no latent encoder/decoder, no distillation, just plain UNet backbone; straightforward regression loss.

(3) **Scalable** – state-of-the-art on BAIR; stable 500-frame rollouts on CLEVRER; scales to 480×480 resolution videos.

(4) **Theory** – show Wasserstein-2 error bound when Stochastic Lifting map is smooth and interpolates data.

We release our datasets Wave, Flow, and Polar and an implementation of our method at `redacted`.

2 Stochastic lifting

2.1 Labeling data points to avoid one-to-many issue

We randomly draw labels $\boldsymbol{\xi}_t^i \in \mathbb{R}^d$ from a distribution ν for all $\boldsymbol{x}_t^i$ in the data set $\mathcal{D}$, which leads to the augmented data set

$$\mathcal{D}_\xi = \bigcup_{i=1}^{M} \bigcup_{t=0}^{T-1} \left\{ \left(\boldsymbol{x}_t^i, \boldsymbol{x}_{t+1}^i, \boldsymbol{\xi}_t^i \right) \right\} .$$

In all of the following, the label distribution ν will be $\mathcal{N}(0, \boldsymbol{I}_d)$, the standard normal distribution of dimension d, which implies that the labels are unique almost surely. We assume that ν is independent of t, but we still denote $\boldsymbol{\xi}_t \sim \nu$ to emphasize that, for a given trajectory $\boldsymbol{x}_0^i, \ldots, \boldsymbol{x}_{T-1}^i$, the label of the frame at time t need not be the same as the label of the frame at time $t+1$.

Uniquely labeling the data points overcomes the one-to-many issue: Even if there are $\boldsymbol{x}_t^i = \boldsymbol{x}_t^j$ for $i \neq j$, the corresponding labeled points $(\boldsymbol{x}_t^i, \boldsymbol{\xi}_t^i)$ and $(\boldsymbol{x}_t^j, \boldsymbol{\xi}_t^j)$ are unique; see Figure 2b. Thus, circling back to the challenges described in Section 1.2, because of the unique label for each data point, there now exists a function $F : \mathcal{X} \times \mathbb{R}^d \to \mathcal{X}$ that can interpolate (memorize) the data in the sense

$$F(\boldsymbol{x}_t^i, \boldsymbol{\xi}_t^i) = \boldsymbol{x}_{t+1}^i , \qquad i = 1, \ldots, M, \quad t = 0, \ldots, T-1 . \tag{3}$$

Recalling the example from Section 1.2 about the stochastic differential equation, we can interpret $\boldsymbol{\xi}_t^i$ as a surrogate of the realization of noise $\mathrm{d}\boldsymbol{W}_t$ that enters the dynamics.

2.2 Stochastic labeling, strong frame conditioning, and push-forward maps

We now show that if the map F is regular in terms of its Lipschitz constant and interpolates the data, then samples generated with F are close to ρ_{t+1} in the Wasserstein-2 metric.

Proposition 1. *Let F interpolate the training data D_ξ as in* (3)*. At any time t, consider $\tilde{M}$ test samples $\mathcal{D}_\xi^{test} = \{(\tilde{\boldsymbol{x}}_t^i, \tilde{\boldsymbol{x}}_{t+1}^i, \tilde{\boldsymbol{\xi}}_t^i)\}_{i=1}^{\tilde{M}}$ sampled from $\pi_t \otimes \nu$. Evaluate now the map F to obtain the generated samples $\hat{\boldsymbol{x}}_{t+1}^i = F(\tilde{\boldsymbol{x}}_t^i, \tilde{\boldsymbol{\xi}}_t^i)$ for $i = 1, \ldots, \tilde{M}$. Then,*

$$\mathbb{E}_{\mathcal{D}_\xi, \mathcal{D}_\xi^{test}} \left[W_2(\hat{\rho}_{t+1}, \tilde{\rho}_{t+1})^2 \right] \leq C(1 + 2L_F^2) \min(M, \tilde{M})^{-2/\alpha} , \tag{4}$$

where $\hat{\rho}_{t+1}$ and $\tilde{\rho}_{t+1}$ are the empirical measures corresponding to the generated samples $\{\hat{\boldsymbol{x}}_{t+1}^i\}_{i=1}^{\tilde{M}}$ and the test samples $\{\tilde{\boldsymbol{x}}_{t+1}^i\}_{i=1}^{\tilde{M}}$, $C > 0$ is a constant, L_F is the Lipschitz constant of F, and $\alpha \geq 0$ depends on the intrinsic dimensions of ρ_t, ρ_{t+1} and ν, whichever is largest.

A proof can be found in Appendix A.2. The bound (4) critically depends on the smoothness of F in terms of the Lipschitz constant L_F. While simply interpolating independently drawn samples from two distributions can lead to regression functions with high Lipschitz constants in low dimensions (see Figure 2b), we will discuss in the following that labels of high dimension can help to reduce L_F.

Furthermore, a strong conditioning induced by the previous frame can be helpful. For next-frame generation, we expect frames to gradually change over time, hence $\|\boldsymbol{x}_{t+1}^i - \boldsymbol{x}_t^i\| = \mathcal{O}(\mathrm{d}t)$ scales with a small time-step size $\mathrm{d}t$ for all times and frames. The following corollary shows that if the next frame depends on the current frame and a stochastic update term $R(\boldsymbol{x}, \boldsymbol{\xi})$ scaled by the time-step size $\mathrm{d}t$, then the scaling $\mathrm{d}t$ enters in front of the Lipschitz constant to help alleviate non-smoothness.

Corollary 1. *If $F(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{x} + \mathrm{d}t R(\boldsymbol{x}, \boldsymbol{\xi})$, then* (4) *can be written as*

$$\mathbb{E}_{\mathcal{D}_\xi, \mathcal{D}_\xi^{test}} \left[W_2(\hat{\rho}_{t+1}, \tilde{\rho}_{t+1})^2 \right] \leq C(3 + 4\mathrm{d}t^2 L_R^2) \min(M, \tilde{M})^{-2/\alpha} ,$$

where C and α are as in Proposition 1 and L_R is the Lipschitz constant of R.

Similar bounds can be derived when assuming more structure on F such as $F(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{x} + \mathrm{d}t\, b(\boldsymbol{x}) + \varepsilon\sigma(\boldsymbol{\xi})$; see Appendix A.2 and compare to the discrete stochastic differential equation in Section 1.2.

2.3 Lifting for smoothness

The key in the previous results was the smoothness of F given by the Lipschitz constant, which we can control by using high-dimensional labels. The minimal Lipschitz constant for a function F that

Algorithm 1 Train $F_{\boldsymbol{\theta}}$ with stochastic lifting

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_{\text{init}}$
$\mathcal{D}_\xi \leftarrow \{(\boldsymbol{x}_t^i, \boldsymbol{x}_{t+1}^i, \boldsymbol{\xi}_t^i) \,:\, \boldsymbol{\xi}_t^i \sim \nu\}_{i=1,\,t=0}^{i=M,\,t=T-1}$
repeat
  $\boldsymbol{\theta} \leftarrow \text{UPDATE}(\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_\xi), \boldsymbol{\theta})$ ▷ optimizer update
until convergence

Algorithm 2 One-step per-frame inference

$\tilde{\boldsymbol{X}} \leftarrow [\tilde{\boldsymbol{x}}_0]$
for $t = 0, \dots, T-1$ **do**
  $\tilde{\boldsymbol{\xi}}_t \sim \nu$ ▷ draw label
  $\tilde{\boldsymbol{X}}[t+1] \leftarrow F_{\boldsymbol{\theta}}(\tilde{\boldsymbol{X}}[t], \tilde{\boldsymbol{\xi}}_t)$
end for

interpolates (3) must have is given by the data as

$$L(\mathcal{D}_\xi) = \max_{i \neq j, t} \frac{\|\boldsymbol{x}_{t+1}^i - \boldsymbol{x}_{t+1}^j\|_2}{\sqrt{\|\boldsymbol{x}_t^i - \boldsymbol{x}_t^j\|_2^2 + \|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2^2}}, \qquad i, j = 1, \dots, M, \quad t = 0, \dots, T-1\,. \tag{5}$$

Note that a function interpolating the data (3) with Lipschitz constant (5) on the data can be extended to $\mathcal{X} \times \mathbb{R}^d$ without increasing the Lipschitz constant [18, Theorem 2.10.43]. In practice, we rely on standard deep learning regularization techniques (weight decay, normalization layers) to learn a regular interpolant of the data.

High-dimensional labels decrease the Lipschitz constant We draw our labels from a standard normal of dimension d. For the sake of the theoretical argument we consider normalized labels with unit norm $\|\boldsymbol{\xi}_t^i\|_2 = 1$, which means that $\frac{1}{2}\|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2^2 = 1 - \boldsymbol{\xi}_t^i \cdot \boldsymbol{\xi}_t^j$. That is, the distance between labels is controlled by their inner product. Because the normalized labels are uniformly distributed on the unit sphere $\mathbb{S}^{d-1}$, $\boldsymbol{\xi}_t^i \cdot \boldsymbol{\xi}_t^j$ is of order $1/\sqrt{d}$ with high probability [64, Section 3.3.3 and Theorem 3.3.9]. Hence increasing d widens the separation between labels and reduces the minimal Lipschitz constant of any interpolant as the following proposition shows (see also Figure 2c).

Proposition 2. *For normalized data and normalized labels, if $M \geq 2$ and $d \gtrsim \min\{c_\delta^2 \ln((T+1)M), 2\}$, then*

$$L(\mathcal{D}_\xi) \leq \frac{\sqrt{n}}{\sqrt{2}} \left(1 + c_\delta \sqrt{\frac{\ln((T+1)M)}{d}}\right),$$

holds with probability at least $1 - \delta$, $\delta \in (0, 1)$, and with constant $c_\delta \geq 0$ independent of T, M, d.

See Appendix A.1 for a proof. Thus, we have some control over the Lipschitz constant via the dimension d. The regularity of F is further improved by the closeness between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ (Corollary 1). We do not normalize our labels in practice because we expect the components of the label $\boldsymbol{\xi}_t^i$ to be of order one, which is true when $\boldsymbol{\xi}_t^i \sim \mathcal{N}(0, \boldsymbol{I}_d)$. The argument that $L(\mathcal{D}_\xi)$ is improved by large d remains unchanged.

Orthogonal labels avoid structure induction Traditionally, one would aim to arrange (couple) labels $\boldsymbol{\xi}_t^i$ and targets (next frames) $\boldsymbol{x}_{t+1}^i$. We do not rearrange because of costs, and thus we need to ensure that no spurious structure is induced between labels and next frames. Choosing the labels close to orthogonal achieves this.

2.4 Fitting maps to stochastically lifted data and one-step inference

Training on lifted data and one-step inference We now formulate a regression problem using the labeled data $\mathcal{D}_\xi$. Let us parametrize F as a neural-network function $F_{\boldsymbol{\theta}}$ with weights $\boldsymbol{\theta} \in \mathbb{R}^p$. We then propose to fit $F_{\boldsymbol{\theta}}$ to the labeled data $\mathcal{D}_\xi$ using a regression based loss function,

$$\mathcal{L}(\theta, \mathcal{D}_\xi) = \frac{1}{MT} \sum\nolimits_{i=1}^{M} \sum\nolimits_{t=0}^{T-1} \|F_{\boldsymbol{\theta}}(\boldsymbol{x}_t^i, \boldsymbol{\xi}_t^i) - \boldsymbol{x}_{t+1}^i\|_2^2\,. \tag{6}$$

Other regression loss functions can be used, such as the cross-entropy loss, as we do for some of the numerical experiments; see Appendix F.

To generate a new sample at a $\boldsymbol{x}_t$, we draw a new label $\boldsymbol{\xi}_t \sim \nu$ and evaluate the $F_{\boldsymbol{\theta}}(\boldsymbol{x}_t, \boldsymbol{\xi}_t) = \boldsymbol{x}_{t+1}$ to obtain $\boldsymbol{x}_{t+1}$. This is one-step inference because only a single function evaluation is necessary to generate a new sample.

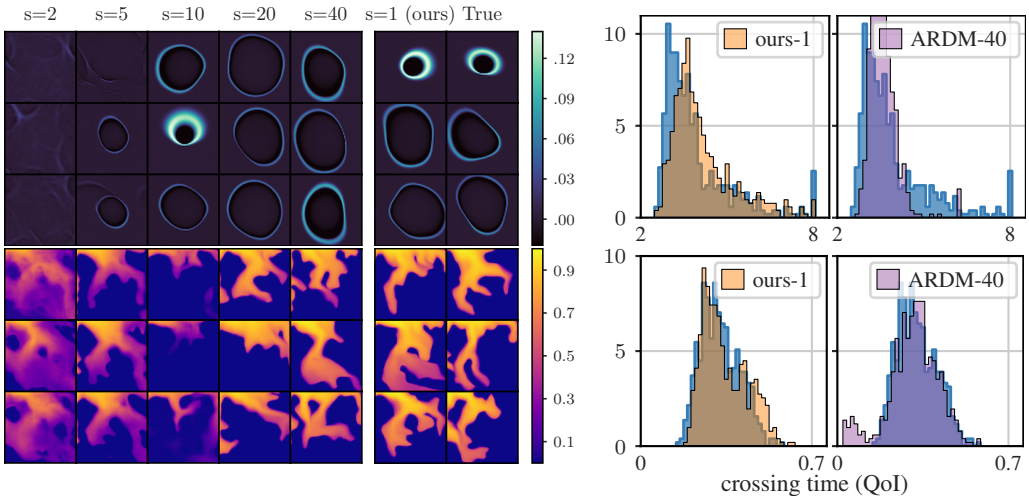


Figure 3: **Left**: Stochastic Lifting generates diverse plausible trajectories in just one step per frame, where ARDMs only begin to generate plausible trajectories when using 20 diffusion steps. **Right**: We compute the crossing time for each trajectory (see appendix D.4) and plot the distribution across 512 generated trajectories vs the ground truth. Stochastic Lifting accurately approximates the ground truth data in distribution, even outperforming ARDM using 40 diffusion steps.

Lifting avoids collapse onto conditional expectation (mean behavior) Critically, when taking the infinite data limit $M \to \infty$, then the minimizer of (6) collapses to the conditional expectation $\mathbb{E}\left[\boldsymbol{X}_{t+1} | \boldsymbol{X}_t = \boldsymbol{x}_t\right]$, since the labels $\boldsymbol{\xi}_t^i$ are drawn independently from $\boldsymbol{x}_t^i$. However, if the label dimension d increases with the number of data points $T \times M$, and the network is rich enough, then the minimizer of the empirical mean-squared error loss can interpolate the data as in (3) and does not collapse to the conditional expectation. In particular, in the case of linear least-squares regression, it has been shown that when the ratio $d/(M \times (T+1))$ between dimension d and number of data points $M \times (T+1)$ remains fixed in the limit $d, M \to \infty$, then collapse is avoided [23].

In summary, with finitely many samples $M < \infty$, we can potentially achieve zero training error (interpolation/memorization) by ensuring that $F_{\boldsymbol{\theta}}$ is expressive enough and that the data is regular enough by choosing d sufficiently large, in which case we expect to obtain a map $F_{\boldsymbol{\theta}}$ that behaves as in Proposition 1. In practice, Adam with weight decay, normalization layers, and standard architecture choices also bias towards learning globally smooth interpolants; see Appendix F.

3 Numerical experiments

Setup For all experiments in this work we parametrize our map $F_{\boldsymbol{\theta}}$ via UNet architectures [52] but our method works with any other standard diffusion backbone [45, 8, 27]. We repurpose the diffusion-time input to condition on our label $\boldsymbol{\xi}_t$; see Appendix E for details.

For all datasets we provide uncurated samples from our model in Appendix G. Additionally, full rendered videos are available in the supplementary zip file.

We stress that we can use Stochastic Lifting directly on the frame pixels and so avoid having to rely on a latent embedding via pre-trained encoder and decoder networks as many other generative modeling methods [51]. By avoiding the latent embedding we ease the significant engineering complexity and hyper-parameter tuning that comes with latent embeddings.

With just one step per frame, Stochastic Lifting achieves comparable accuracy as multi-step models on physics problems We consider two physics problems: first, a traveling wave through a spatially varying random medium, which is motivated by seismic wave propagation [56]; see Appendix D.1. Second, an incompressible two-phase flow in random porous media, which is motivated by petroleum engineering (oil/water) and models of groundwater flow; see Appendix D.2. Importantly, for both wave and flow problem, the initial condition is deterministic and fixed, thus

	Wave		Flow	
	WCT $\downarrow$	**WIM** $\downarrow$	**WCT** $\downarrow$	**WIM** $\downarrow$
ARDM s=2	2.49e–2	9.11e–5	1.18e–1	3.46e–2
ARDM s=5	2.13e–2	1.04e–4	3.39e–2	4.58e–2
ARDM s=10	1.04e–2	6.00e–5	9.13e–2	1.29e–1
ARDM s=20	7.28e–3	1.06e–4	3.25e–2	1.27e–3
ARDM s=40	1.11e–2	9.85e–5	1.86e–3	6.00e–4
SL (ours) s=1	4.53e–3	2.06e–5	9.47e–4	6.08e–4

Table 1: Wasserstein distance between the distribution of 512 samples from the true model vs. the generative model measured on various quantities of interest; see Appendix D.4.

One-step methods ($s = 1$)		
SV2P [6]		262.5
SAVP [35]		109.8
DVD-GAN [12]		109.8
TrIVD-GAN-FP [40]		103.3
FitVid [7]		93.6
NUWA [68]		86.9
Stochastic Lifting (ours)		**74.8**
RIVER [15]	s=100	106.1
MCVD [66]	s=1000	98.8
RaMViD [28]	s=750	84.2
Video Diffusion [25]	s=16	66.9
Rolling Diffusion [53]	s=32	59.6

Table 2: Stochastic lifting outperforms all other one-step methods on the BAIR video-generation dataset.

deterministic approaches such as neural operators cannot capture the stochasticity induced by the random media and permeability fields during the autoregressive roll out; see Section 1.2.

We compare our Stochastic Lifting to an autoregressive diffusion model (ARDM) trained on the same dataset. We use the implementation given in [32], also built around a UNet backbone and comparable parameter count as we use for Stochastic Lifting. We vary the number of diffusion steps s and compare the accuracy to Stochastic Lifting that uses only a single step ($s = 1$).

For the wave and flow problems, ARDM requires 40 steps to generate accurate samples; see Figure 3(left). In contrast, Stochastic Lifting generates accurate samples in one step. Figure 6–7 in the appendix show nearest-neighbor trajectories and frames and so provide evidence that Stochastic Lifting is indeed generating new samples instead of simply retrieving memorized ones. To quantitatively compare the generated samples, we consider the crossing time as a physical quantity of interest, which is the time needed that the wave hits the boundary and the saturation reaches the right-bottom corner of the domain, respectively; see Appendix D.4. We plot the distribution of the crossing time of 512 generated trajectories. Stochastic Lifting accurately matches the ground truth data in distribution; see Figure 3(right). Notice that Stochastic Lifting approximates well the non-Gaussian behavior of the crossing time in the wave example (heavy tail), whereas ARDM fails to capture it even with $s = 40$ steps. Table 1 shows the Wasserstein-2 distance between the distribution of the crossing time (WCT) obtained with Stochastic Lifting and ARDM over various steps. Table 1 also plots the Wasserstein-2 distance (WIM) for another physics quantity of interest, the integrated mass; see Appendix D.4.

Stochastic Lifting is robust to label dimension, once sufficiently high to allow interpolation In Figure 4(left) we plot the final L2 training loss achieved after optimization. For low label dimensions, the neural network cannot interpolate the data, resulting in high training loss. Figure 4(middle) shows that for a sufficiently high label dimension (in particular once interpolation is achieved), the model produces an accurate approximation in the WTC metric (see Table 1). These results align well with Proposition 1. At the same time, the approach is fairly robust to the label dimension once it is sufficiently high because architectures such as UNet and the corresponding modulation (see Appendix E) can compress the labels if necessary.

Stochastic lifting critically depends on conditioning of current frame In Figure 4(right, top) we show the final frame of the rollout from our model which is accurate. In the bottom row, we show what happens when we train a Stochastic Lifting model which maps directly from the initial condition $\boldsymbol{x}_0$ to the final state $\boldsymbol{x}_{T-1}$ (i.e. without sequential rollout). In this case, Stochastic Lifting breaks, producing near noise in the wave problem and non-physical, disconnected flows in the flow problem. This study provides further evidence that Stochastic Lifting crucially depends on the "closeness" between the distributions corresponding to the current and next frame, which aligns well with Corollary 1.

Stochastic Lifting outperforms other one-step methods on BAIR We apply Stochastic Lifting to the Berkeley AI Research (BAIR) robot pushing dataset, which is a standard benchmark [17]. It contains roughly 44000 videos at 64×64 resolution. The standard prediction task is to generate 15

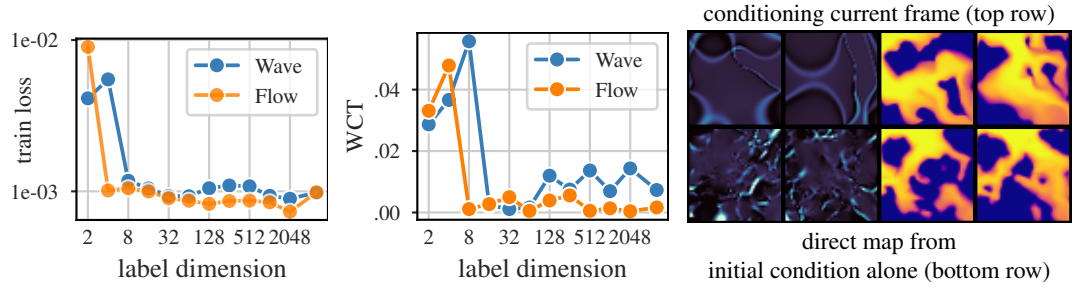


Figure 4: **Left**: Choosing a higher label dimension d allows the neural network to interpolate the data, resulting in one order-of-magnitude lower L2 final training loss. **Middle**: Stochastic Lifting is robust to the choice of the label dimension as long as it is sufficiently high for interpolation. **Right**: Stochastic Lifting critically depends on the conditioning on the current frame to generate a high-quality next frame. If we attempt to jump to the last frame directly, the method breaks down.

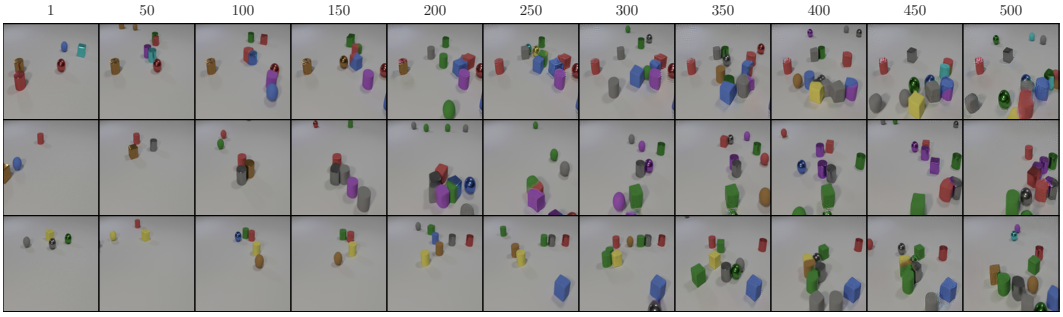


Figure 5: We generate 500 frames of a 128×128 color video in 0.28 seconds on an H100 GPU. The rollout is stable despite being trained on video with only 16 frames. Throughout each trajectory many more objects are in frame than ever occur in the training set.

frames conditioned on one starting frame. We calculate FVD (using the I3D network [10]) via the standard procedure on BAIR by generating 100 new videos starting from 256 initial frames taken from the test set. FVD is then computed between the generated and test videos. Frames are shown in Appendix G.3. Table 2 shows that Stochastic Lifting achieves the lowest (best) FVD [63] over current state-of-the-art one-step generation approaches, and starts closing the gap to multi-step methods [53].

Stochastic lifting scales to long-time rollouts (CLEVRER) and high resolutions (natural video) The CLEVRER dataset [69] is a synthetic dataset designed specifically for video prediction and reasoning tasks. The videos capture diverse objects moving at high speeds from off frame and colliding with each other. There is inherent stochasticity in when and what objects come into frame. The training videos are 16 frames long. For testing, we evaluate our model starting with unseen initial frames and then roll out for 500 frames, well over one order of magnitude longer than the training videos (compare to 14 frames in [11], 120 frames in [15], 64 frames in [43]). The long rollout results in objects continuing to enter from off frame and accumulating in clusters as they collide. Importantly, Stochastic Lifting is able to stably generate the interactions and accumulation of objects, even though such accumulations do not occur in the training set.

Despite learning in pixel space (no latent embedding), we can generate natural videos at relatively high resolutions. We consider the polar bear dataset (Appendix G.5), which consists of 128 videos showing a polar bear walking in an arctic setting, from various viewing angles and distances. Stochastic Lifting generates a new trajectory of 32 frames of size 480×480 in 0.96 seconds on one H100; see Figure 1. The purpose of this experiment is to demonstrate that our approach is scalable and applicable to natural video generation.

4 Conclusions, limitations, and impact statement

Conclusions We provide evidence with our Stochastic Lifting approach that (dynamic) transport, which is inherently expensive due to the rearrangement step, is unnecessary for generating video and trajectories of physics simulations to some extent. We show that interpolation plus smoothness leads to accurate samples in the Wasserstein-2 metric. Empirical results demonstrate that introducing high-dimensional stochastic labels enables generating accurate samples on video generation benchmarks and physics problems.

Limitations (a) We critically build on the strong coupling given by the pair of current and next frame in the training data, which means that Stochastic Lifting fails when aiming to generate images from noise. (b) Increasing the label dimension eventually guarantees the existence of a linear interpolant. However, doing so may require proportionally more data; beyond a point, the regression function cannot be trained accurately.

Impact Statement This paper presents a generative method for video data. Video generation can promote harm through biases in the model and when used with nefarious intentions. We have no reason to assume that the proposed method is more susceptible to this than others.

References

[1] Jørg E. Aarnes, Tore Gimse, and Knut-Andreas Lie. *An Introduction to the Numerics of Flow in Porous Media using Matlab*, pages 265–306. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

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

[4] Sigurd Angenent, Steven Haker, and Allen Tannenbaum. Minimizing flows for the monge–kantorovich problem. *SIAM Journal on Mathematical Analysis*, 35(1):61–97, 2003.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[6] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[7] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

[8] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023.

[9] Jules Berman, Tobias Blickhan, and Benjamin Peherstorfer. Parametric model reduction of mean-field and stochastic systems via higher-order action matching. *Advances in neural information processing systems*, 2024.

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[11] Yifan Chen, Mark Goldstein, Mengjian Hua, Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Probabilistic forecasting with stochastic interpolants and foellmer processes. *arXiv preprint arXiv:2403.13724*, 2024.

[12] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

[13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[14] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[15] Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23263–23274, 2023.

[16] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in neural information processing systems*, 32, 2019.

[17] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 344–356. PMLR, 13–15 Nov 2017.

[18] Herbert Federer. *Geometric Measure Theory*. Springer, 1996.

[19] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

[20] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J. Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.

[21] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[22] Chenjie Gu. Qlmor: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(9):1307–1320, 2011.

[23] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949 – 986, 2022.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[25] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[26] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

[27] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

[28] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022.

[29] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[30] Patrick Kidger. On neural differential equations. *arXiv preprint arXiv:2202.02435*, 2022.

[31] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[32] Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *arXiv preprint arXiv:2309.01745*, 2023.

[33] Nikola Kovachki, Simon Lanthaler, and Siddhartha Mishra. Neural operators: A survey on approximating operators with machine learning. *Acta Numerica*, 32:1–156, 2023.

[34] Boris Kramer and Karen Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, 57(6):2297–2307, 2019.

[35] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[36] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Animashree Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[37] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[38] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.

[39] Lu Lu, Pengzhan Jin, Guofei Pang, Zhiqiang Zhang, and George Em Karniadakis. Learning operators with deeponet: Theory and applications. *Journal of Computational Physics*, 447:110683, 2021.

[40] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint arXiv:2003.04035*, 2020.

[41] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i–convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[42] S. McQuarrie, C. Huang, and K. Willcox. Data-driven reduced-order models via regularised operator inference for a single-injector combustion process. *Journal of the Royal Society of New Zealand*, 51(2):194–211, 2021.

[43] Kangfu Mei and Vishal Patel. Vidm: Video implicit diffusion models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9117–9125, Jun. 2023.

[44] Kirill Neklyudov, Rob Brekelmans, Daniel Severo, and Alireza Makhzani. Action matching: Learning stochastic dynamics from samples. In *International conference on machine learning*, pages 25858–25889. PMLR, 2023.

[45] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[46] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[47] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample Flow Matching: Straightening Flows with Minibatch Couplings, May 2023. arXiv:2304.14772.

[48] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025.

[49] Elizabeth Qian, Boris Kramer, Benjamin Peherstorfer, and Karen Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 2020.

[50] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[53] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. *arXiv preprint arXiv:2402.09470*, 2024.

[54] Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287, 2023.

[55] Filippo Santambrogio. *Optimal Transport for Applied Mathematicians*. Birkhäuser, 2015.

[56] Haruo Sato, Michael C. Fehler, and Takuto Maeda. *Seismic Wave Propagation and Scattering in the Heterogeneous Earth*. Springer, Berlin, Heidelberg, 2nd edition, 2012.

[57] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXVI*, page 87–103, Berlin, Heidelberg, 2024. Springer-Verlag.

[58] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[59] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *International conference on machine learning*, 2023.

[60] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[61] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, page 204, 2019.

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

[63] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[64] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018.

[65] Cédric Villani. *Optimal transport: old and new*. Number 338 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009.

[66] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*, 35:23371–23385, 2022.

[67] Jonathan Weed and Francis Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A), November 2019.

[68] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European conference on computer vision*, pages 720–736. Springer, 2022.

[69] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020.

[70] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024.

[71] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

[72] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *ICML*, 2024.

A Proofs

A.1 Proof of discrete Lipschitz constant bound

Proposition 3. *Assume that all $\boldsymbol{x}_t^i$ are normalized so that $\boldsymbol{x}_t^i \in [0,1]^n$. Assume further that the labels $\boldsymbol{\xi}_t^i$ are sampled uniformly from the unit sphere $\mathbb{S}^{d-1}$, which corresponds to sampling standard normals and normalizing them afterwards. Assume further that $M \geq 2$ and $d \geq \min\{c_\delta^2 \ln((T+1)M), 2\}$ with*

$$c_\delta = \frac{\sqrt{2}}{\sqrt{c}} + \sqrt{\frac{2\ln(2/\delta)}{c\ln(2)}}\,, \tag{7}$$

where $c > 0$ is a constant independent of M, d, T. For $\delta \in (0,1)$, with probability at least $1-\delta$, it holds

$$L(\mathcal{D}_\xi) \leq \frac{\sqrt{n}}{\sqrt{2}}\left(1 + c_\delta\sqrt{\frac{\ln((T+1)M)}{d}}\right). \tag{8}$$

Proof. First, notice that

$$\Delta_{\max} = \max_{i,j,t} \|\boldsymbol{x}_t^i - \boldsymbol{x}_t^j\|_2 \leq \sqrt{n}$$

because of the normalization $\boldsymbol{x}_t^i \in [0,1]^n$. For a fixed triplet (i,j,t), let us consider

$$\frac{\|\boldsymbol{x}_{t+1}^i - \boldsymbol{x}_{t+1}^j\|_2}{\sqrt{\|\boldsymbol{x}_t^i - \boldsymbol{x}_t^j\|_2^2 + \|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2^2}} \leq \frac{\sqrt{n}}{\|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2}\,,$$

because $\|\boldsymbol{x}_t^i - \boldsymbol{x}_t^j\|_2 \geq 0$ and $\Delta_{\max} \leq \sqrt{n}$. Thus, it suffices to lower bound $\min_{i,j,t} \|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2$.

Because the labels $\boldsymbol{\xi}_t^i$ are uniformly sampled from the unit sphere $\mathbb{S}^{d-1}$, for the inner product, we have the following concentration inequality that follows from [64, Exercise 5.1.12]

$$\mathbb{P}[|\langle \boldsymbol{\xi}_t^i, \boldsymbol{\xi}_t^j\rangle| \geq \epsilon] \leq 2\exp\left(-cd\epsilon^2\right)\,, \qquad \epsilon \in [0,1] \tag{9}$$

for a universal constant $c \geq 0$. (To see (9): Set $f_{\boldsymbol{u}} : \mathbb{S}^{d-1} \to \mathbb{R}, \boldsymbol{z} \mapsto \langle \boldsymbol{z}, \boldsymbol{u}\rangle$ for a fixed $\boldsymbol{u} \in \mathbb{S}^{d-1}$. For any $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{S}^{d-1}$ obtain that $\|f_{\boldsymbol{u}}(\boldsymbol{y}) - f_{\boldsymbol{u}}(\boldsymbol{z})\|_2 = \|\langle \boldsymbol{y} - \boldsymbol{z}, \boldsymbol{u}\rangle\|_2 \leq \|\boldsymbol{y} - \boldsymbol{z}\|_2\|\boldsymbol{u}\|_2 = \|\boldsymbol{y} - \boldsymbol{z}\|_2$ because $\|\boldsymbol{u}\|_2 = 1$ by definition. Thus the Lipschitz constant of $f_{\boldsymbol{u}}$ is one. Now notice that $\mathbb{E}[f_{\boldsymbol{u}}(\boldsymbol{z})] = 0$ because $\mathbb{E}[\boldsymbol{z}] = 0$ for $\boldsymbol{z}$ uniformly on $\mathbb{S}^{d-1}$. Therefore with [64, Exercise 5.1.12] obtain that $\mathbb{P}[|f_{\boldsymbol{u}}(\boldsymbol{z})| \geq \epsilon) \leq 2\exp(-cd\,\epsilon^2)$ for $\epsilon \in (0,1]$. Taking two independent $\boldsymbol{\xi}_t^i, \boldsymbol{\xi}_t^j$ uniformly on $\mathbb{S}^{d-1}$, we obtain $\mathbb{P}[\langle \boldsymbol{\xi}_t^i, \boldsymbol{\xi}_t^j\rangle \geq \epsilon] = \mathbb{E}_{\boldsymbol{\xi}_t^j}[\mathbb{P}[\langle \boldsymbol{\xi}_t^i, \boldsymbol{\xi}_t^j\rangle \geq \epsilon | \boldsymbol{\xi}_t^j]] \leq \mathbb{E}_{\boldsymbol{\xi}_t^j}[2\exp(-cd\,\epsilon^2)] = 2\exp(-cd\,\epsilon^2)$ because the right-hand side of the bound for $f_{\boldsymbol{u}}$ is independent of $\boldsymbol{\xi}_t^j$.)

Thus, we obtain

$$\mathbb{P}[\sqrt{2(1 - |\langle \boldsymbol{\xi}_t^i, \boldsymbol{\xi}_t^j\rangle|} \leq \sqrt{2(1-\epsilon)}] \leq 2\exp\left(-cd\epsilon^2\right)\,, \qquad \epsilon \in [0,1]\,,$$

holds. This is useful because $\|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2^2 = 2 - 2\langle \boldsymbol{\xi}_t^i, \boldsymbol{\xi}_t^j\rangle$ and thus $\|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2 \geq \sqrt{2(1 - |\langle \boldsymbol{\xi}_t^i, \boldsymbol{\xi}_t^j\rangle|)}$. Selecting a $\delta \in (0,1)$ and using a union bound, we obtain

$$\mathbb{P}[\min_{i,j,t} \|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2 \geq \sqrt{2(1-\epsilon_\delta)}] \geq 1 - \delta\,,$$

for

$$\epsilon_\delta = \sqrt{\frac{1}{cd}\left(\ln(\hat{M}) + \ln(2/\delta)\right)}$$

where $\hat{M} = (T+1)\binom{M}{2}$.

We have $\ln(\hat{M}) \leq 2\ln((T+1)M)$ because $T \geq 0$ and under the assumption $M \geq 2$,

$$\begin{aligned}
\ln(\hat{M}) &= \ln\Big((T+1)\,\frac{M(M-1)}{2}\Big) \\
&= \ln(T+1) + \ln\Big(\frac{M(M-1)}{2}\Big) \\
&\leq \ln(T+1) + \ln\Big(\tfrac{M^2}{2}\Big) \quad \text{(because } M(M-1) \leq M^2\text{)} \\
&= \ln(T+1) + 2\ln M - \ln 2 \\
&\leq \ln(T+1) + 2\ln M \quad \text{(dropping the negative term } -\ln 2\text{)} \\
&= \big[\ln(T+1) + \ln M\big] + \ln M \\
&= \ln\big((T+1)M\big) + \ln M \\
&\leq 2\,\ln\big((T+1)M\big) \quad \text{(since } \ln M \leq \ln\big((T+1)M\big)\text{)},
\end{aligned}$$

and thus obtain

$$\epsilon_\delta^2 = \frac{1}{c}\left(\frac{2\ln((T+1)M)}{d} + \frac{\ln(2/\delta)}{d}\right),$$

which leads to

$$\epsilon_\delta \leq \frac{1}{\sqrt{c}}\left(\sqrt{\frac{2\ln((T+1)M)}{d}} + \sqrt{\frac{\ln(2/\delta)}{d}}\right)$$

Because $\ln((T+1)M) \geq \ln(2)$ (because we have $(T+1)M \geq 2$),

$$\sqrt{\frac{\ln(2/\delta)}{d}} = \sqrt{\frac{\ln(2/\delta)}{\ln(2)}}\sqrt{\frac{\ln(2)}{d}} \leq \sqrt{\frac{\ln(2/\delta)}{\ln(2)}}\sqrt{\frac{\ln((T+1)M)}{d}}\,.$$

Now define c_δ as in (7) to obtain

$$\epsilon_\delta \leq c_\delta\sqrt{\frac{\ln((T+1)M)}{d}}\,.$$

With (9), we have with probability at least $1-\delta$ that

$$\max_{i,j,t}\frac{\sqrt{n}}{\|\boldsymbol{\xi}_t^i - \boldsymbol{\xi}_t^j\|_2} \leq \frac{\sqrt{n}}{\sqrt{2(1-\epsilon_\delta)}}\,,$$

which is non-vacuous as long as $d \geq c_\delta^2\ln((T+1)M)$ so that $\epsilon_\delta \leq 1$. We further have $\sqrt{2(1-\epsilon_\delta)} \geq \sqrt{2}(1-\frac{1}{2}\epsilon_\delta)$ and thus

$$\frac{\sqrt{n}}{\sqrt{2}(1-\frac{1}{2}\epsilon_\delta)} \leq \frac{\sqrt{n}}{\sqrt{2}}(1+\epsilon_\delta) \leq \frac{\sqrt{n}}{\sqrt{2}}\left(1 + c_\delta\sqrt{\frac{\ln((T+1)M)}{d}}\right),$$

which shows the bound (8). □

A.2 Interpolation of training data + smoothness = bound on test data

We recall the following lemma for the sake of self-containedness:

Lemma 1. *For $F : \mathbb{R}^n \to \mathbb{R}^m$ Lipschitz continuous and $\mu, \nu \in \mathcal{P}(\mathbb{R}^n)$,*

$$W_2(F_\sharp\mu, F_\sharp\nu) \leq L_F W_2(\mu,\nu)$$

Proof. Assume $\gamma^* \in \Gamma(\mu,\nu)$ is the optimal coupling between μ and ν, hence

$$\gamma^*(\cdot,\mathbb{R}^n) = \mu,\ \gamma^*(\mathbb{R}^n,\cdot) = \nu, \text{ and } \int_{\mathbb{R}^n\times\mathbb{R}^n}|\boldsymbol{x}-\boldsymbol{y}|^2\,\mathrm{d}\gamma^*(x,y) = W_2(\mu,\nu)^2.$$

Consider the coupling $\gamma_F := (F \times F)_\sharp \gamma^*$. Take $B \subset \mathbb{R}^m$ measureable. Then,

$$\begin{aligned}\gamma_F(B \times \mathbb{R}^m) &= ((F \times F)_\sharp \gamma^*)(B \times \mathbb{R}^m) = \gamma^*((F \times F)^{-1}(B \times \mathbb{R}^m)) \\ &= \gamma^*(F^{-1}(B) \times F^{-1}(\mathbb{R}^m)) = \gamma^*(F^{-1}(B) \times \mathbb{R}^n) = \mu(F^{-1}(B)) = (F_\sharp \mu)(B).\end{aligned}$$

Analogously, $\gamma_F^*(\mathbb{R}^m \times B) = (F_\sharp \nu)(B)$. Hence γ_F^* is a valid competitor for the optimal transport problem from $F_\sharp \mu$ to $F_\sharp \nu$:

$$\begin{aligned}W_2(F_\sharp \mu, F_\sharp \nu)^2 &= \min_{\gamma \in \Gamma(F_\sharp \nu, F_\sharp \mu)} \int_{\mathbb{R}^m \times \mathbb{R}^m} |\boldsymbol{x} - \boldsymbol{y}|^2 \, \mathrm{d}\gamma(\boldsymbol{x}, \boldsymbol{y}) \leq \int_{\mathbb{R}^m \times \mathbb{R}^m} |\boldsymbol{x} - \boldsymbol{y}|^2 \, \mathrm{d}\gamma_F(\boldsymbol{x}, \boldsymbol{y}) \\ &= \int_{\mathbb{R}^n \times \mathbb{R}^n} |F(x) - F(y)|^2 \, \mathrm{d}\gamma^*(\boldsymbol{x}, \boldsymbol{y}) \leq L_F^2 \int_{\mathbb{R}^n \times \mathbb{R}^n} |\boldsymbol{x} - \boldsymbol{y}|^2 \, \mathrm{d}\gamma^*(\boldsymbol{x}, \boldsymbol{y}) = L_F^2 W_2(\mu, \nu)^2\end{aligned}$$

Taking the square root gives the claimed result. □

Proposition (1)**.** *Let F interpolate the training data D_ξ as in* (3)*. At any time t, consider $\tilde{M}$ test samples $\mathcal{D}_\xi^{test} = \{(\tilde{\boldsymbol{x}}_t^i, \tilde{\boldsymbol{x}}_{t+1}^i, \tilde{\boldsymbol{\xi}}_t^i)\}_{i=1}^{\tilde{M}}$ sampled from $\pi_t \otimes \nu$. Evaluate now the map F to obtain the generated samples $\hat{\boldsymbol{x}}_t^i = F(\tilde{\boldsymbol{x}}_t^i, \tilde{\boldsymbol{\xi}}_t^i)$ for $i = 1, \ldots, \tilde{M}$. Then,*

$$\mathbb{E}_{\mathcal{D}_\xi, \mathcal{D}_\xi^{test}} \left[W_2(\hat{\rho}_{t+1}, \tilde{\rho}_{t+1})^2 \right] \leq 2C(1 + L_F^2) \min(M, \tilde{M})^{-1/\alpha},$$

where $\hat{\rho}_{t+1}$ and $\tilde{\rho}_{t+1}$ are the empirical measures corresponding to the generated samples $\{\hat{\boldsymbol{x}}_{t+1}^i\}_{i=1}^{\tilde{M}}$ and the test samples $\{\tilde{\boldsymbol{x}}_{t+1}^i\}_{i=1}^{\tilde{M}}$, $C > 0$ is a constant, L_F is the Lipschitz constant of F, and $\alpha \geq 0$ depends on the intrinsic dimension of ρ_t, ρ_{t+1}, or ν, whichever is largest. When $F(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{x} + \mathrm{d}t R(\boldsymbol{x}, \boldsymbol{\xi})$, equation 4 can be written as

$$\mathbb{E}_{\mathcal{D}_\xi, \mathcal{D}_\xi^{test}} \left[W_2(\hat{\rho}_{t+1}, \tilde{\rho}_{t+1})^2 \right] \leq C(3 + 4\mathrm{d}t^2 L_R^2) \min(M, \tilde{M})^{-2/\alpha}.$$

If $F(\boldsymbol{x}, \boldsymbol{\xi}) = \boldsymbol{x} + \mathrm{d}t\, b(\boldsymbol{x}) + \varepsilon\sigma(\boldsymbol{\xi})$, then (4) *can be written as*

$$\mathbb{E}_{\mathcal{D}_\xi, \mathcal{D}_\xi^{test}} \left[W_2(\hat{\rho}_{t+1}, \tilde{\rho}_{t+1})^2 \right] \leq C(3 + 4\mathrm{d}t^2 L_b^2 + 4\varepsilon^2 L_\sigma^2) \min(M, \tilde{M})^{-2/\alpha}.$$

Notice that the strong concentration bound [67, Proposition 20] for empirical measures $\hat{\mu}$ and μ imply that control over the expectation is sufficient to bound the error with high probability,

$$\mathbb{P}\left[W_2(\mu, \hat{\mu})^2 \geq \mathbb{E}\left[W_2(\mu, \hat{\mu})^2 \right] + \varepsilon \right] \leq \exp\left(-2M\varepsilon^2\right), \qquad \epsilon > 0.$$

Proof. By the triangle inequality,

$$W_2(F_\sharp(\tilde{\rho}_t \otimes \tilde{\nu}), \tilde{\rho}_{t+1}) \leq \underbrace{W_2(F_\sharp(\hat{\rho}_t \otimes \hat{\nu}), \hat{\rho}_{t+1})}_{=:(i)} + \underbrace{W_2(\hat{\rho}_{t+1}, \tilde{\rho}_{t+1})}_{=:(ii)} + \underbrace{W_2(F_\sharp(\tilde{\rho}_t \otimes \tilde{\nu}), F_\sharp(\hat{\rho}_t \otimes \hat{\nu}))}_{=:(iii)}$$

The first term (i) is zero by the assumption that F interpolates the training data.

The second term (ii) is bounded by the Wasserstein distance between the training and test data. Its expectation value can be controlled by the fact that both empirical measures are drawn from the same distribution. In [67, Proposition 5], it is shown that for an empirical measures $\hat{\mu} \approx \mu$ with M samples,

$$\mathbb{E}\left[W_2(\mu, \hat{\mu})^2 \right] \leq CM^{-2/\alpha},$$

where α denotes the intrinsic dimension of the data. Hence

$$\mathbb{E}\left[W_2(\hat{\rho}_{t+1}, \tilde{\rho}_{t+1})^2 \right] \leq C_{t+1}(M^{-2/\alpha_{t+1}} + \tilde{M}^{-2/\alpha_{t+1}}).$$

For the third term (iii), we can estimate

$$\mathbb{E}\left[W_2(F_\sharp(\tilde{\rho}_t \otimes \tilde{\nu}), F_\sharp(\hat{\rho}_t \otimes \hat{\nu})) \right] \leq \mathbb{E}\left[L_F W_2(\tilde{\rho}_t \otimes \tilde{\nu}, \hat{\rho}_t \otimes \hat{\nu}) \right]$$

where L_F is the Lipschitz constant of F:

$$|F(\boldsymbol{x},\boldsymbol{\xi}) - F(\boldsymbol{x}',\boldsymbol{\xi}')| \leq L_F|(\boldsymbol{x},\boldsymbol{\xi}) - (\boldsymbol{x}',\boldsymbol{\xi}')| \quad \forall(\boldsymbol{x},\boldsymbol{\xi},\boldsymbol{x}',\boldsymbol{\xi}') \in \mathcal{X}\times\mathbb{R}^d\times\mathcal{X}\times\mathbb{R}^d$$

Now, note that the quadratic Wasserstein distance in this case factorizes as

$$W_2(\tilde{\rho}_t\otimes\tilde{\nu},\hat{\rho}_t\otimes\hat{\nu})^2 = W_2(\tilde{\rho}_t,\hat{\rho}_t)^2 + W_2(\tilde{\nu},\hat{\nu})^2$$

and hence

$$\mathbb{E}\left[W_2(\tilde{\rho}_t\otimes\tilde{\nu},\hat{\rho}_t\otimes\hat{\nu})^2\right] \leq C_t(M^{-2/\alpha_t} + \tilde{M}^{-2/\alpha_t}) + C_\nu(M^{-2/\alpha_\nu} + \tilde{M}^{-2/\alpha_\nu})\,.$$

Collecting all terms, we get

$$\begin{aligned}&\mathbb{E}\left[W_2(F_\sharp(\tilde{\rho}_t\otimes\tilde{\nu}),\tilde{\rho}_{t+1})^2\right]\\ &\leq C_{t+1}(M^{-2/\alpha_{t+1}} + \tilde{M}^{-2/\alpha_{t+1}}) + L_F^2 C_t(M^{-2/\alpha_t} + \tilde{M}^{-2/\alpha_t}) + L_F^2 C_\nu(M^{-2/\alpha_\nu} + \tilde{M}^{-2/\alpha_\nu})\end{aligned}$$

With $\alpha = \max(\alpha_t,\alpha_{t+1},\alpha_\nu)$ and $C = \max(C_t, C_{t+1}, C_\nu)$, we obtain the result from Proposition 1.

To show the corollary, we use that the optimal coupling between $\tilde{\rho}_t\otimes\tilde{\nu}$ and $\hat{\rho}_t\otimes\hat{\nu}$ is given by $\gamma_x\otimes\gamma_\xi$, where γ_x is the optimal coupling between $\tilde{\rho}_t$ and $\hat{\rho}_t$ and γ_ξ is the optimal coupling between $\tilde{\nu}$ and $\hat{\nu}$:

$$\begin{aligned}W_2(F_\sharp(\tilde{\rho}_t\otimes\tilde{\nu}),F_\sharp(\hat{\rho}_t\otimes\hat{\nu}))^2 &\leq \int_{\mathcal{X}^2\times\mathbb{R}^{2d}} |F(\hat{x},\hat{\boldsymbol{\xi}}) - F(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\xi}})|^2 \mathrm{d}\gamma_x\otimes\gamma_\xi\\ &= \int_{\mathcal{X}^2\times\mathbb{R}^{2d}} |\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}} + \mathrm{d}t(R(\hat{\boldsymbol{x}},\hat{\boldsymbol{\xi}}) - R(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\xi}}))|^2 \mathrm{d}\gamma_x\otimes\gamma_\xi \quad (10)\end{aligned}$$

Next, Young's inequality implies that

$$2|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}||R(\hat{\boldsymbol{x}},\hat{\boldsymbol{\xi}}) - R(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\xi}})| \leq \frac{1}{\varepsilon}|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}|^2 + \varepsilon|R(\hat{\boldsymbol{x}},\hat{\boldsymbol{\xi}}) - R(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\xi}})|^2$$

for any $\varepsilon > 0$ and hence, with $\varepsilon = \mathrm{d}t$,

$$|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}} + \mathrm{d}t(R(\hat{\boldsymbol{x}},\hat{\boldsymbol{\xi}}) - R(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\xi}}))|^2 \leq 2\left(|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}|^2 + \mathrm{d}t^2|R(\hat{\boldsymbol{x}},\hat{\boldsymbol{\xi}}) - R(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\xi}})|^2\right)$$

and

$$\begin{aligned}(10) &\leq 2\int_{\mathcal{X}\times\mathcal{X}} |\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}|^2 \mathrm{d}\gamma_x + 2\mathrm{d}t^2 \int_{\mathcal{X}^2\times\mathbb{R}^{2d}} |R(\hat{\boldsymbol{x}},\hat{\boldsymbol{\xi}}) - R(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{\xi}})|^2 \mathrm{d}\gamma_x\otimes\gamma_\xi\\ &\leq 2W_2(\tilde{\rho}_t,\hat{\rho}_t)^2 + 2\mathrm{d}t^2 L_R^2 W_2(\tilde{\rho}_t\otimes\nu,\hat{\rho}_t\otimes\hat{\nu})^2\\ &= 2W_2(\tilde{\rho}_t,\hat{\rho}_t)^2 + 2\mathrm{d}t^2 L_R^2 (W_2(\tilde{\rho}_t,\hat{\rho}_t)^2 + W_2(\tilde{\nu},\hat{\nu})^2)\end{aligned}$$

The proof for the case $F(\boldsymbol{x},\boldsymbol{\xi}) = \boldsymbol{x} + \mathrm{d}t\, b(\boldsymbol{x}) + \varepsilon\sigma(\boldsymbol{\xi})$ follows analogously, using

$$\begin{aligned}&|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}} + \mathrm{d}t(b(\hat{\boldsymbol{x}}) - b(\tilde{\boldsymbol{x}})) + \varepsilon(\sigma(\hat{\boldsymbol{\xi}})) - \sigma(\tilde{\boldsymbol{\xi}}))|^2\\ &\qquad\qquad \leq 2|\hat{\boldsymbol{x}} - \tilde{\boldsymbol{x}}| + 2\mathrm{d}t^2|b(\hat{\boldsymbol{x}}) - b(\tilde{\boldsymbol{x}}) + \frac{\varepsilon^2}{\mathrm{d}t^2}(\sigma(\hat{\boldsymbol{\xi}})) - \sigma(\tilde{\boldsymbol{\xi}})|^2\,.\end{aligned}$$

□

B Explicit construction of an interpolating map

Assume $\boldsymbol{\xi}_t$ is distributed uniformly on the unit sphere $\mathbb{S}^{d-1}$ and $d \geq M$. Consider the following general form for F:

$$F(\boldsymbol{x}_t,\boldsymbol{\xi}_t) = b(\boldsymbol{x}_t) + \sum_{i,k=1}^{M} (\hat{\boldsymbol{x}}_{t+1}^i - b(\boldsymbol{x}_t))\hat{\mathbb{G}}_{ik}^{-1}\hat{\boldsymbol{\xi}}_t^k\cdot\boldsymbol{\xi}_t,$$

where $b:\mathcal{X}\to\mathcal{X}$ captures the mean behavior, i.e. $b(\boldsymbol{x}_t) = \mathbb{E}[\boldsymbol{X}_{t+1}|\boldsymbol{X}_t = \boldsymbol{x}_t]$, $(\hat{\boldsymbol{x}}_{t+1}^i,\hat{\boldsymbol{\xi}}_t^i),\ i = 1,\ldots,M$ are the training points, and $\hat{\mathbb{G}}_{ik} = \hat{\boldsymbol{\xi}}_t^i\cdot\boldsymbol{\xi}_t^k$. Note that $\hat{\mathbb{G}}$ is almost surely invertible when $d \geq M$.

This function interpolates the training data:

$$F(\hat{\boldsymbol{x}}_t^j, \hat{\boldsymbol{\xi}}_t^j) = b(\boldsymbol{x}_t) + \sum_{i,k=1}^{M} (\hat{\boldsymbol{x}}_{t+1}^i - b(\boldsymbol{x}_t))\hat{\mathbb{G}}_{ik}^{-1}\hat{\boldsymbol{\xi}}_t^k \cdot \boldsymbol{\xi}_t^j$$
$$= b(\boldsymbol{x}_t) + \sum_{i=1}^{M} (\hat{\boldsymbol{x}}_{t+1}^i - b(\boldsymbol{x}_t))\delta_{ij} = \hat{\boldsymbol{x}}_{t+1}^j \quad \forall j = 1, \dots, M$$

At the same time, it is an affine function in $\boldsymbol{\xi}_t$ and as regular in $\boldsymbol{x}_t$ as b is, and the latter is smooth in physical systems and natural videos. We can improve the regularity of F by improving the conditioning of $\hat{\mathbb{G}}$. As discussed, $\hat{\mathbb{G}}$ approaches the identity matrix for large d and fixed M.

Furthermore, note that for any $\mathbf{v} \in \mathbb{S}^{d-1}$, $\sqrt{d}\mathbf{v} \cdot \boldsymbol{\xi}_t \to \eta \sim \mathcal{N}(0,1)$ in distribution as $d \to \infty$ [64, Theorem 3.3.9]. For fixed $\boldsymbol{x}_t$, the distribution of $F(\boldsymbol{x}_t, \boldsymbol{\xi}_t)$ is thus similar to

$$b(\boldsymbol{x}_t) + \frac{1}{\sqrt{d}} \sum_{i=1}^{M} (\hat{\boldsymbol{x}}_{t+1}^i - b(\boldsymbol{x}_t))\eta_i \quad \text{where} \quad \eta_i \sim \mathcal{N}(0,1)\ \forall i\,,$$

namely a Gaussian perturbation of the expected value of $\boldsymbol{X}_{t+1}|\boldsymbol{X}_t = \boldsymbol{x}_t$ in the directions of other training points.

C Nearest Neighbor analysis

We conduct a nearest neighbor analysis to ensure that our model generates novel samples rather than memorizing training data. Specifically, for each example: the top row shows a randomly selected generated sample, the middle row displays the closest matching sample from the training set, and the bottom row shows the closest training-set neighbor to the middle row itself. The visual variation between the generated sample (top row) and its nearest training-set neighbor (middle row) is similar in scale to the variation observed between the two training-set samples (middle and bottom rows). This demonstrates that our model generalizes effectively without simply replicating training examples.

D Numerical experiments

D.1 Traveling wave through random media (Wave)

The initial condition is a Gaussian bump at the center of the domain. As the wave evolves, the random medium leads to different wave speeds that lead to a deformation of the wave front, which leads to a distribution of diverse solution fields; see Appendix G.1 for visualizations. Importantly, for both wave and flow problem, the initial condition is deterministic and fixed, thus deterministic approaches such as neural operators cannot capture the stochasticity induced by the random media and permeability fields during the autoregressive roll out; see Section 1.2.

Problem setup:

$$\begin{cases} \partial_{tt} u(t, \boldsymbol{x}) = c^2(\boldsymbol{x})\Delta u(t, \boldsymbol{x}), & t \in (0, 8],\ \boldsymbol{x} \in \Omega, \\ u(0, \boldsymbol{x}) = \exp\Big(-30\,\|\boldsymbol{x} - [\pi, \pi]\|^2\Big), & \boldsymbol{x} \in \Omega, \\ \partial_t u(0, \boldsymbol{x}) = 0, & \boldsymbol{x} \in \Omega. \end{cases}$$

$\Omega = [0, 2\pi] \times [0, 2\pi], T = 8.0$ We look at the 2D wave equation with periodic boundary conditions.

The wave speed $c(\boldsymbol{x})$ varies in space; it is generated by choosing random Fourier modes from a log-normal distribution which peaks at some wave number $k = 1$ which controls the regularity of the random field.

We have $M = 1024$ trajectories, discretized at $T = 64$ time points and 64×64 points in space.

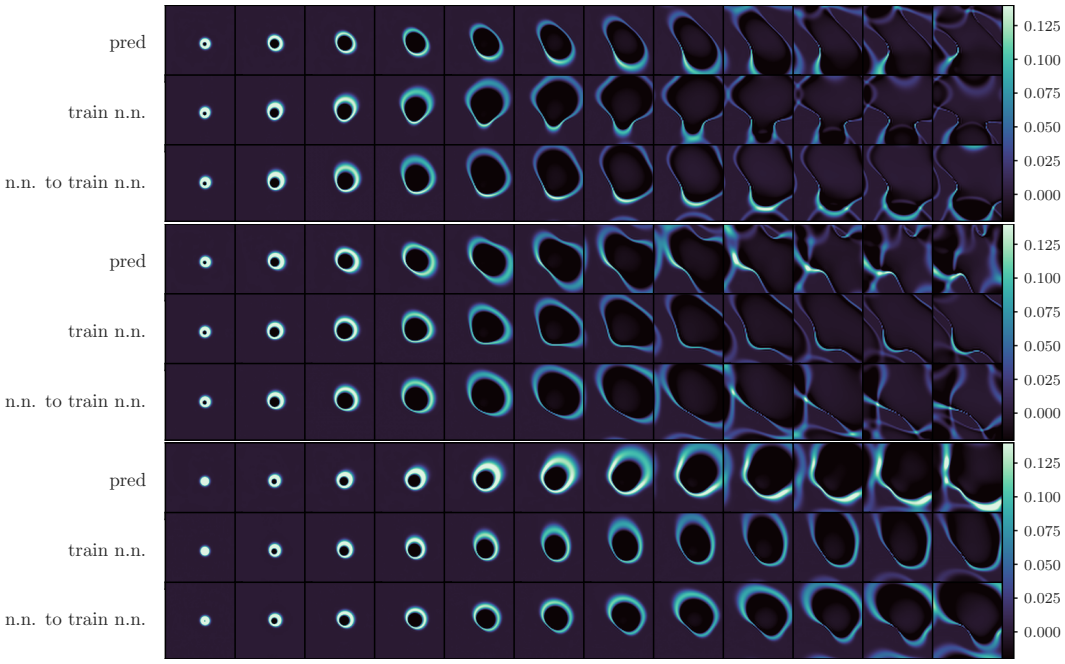


Figure 6: **Wave nearest neighbor analysis**. Of each image: top row randomly chosen generated sample, middle row nearest neighbor in the training set, bottom row nearest neighbor in the training set to the middle row. The variation between the top and middle row should be roughly similar to the variation between the middle and bottom row, indicating generalization without memorization.

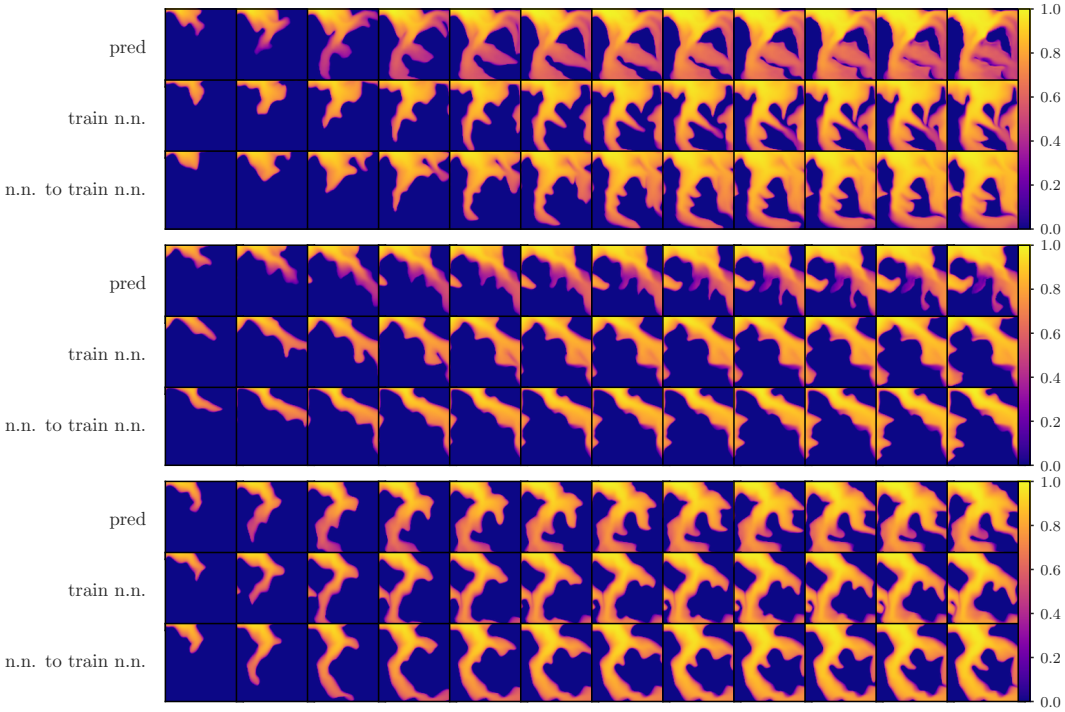


Figure 7: **Flow nearest neighbor analysis**. Of each image: top row randomly chosen generated sample, middle row nearest neighbor in the training set, bottom row nearest neighbor in the training set to the middle row. The variation between the top and middle row should be roughly similar to the variation between the middle and bottom row, indicating generalization without memorization.

D.2 Two-phase flow in random porous media (Flow)

We consider an incompressible and immiscible two-phase flow system of water and oil phases [1]. One phase of the flow moves through the domain as it interacts with a randomly permeability field which deforms the flow into a high variance distribution of complex shapes; see appendix G.2. The governing equations are

$$-\nabla \cdot (K\lambda(s)\nabla p) = q\,,$$
$$\phi \frac{\partial}{\partial t} s + \nabla \cdot (f(s)v) = \frac{q_\omega}{\rho_\omega}\,,$$

where s is the saturation field , K represents the permeability tensor, $\lambda(s)$ is the total fluid mobility (the sum of water and oil mobilities), and q is the source term. In the second equation, ϕ denotes the porosity, $f(s)$ the fractional flow of water, and v is the Darcy velocity.

The randomness enters via the permeability field K, is the exponential $K(x) = \exp(\epsilon Z(x) + \bar{Z} + Y(x))$, where Z and Y are independent Gaussian random fields, each generated with a Matérn covariance function (marginal standard deviation $\sigma = 3.0$, correlation length $\lambda = 10$, smoothness parameter $\nu = 1$, noisiness parameter $\epsilon = 0.01$.

The equations are discretized with an explicit upwind finite-volume discretization method. We use 100×100 cells, end time 0.7, and time-step size 0.007. The initial condition is zero saturation with a localized source on the upper left corner of the spatial domain.

D.3 Video data sets

Polar bear The training data set for the polar bear video generation task has been generated with OpenAI's SORA engines using prompts such as "In a snow-covered Arctic environment with crisp, clear natural lighting, a white polar bear moves laterally from left to right at a moderate walking speed. The camera is at shoulder height, 4 feet from the bear, and maintains a locked side view directly perpendicular to the bear's path. The animal stays centered and occupies about half the frame throughout. Footage is in photorealistic high-definition." We generated 152 videos, 300 frames long, at resolution 480 $\times$ 480. We use 128 videos for training and 24 videos for testing. For generation, we use the starting frames of the test as the initial condition and we rollout for 32 frames.

We measure inference runtime by using the largest batch size that fits into the memory of our H100 GPUs and then divide the total runtime by that batch to get the per-video generation runtime.

D.4 Metrics

For our physics-based examples, we regard the realizations of our stochastic process as time-dependent solution fields. That is, the j-th component of a sample $\boldsymbol{x}_t$ corresponds to the evaluation of $u : \mathcal{T} \times [0, L]^2 \to \mathbb{R}$ at the j-th grid point in $[0, L]^2$ and at time t.

To ease comparisons, we normalize $\mathcal{T}$ to $[0, 1]$ and L to 1 when computing all metrics.

For the sake of comparison and visualization, we require low-dimensional metrics. We construct them as follows: Take a map $f : \boldsymbol{x}_t \mapsto f(\boldsymbol{x}_t) \in \mathbb{R}$. The push-forward of the empirical distribution of $\boldsymbol{x}_t$ under f defines a distribution on $\mathbb{R}$.

Given samples from the ground truth $\boldsymbol{x}_t \sim \hat{\rho}_t$ and generated samples $\tilde{\boldsymbol{x}}_t \sim \tilde{\rho}_t$, the mismatch can be described by $W_2(f_\sharp \hat{\rho}_t, f_\sharp \tilde{\rho}_t)$, which is easy to compute as these are one-dimensional distributions.

The mismatch in this case is defined as a function of time t. It is also possible to consider a map g that takes $\{\boldsymbol{x}_t\}_t$ an entire trajectory and returns a one-dimensional quantity of interest.

Wasserstein distance of Mass We look at Mass Wasserstein distance (WDM). In this case, the function f corresponds to an l_1 norm, denoted m:

$$m(\boldsymbol{x}_t) = \frac{1}{n}\|\boldsymbol{x}_t\|_1 \approx \int_{[0,L]^2} |u(t,x)|\,\mathrm{d}x, \quad \text{MWD} = \frac{1}{T}\sum_{t=1}^{T} W_2(m_\sharp \hat{\rho}_t, m_\sharp \tilde{\rho}_t).$$

Wasserstein crossing time Given some subset $B \subset [0, L]^2$ the earliest arrival time τ for some threshold value $c \in \mathbb{R}$ is given by

$$\tau(u) = \min\{t \in \mathcal{T} : u(t, x) > c \text{ for } x \in B\}.$$

For a trajectory $\{\boldsymbol{x}_t\}_t$, this corresponds to a threshold value being crossed for a subset of entries of $\boldsymbol{x}_t$. Denote by ρ the union of the time marginals ρ_t for all t. The Wasserstein crossing time is defined as

$$\text{WCT} = W_2(\tau_\sharp \hat{\rho}, \tau_\sharp \tilde{\rho}).$$

E Architecture details

UNet backbone Our network follows the canonical encoder–decoder UNet with skip–connections, implemented in `Flax`. All architecture decisions are standard. We use GroupNorm, no dropout, stride-2 3×3 convolution. At each spatial scale for our residual blocks the channel depths are,

$$\texttt{medium\_feature\_depths} = [128,\ 256,\ 512],$$
$$\texttt{large\_feature\_depths} = [128,\ 256,\ 512,\ 1024].$$

The only architectural modification is the conditioning on the label $\boldsymbol{\xi}_t$. Many diffusion backbones condition on the "diffusion time" via some modulation scheme: The only difference is we skip the initial sinusoidal embedding which maps the diffusion-time scalar to a vector. Instead we directly embed $\boldsymbol{\xi}$ with a two-layer MLP with width 512. The MLP output then modulates the residual feature maps via a standard FiLM modulation scheme [46].

As stated in the main text, apart from re-purposing the timestep embedding to encode the conditioning label ξ, all components adhere to established UNet practice, enabling a like-for-like comparison with standard diffusion backbones.

F Training details

For the physics-based problems (wave, flow), we use the L2 loss defined in equation 6. For the video datasets, we replace the L2 metric ($\|x - y\|_2^2$) with a binary cross entropy (BCE) loss. We found the BCE loss beneficial because it implicitly normalizes the network outputs (logits) to the $[0, 1]$ range using a sigmoid function. This normalization prevents the network's predictions from diverging during rollout, making it particularly suitable for video data, which naturally lies within $[0, 1]$.

Table 3: Hyper-parameters used for each dataset.

	Wave	Flow	BAIR	CLEVRER	POLAR
Label Dim	64	64	128	32	8
Batch Size	64	64	512	256	1024
Iterations	350 000	350 000	500 000	500 000	1 000 000
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
Schedule	cosine	cosine	cosine	cosine	cosine
Network Size	Medium	Medium	Medium	Medium	Large
Loss	L2	L2	BCE	BCE	BCE
Trajectories (M)	1024	1000	43 000	10 000	128
Time Points (T)	64	50	16	16	300
State Dim (n)	64^2	96^2	$64^2 \times 3$	$128^2 \times 3$	$480^2 \times 3$
Markov Window (m)	3	3	1	3	5

G Uncurated Samples

G.1 Wave

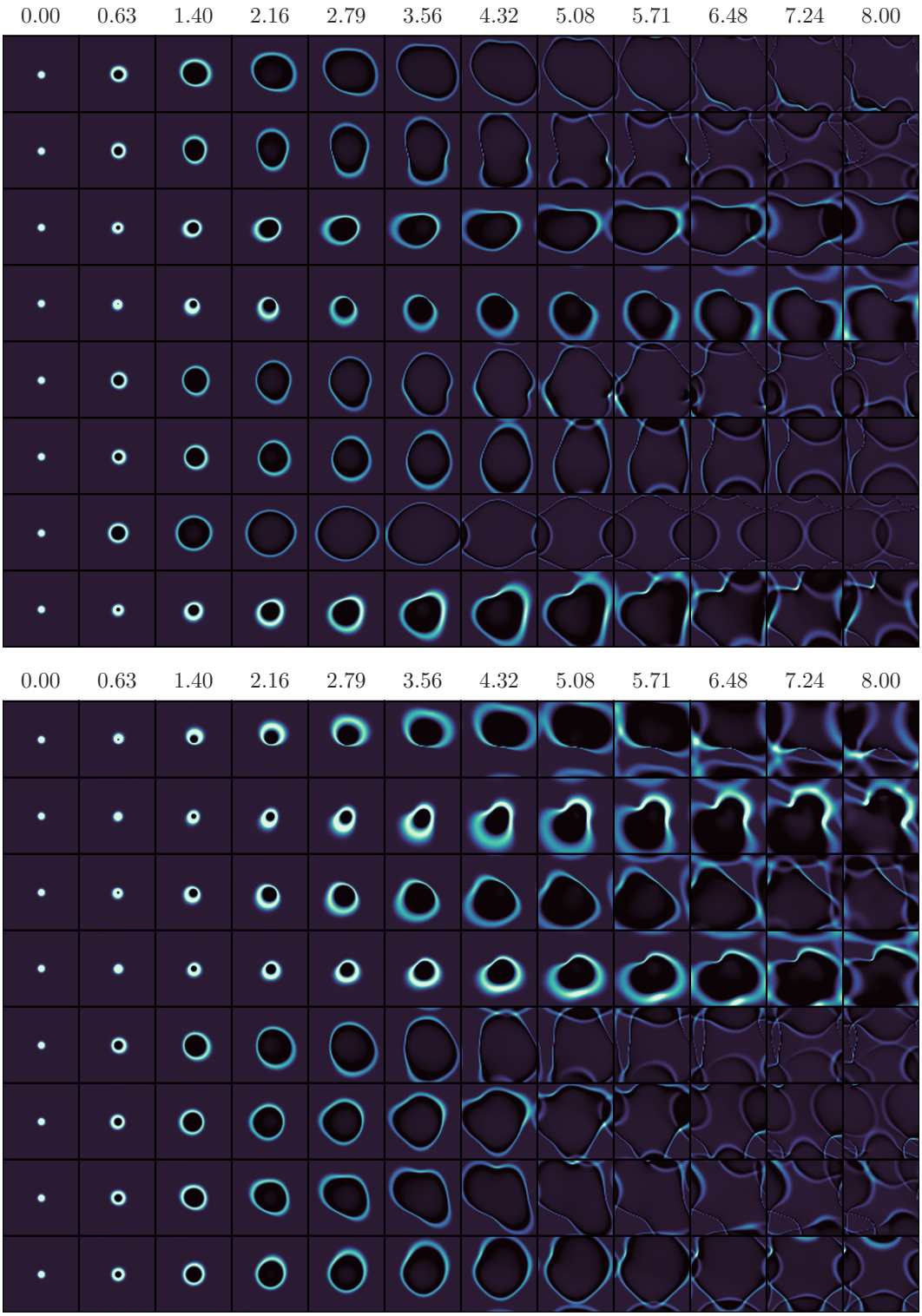


Figure 8: **Wave samples (uncurated)**. **Top**: true samples. **Bottom**: generated from Stochastic Lifting.

G.2 Flow

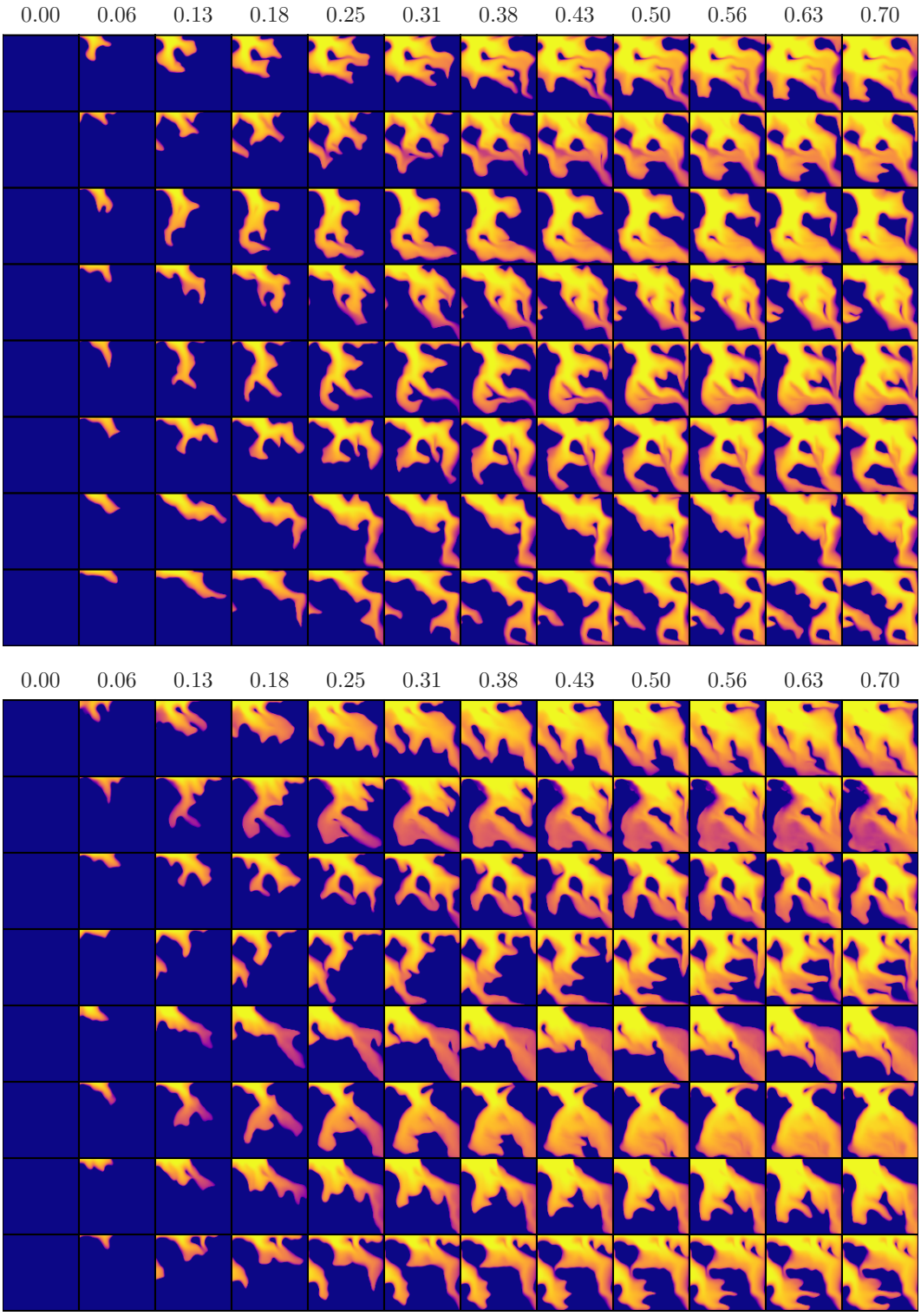


Figure 9: **Flow samples (uncurated)**. **Top**: true samples. **Bottom**: generated from Stochastic Lifting.

G.3 BAIR

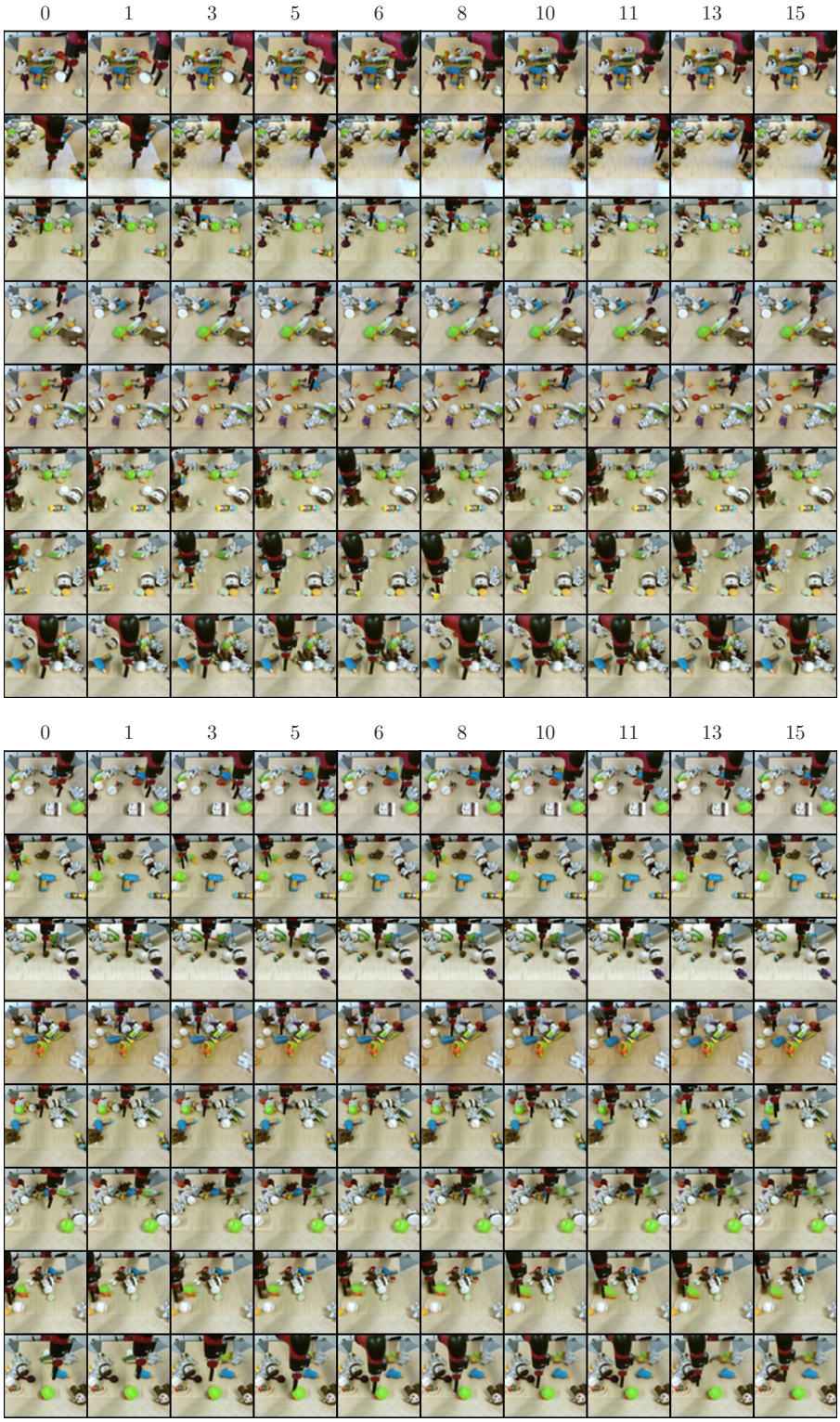


Figure 10: **BAIR samples (uncurated)**. **Top**: true samples. **Bottom**: generated from Stochastic Lifting.

G.4 CLEVRER

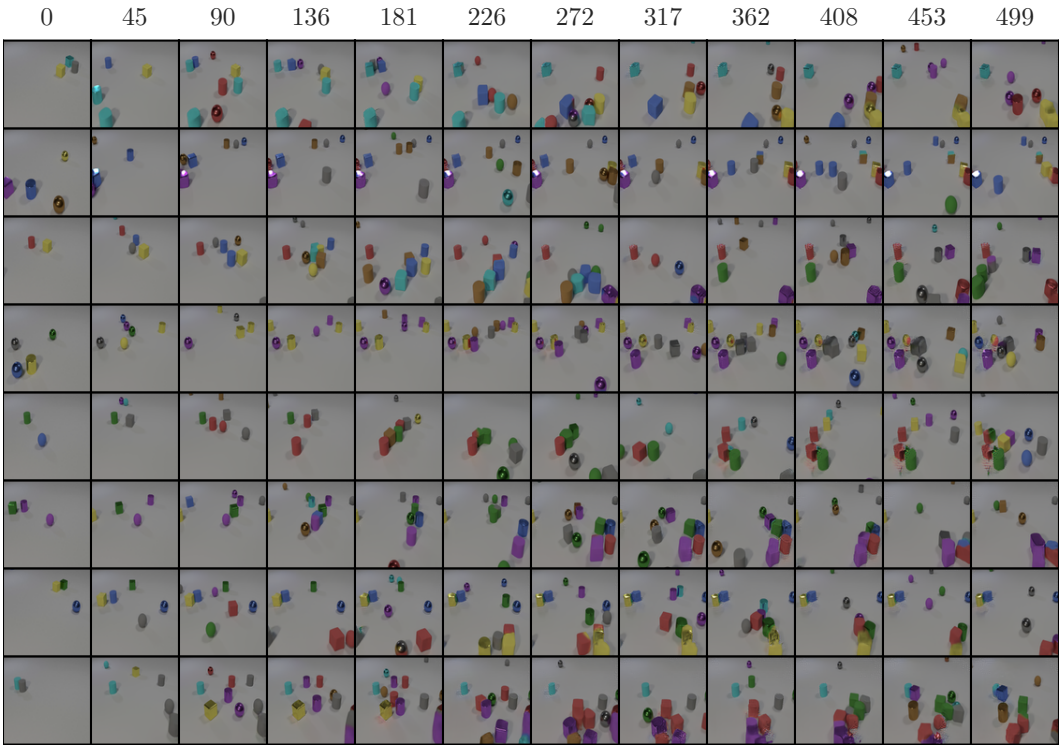


Figure 11: **CLEVRER samples (uncurated)**. Generated from Stochastic Lifting. Long rollout.

G.5 Polar bear walking on snow

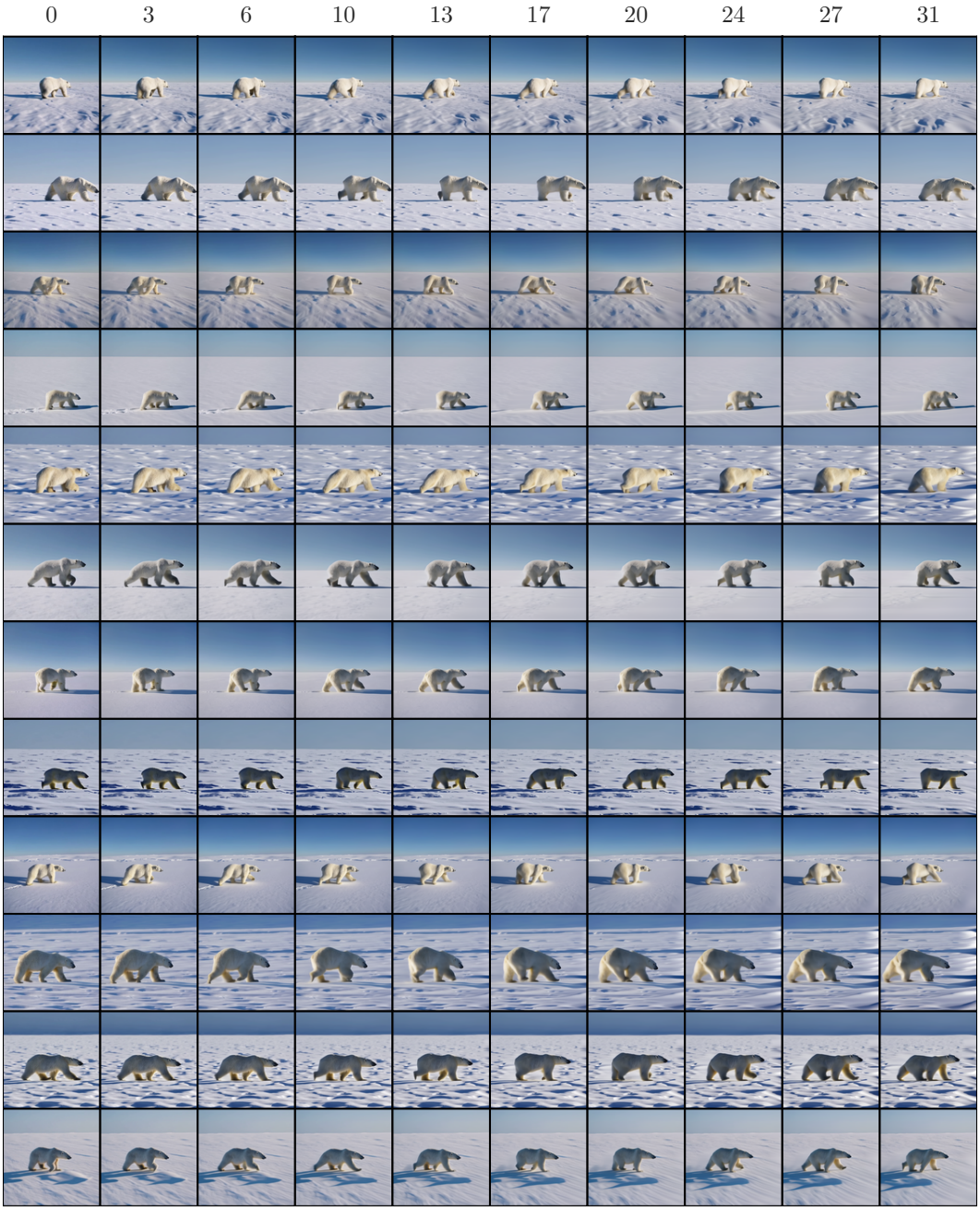


Figure 12: **POLAR samples (uncurated)**. Generated from Stochastic Lifting. 480×480 resolution.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Claims made:

   - Stochastic labeling achieves competitive results in video generation tasks. This is shown in the numerical experiments section 3, in particular tables 1 and 2.
   - Stochastic labeling works with off-the-shelf architectures (c.f. appendix E) and is easy to implement (c.f. algorithms 1 and 2).
   - Stochastic labeling is fast because it is a one-step method. The numerical experiments, in particular on the polar bear data set, confirm this.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in section 4.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Proofs of the propositions are given in the appendix.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Hyperparameters used (Table 3 and a description of the algorithm (algorithms 1 and 2) are provided in the paper. The code used to generate the numerical results will be published with the paper, along with the used data sets - if they are not already publicly available.

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

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

   Answer: [Yes]

   Justification: Code and data will be made available upon publication.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: A description of hyperparameters and architecture used (Table 3) is provided in the paper. All datasets are described in more detail in the appendix.

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: All quantities of interest that are reported in the numerical experiments are population-type quantities (Wasserstein distances, FVD evaluated on a large number of samples.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide information about runtime, computational complexity (number of function evaluation, optimizer iterations), and hardware used in the numerical experiments.

9. **Code of ethics**

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

   Answer: [Yes]

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We address the implications in the conclusion.

11. **Safeguards**

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

    Answer: [NA]

    Justification: All experiments on the data are done on either standard benchmark data sets, synthetic ones, or physical system data. Natural video generation is already comercially available. Therefore, we do not believe that the method and data used in this work have a high risk for misuse.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We provide a reference for every tool used in this work.

13. **New assets**

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

    Answer: [NA]

    Justification: No new assets are released in this work.

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

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

    Answer: [NA]

    Justification: Neither crowdsourcing experiments nor research with human subjects are present.

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

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

Answer: [NA]

Justification: This work did not deal with human subjects.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: Core method development in this research did not involve LLMs.