Video Latent Flow Matching: Optimal Polynomial Projections for Video Interpolation and Extrapolation

Yang Cao
Wyoming Seminary
ycao4@wyomingseminary.org

Zhao Song
Simons Institute for the Theory of Computing
at the University of California, Berkeley
magic.linuxkde@gmail.com

Chiwun Yang
Sun Yat-sen University
christiannyang37@gmail.com

Abstract

This paper considers an efficient video modeling process called Video Latent Flow Matching (VLFM). Unlike prior works, which randomly sampled latent patches for video generation, our method relies on current strong pre-trained image generation models, modeling a certain caption-guided flow of latent patches that can be decoded to time-dependent video frames. We first speculate multiple images of a video are differentiable with respect to time in some latent space. Based on this conjecture, we introduce the HiPPO framework to approximate the optimal projection for polynomials to generate the probability path. Our approach gains the theoretical benefits of the bounded universal approximation error and timescale robustness. Moreover, VLFM processes the interpolation and extrapolation abilities for video generation with arbitrary frame rates. We conduct experiments on several text-to-video datasets to showcase the effectiveness of our method.

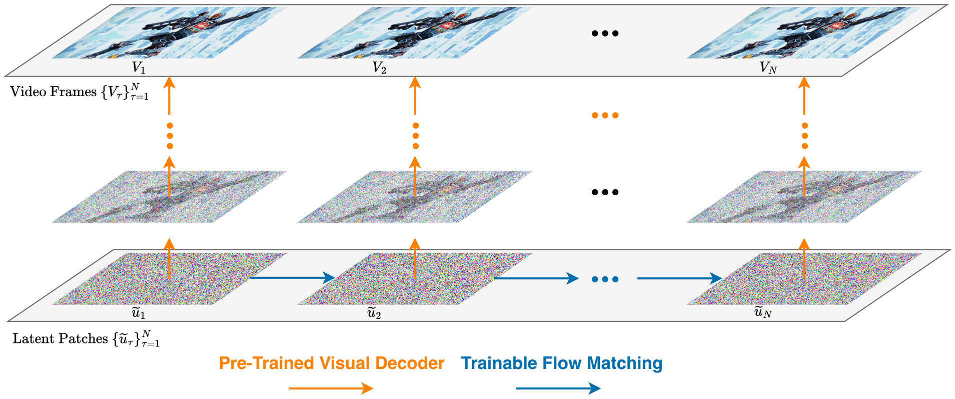

V_1
V_2
V_N
Video Frames $\{V_\tau\}_{\tau=1}^N$
$\widetilde{u}_1$
$\widetilde{u}_2$
$\widetilde{u}_N$
Latent Patches $\{\widetilde{u}_\tau\}_{\tau=1}^N$
Pre-Trained Visual Decoder
Trainable Flow Matching

Figure 1: Illustration of the working mechanism behind *Video Latent Flow Matching*.

1 Introduction

The rise of generative models has already demonstrated excellent performance in various fields like image generation Saharia et al. (2022); Rombach et al. (2022), text generation Achiam et al. (2023); Dubey et al. (2024); Liu et al. (2024), video generation Brooks et al. (2024); Zheng et al. (2024); Jin et al. (2024); Tian et al. (2024), etc. Suno-AI. Among them, some of the most popular algorithms - Flow Matching Lipman et al. (2022); Liu et al. (2022), Diffusion Ho et al. (2020); Song et al. (2020a) and VAEs Kingma & Welling (2013), perform surprise generative capabilities, however, requiring comprehensive computational resources for training. In particular, this efficiency drawback harms the development of more successful text-to-video modeling Brooks et al. (2024), becoming a frontier challenge in the field of generative modeling.

The prior works about the generation from textual descriptions to realistic and coherent videos usually involve two strong pre-trained networks Ho et al. (2022b); Zheng et al. (2024). One encodes input captions to rich embedding representations, and another one decodes from sequences of latent patches (also considered as Gaussian noise) under the guidance of text embedding representations. Although variants based on such modeling processes are already showing some fine initial results, the necessity of training on large-scale models and datasets leads these studies to be undemocratic Brooks et al. (2024); Kong et al. (2024). In response to this issue, the motivation of this paper is to design a novel algorithm to simplify the process of text-to-video modeling.

In this paper, we propose *Video Latent Flow Matching* (VLFM), which relies on the most advanced pre-trained image generation models (we call visual decoder in the range of this paper) for their extension in the field of text-to-video generation. In detail, we first introduce a deterministic inversion algorithm Song et al. (2020a); Lipman et al. (2022); Liu et al. (2022) to the visual decoder and apply this inversion operation to the frames of all videos, obtaining a sequence including initial latent patches from each video. Thus, the base of this paper is that a sequence of latent patches is a time-dependent and caption-conditional flow, so-called *Video Latent Flow*. Therefore, we use Flow Matching Lipman et al. (2022); Liu et al. (2022) to model it.

Especially, we emphasize four advantages of our VLFM:

- **Modeling efficiency.** The modeling of VLFM only needs to fit N flows where N is the size of the training dataset. This computational requirement is close to training a text-to-image model.
- **Optimal polynomial projections.** We use discrete HiPPO LegS to generate the time-dependent flow with provable optimal polynomial projections. The approximating error decreases with the enlarging order of polynomials.
- **Arbitrary frame rate.** The reason for applying Flow Matching instead of other approaches is that it allows solving ODE with arbitrary time t. This further leads to precise video generation with high frame rates.
- **Interpolation and extrapolation.** Besides, VLFM is suitable for interpolation and extrapolation for high-precision video recovery and generation since its generalization performance is confirmed in our theoretical part.

In summary, we make the following contributions:

- We give this paper's preliminary as a theoretical background with several mild assumptions in Section 2. Hence, we state the derivation of our VLFM in Section 3, which introduces the HiPPO framework to online approximate the sequence of latent patches.
- The theoretical benefits of VLFM are shown in Section 4. We first utilize the universal approximation theorem of Diffusion Transformer (DiT) to ensure an appropriate learner for modeling. The approximation bound then is guaranteed. We also discuss how our VLFM processes interpolation and extrapolation to real-world videos with an upper bound on error and its timescale robustness.
- We validate our approach by conducting extensive experiments in Section 5. Our model leverages DiT-XL-2 and is trained on a diverse collection of seven large-scale video datasets, including OpenVid-1M, MiraData, and videos from Pixabay. The results demonstrate strong performance in text-to-video generation, interpolation, and extrapolation, achieving robust and reliable outputs with significant potential for real-world video applications.

2 Preliminary

In this section, we formalize the background of this paper. We first introduce how we invert video frames into some latent space using the strong pre-trained visual decoder in Section 2.1. We state the definition of data and assumption in Section 2.2. Section 2.3 defines the main problem we aim to address in this paper.

2.1 Inverting Video Frames to Latent Patches

Notations. We use D to denote the flattened dimension of real-world images. We use d to represent the dimension of latent patches. We introduce d_0 as the dimension of Diffusion Transformers. We utilize $V : [0, T] \to \mathbb{R}^D$ to denote a video with T duration, where T is the longest time for each video. We omit $\nabla_t a(t)$ and $a'(t)$ to denote taking differentiation to some function $a(t)$ w.r.t. time t. We use integer s to denote the order of polynomials. The dimensional number of the text embedding vector is given by integer ℓ.

Visual decoder. Here we denote the visual decoder $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ satisfies that: For any flattened image $V \in \mathbb{R}^D$, there is a unique $u \in \mathbb{R}^d$ such that $\mathcal{D}(u) = V$. Then we say $\mathcal{D}$ is bijective. Denote the reverse function of $\mathcal{D}$ as $\mathcal{D}^{-1} : \mathbb{R}^D \to \mathbb{R}^d$. Note that this visual decoder $\mathcal{D}$ could be considered as any generative algorithm practically, e.g. LDM Rombach et al. (2022), DDIM Song et al. (2020a) and VAE Kingma (2013). We thus implement an inversion algorithm to invert video frames to latent patches Mokady et al. (2023). In particular, we define these latent patches here, which depend on the detailed visual decoder. We consider these latent patches following arbitrary distribution.

We abuse the notation $u : [0, T] \to \mathbb{R}^d$ to denote a sequence of latent patches of a video V. In detail, we define: $u_t := \mathcal{D}^{-1}(V_t)$ for any $t \in [0, T]$.

Discretization for cases of real-world data. We denote Δt as the minimal time unit of measurement in the real world (Planck time). Hence, a video V with T duration can be divided into at most $\frac{T}{\Delta t}$ frames. We use matrix $\widetilde{V} \in \mathbb{R}^{\frac{T}{\Delta t} \times D}$ to denote the compact form of discretized video. We use $\Phi \in \{0, 1\}^{\frac{T}{\Delta t} \times N}$ for $N \leq \frac{T}{\Delta t}$ to denote the corresponding observation matrix due to the real-world consideration, especially $\Phi^\top \mathbf{1}_{\frac{T}{\Delta t}} = \mathbf{1}_N$. Then the practical form of latent patches is given by:

$$\widetilde{u}_\tau := \mathcal{D}^{-1}([\Phi \widetilde{V}]_\tau) \in \mathbb{R}^d, \forall \tau \in [N]. \tag{1}$$

2.2 Data and Assumptions

Caption-video data pairs. Given a video distribution $\mathcal{V}$, we introduce a text embedding state distribution $\mathcal{C}$ that maps one-to-one to $\mathcal{V}$. Then for any video data $V \sim \mathcal{V}$, $c \in \mathbb{R}^\ell$ is denoted as the corresponding caption embedding state vector. We use $\mathcal{V}_c$ to denote the distribution that contains video and embedding vector, such that $(V, c) \sim \mathcal{V}_c$.

Assumptions. Here we list several mild assumptions in this paper, such that:

- **k-differentiable latent patches u.** We assume $u : [0, T] \to \mathbb{R}^d$ is a differentiable function with order k.
- **Lipschitz smooth visual decoder function $\mathcal{D}$.** We assume the visual decoder function $\mathcal{D}$ is L_0-smooth for constant $L_0 > 0$. Formally, it is: $\|\mathcal{D}(x) - \mathcal{D}(y)\|_2 \leq L_0 \cdot \|x - y\|_2, \forall x, y \in \mathbb{R}^d$.
- **Bounded entries of u.** For each entry in latent patches u, we assume it is smaller than a upper bound U for some constant $U > 0$.
- **Caption-to-latency function.** For any video-caption data $(V, c) \sim \mathcal{V}_c$, there exists a function $\mathcal{M} : [0, T] \times \mathbb{R}^\ell \to \mathbb{R}^D$ satisfies $V_t = \mathcal{M}_t(c)$.

2.3 Problem definition: Modeling Text-to-Latency Data from Discretized video

In this paper, we consider the video modeling problems as follows:

- Given a video-caption pair $(\mathcal{V}, c) \sim \mathcal{V}_c$, we obtain the data $\widetilde{u}_\tau \in \mathbb{R}^d, \forall \tau \in [N]$ via Eq. (1), we aim to find a algorithm that inputs a time $t \in [0, T]$ and encoded text state vector $c \in \mathbb{R}^\ell$ and output a predicted latent patch $\widehat{u}_t \in \mathbb{R}^d$, it satisfies:

  $$\|\mathcal{D}(\widehat{u}_t) - V_t\|_p \leq \epsilon. \tag{2}$$

  Here we denote the error $\epsilon \geq 0$ and some ℓ_p norm metric.

Connecting the main problem to interpolation and extrapolation. Since the frames number N of obtained video data may be greatly smaller than $T/\Delta t$. Recovering the continuous video data T as completely as possible (both interpolation and extrapolation) would also be our goal in the range of this paper. Theoretically, we see such interpolation and extrapolation as one: given a discrete video data $\Phi\widetilde{V} \in \mathbb{R}^{N\times D}$, the sequence of latent patches is $\widetilde{u} = [\widetilde{u}_1^\top, \widetilde{u}_2^\top, \cdots, \widetilde{u}_N^\top] \in \mathbb{R}^{N\times d}$ using Eq. (1). The text embedding state vector $c \in \mathbb{R}^\ell$ could be ensured by some video-to-caption methods. Our target is to find an algorithm that inputs $\widetilde{u}$ and outputs $\widehat{u}_\tau, \tau \in [T/\Delta t]$ that meets the requirement: $\|\mathcal{D}(\widehat{u}_\tau) - \widetilde{V}_\tau\|_p \le \epsilon$ for error $\epsilon \ge 0$ and some ℓ_p norm metric.

3 Video Latent Flow Matching

In this section, we propose Video Latent Flow Matching (VLFM) in response to the main problem in Section 2.3. Especially, we briefly review the HiPPO (high-order polynomial projection operators) framework Gu et al. (2020) in Section 3.1. We state the derivation of our VLFM based on the popular flow matching approach Lipman et al. (2022) in Section 3.2. Finally, we define the training objective of the VLFM for efficient video modeling in Section 3.3.

3.1 HiPPO Framework and LegS State Space Model

Given an input function $f(t) \in \mathbb{R}$ for $t \ge 0$, we use $f_{\le t}$ to denote the cumulative history of $f(t)$ for every time $t \ge 0$. We choose integer $s \ge 1$ as the order of approximation. Then, any s-dimensional subspace $\mathcal{G}$ of this function space is a suitable candidate for the approximation. Given a time-varying measure family $p(t)$ supported on $(-\infty, t)$, a sequence of basis functions $\mathcal{G} = \mathrm{span}\{g_i(t)\}_{i=1}^s$. HiPPO Gu et al. (2020) defines an operator that maps f to the optimal projection coefficients $c : \mathbb{R}_{\ge 0} \to \mathbb{R}^s$, such that:

$$g(t) := \arg\min_{g\in\mathcal{G}} \|f_{\le t} - g\|_{p(t)},$$

$$g(t) = \sum_{i=1}^{s} c_i(t) \cdot g_i(t).$$

We focus on the case where the coefficients $c(t)$ have the form of a linear ODE satisfying $\nabla c(t) = A(t)c(t) + B(t)f(t)$ for some $A(t) \in \mathbb{R}^{s\times s}$ and $B(t) \in \mathbb{R}^{s\times 1}$. This equation is now also known as the state space model (SSM) in many works Kantas et al. (2015); Gu et al. (2022); Gu & Dao (2023); Dao & Gu (2024); Zhu et al. (2024); Xing et al. (2024); Ma et al. (2024); Ruan & Xiang (2024); Sun et al. (2024).

Discrete HiPPO-LegS. The setting of HiPPO-LegS defines the update rule of SSM and the discrete version of A and B matrices, which are $c_{\tau+1} = (I_s - \frac{A}{\tau})c_\tau + \frac{1}{\tau}Bf_\tau$ and:

$$\begin{aligned} A_{i_1,i_2} &= \begin{cases} \sqrt{(2i_1+1)(2i_2+1)}, & \text{if } i_1 > i_2 \\ i_1 + 1, & \text{if } i_1 = i_2 \\ 0, & \text{if } i_1 < i_2 \end{cases}, \\ B_{i_1} &= \sqrt{2i_1+1}, \forall i_1, i_2 \in [s]. \end{aligned}$$

3.2 Conditional Video Latent Flow

Here, we emphasize that the core idea of VLFM is to approximate a continuous video distribution from limited discrete video frame data utilizing the optimal high-order polynomial approximation.

Given a video-caption distribution $\mathcal{V}_c$, then for any video-caption data pair $(V, c) \sim \mathcal{V}_c$, we obtain the data $\widetilde{u}_\tau \in \mathbb{R}^d, \forall \tau \in [N]$ via Eq. (1). We aim to define a time-dependent flow $\psi_t(\widetilde{u})$ that takes inputs $\widetilde{u}$ and time t, and could match $\widehat{u}_\tau$ for all time $\tau \in [N]$. Since $\widehat{u}$ is discrete, HiPPO-LegS will be the best solution to approximate the continuous data. We define the *Video Latent Flow* as:

$$\psi_t(\widetilde{u}) := \sigma_t(\widetilde{u}) \cdot z + \mu_t(\widetilde{u}) \in \mathbb{R}^d, \tag{3}$$

where $t \in [0, T]$ and $z \sim \mathcal{N}(0, I_d)$, $\sigma : [0, T] \times \mathbb{R}^{N\times d} \to \mathbb{R}_{>0}$ denotes the time-dependent standard deviation, where $\sigma_0(\widetilde{u}) = 1$, and $\sigma_{\frac{T}{N}\cdot\tau}(\widetilde{u}) = \sigma_{\min}$, for all $\tau \in [N]$; $\mu : [0, T] \times \mathbb{R}^{N\times d} \to \mathbb{R}^d$

denotes the time-dependent mean of Gaussian distribution, where $\mu_0(\widetilde{u}) = \mathbf{0}_d$, $\mu_{\frac{T}{N}\cdot\tau}(\widetilde{u}) = \widetilde{u}_\tau$, for all $\tau \in [N]$.

Especially, we define:

$$\mu_t(\widetilde{u}) := H_N g(t),$$
$$H_{\tau+1} := H_\tau(I_s - \frac{1}{\tau}A)^\top + \frac{1}{\tau}\widetilde{u}_\tau B^\top,$$

where $g(t) := [\sqrt{\frac{1}{2}}P_0(t), \sqrt{\frac{3}{2}}P_1(t), \cdots, \sqrt{\frac{2s-1}{2}}P_{s-1}(t)]^\top \in \mathbb{R}^s$, $P_i(t), \forall i \in [s]$ is Legendre polynomials. We initialize $H_0 := \mathbf{0}_{d\times s}$.

Besides, having a large scalar $\alpha > 0$, we give:

$$\sigma_t(\widetilde{u}) := (1 - \sigma_{\min}) \cdot [\sin^2(\pi\frac{N}{T}t) + \exp(-\alpha t)] + \sigma_{\min}.$$

3.3 TRAINING OBJECTIVE

Here we define a model function $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ with parameters θ to learn the conditional video latent flow $\psi_t(\widetilde{u})$ defined in Eq. (3). This function takes inputs of flow and time to predict the vector field. The training objective is based on the Flow Matching framework Lipman et al. (2022), which aims to minimize the distance between the model's prediction and the true derivative of the flow.

The training objective of VLFM is defined as the expectation of the square ℓ_2 norm of the difference, which is:

$$\mathcal{L}(\theta) := \mathop{\mathbb{E}}_{z,t,(V,c)}[\|F_\theta(\psi_t(\widetilde{u}), c, t) - \frac{\mathrm{d}}{\mathrm{d}t}\psi_t(\widetilde{u})\|_2^2],$$

where $z \sim \mathcal{N}(0, I_d)$, $t \sim \mathsf{Uniform}[0, T]$ and $(V, c) \sim \mathcal{V}_c$. By minimizing this objective, the model learns to approximate the vector field that transports the initial noise distribution to the distribution of video latent patches. Formally, we solve: $\min_\theta \mathcal{L}(\theta)$.

Close-form solution. Furthermore, the close-form solution could be easily obtained as follows:

Theorem 3.1. *The minimum solution for function F_θ that takes $z \sim N(0, I_d)$ and $t \sim \mathsf{Uniform}[0, T]$ is:*

$$F_\theta(z, c, t) = \frac{\sigma_t'(\widetilde{u})}{\sigma_t(\widetilde{u})}(z - \mu_t(\widetilde{u})) + \mu_t'(\widetilde{u}).$$

Proof. This proof follows from Theorem 3 in Lipman et al. (2022). □

4 THEORY

This section provides several theoretical advantages of our VLFM. The approximation theory in this approach builds up based on using the Diffusion Transformer (DiT) Peebles & Xie (2023), which is a popular choice in previous empirical and theoretical part generative model works Chen et al. (2023); Hu et al. (2024e), we briefly state its definitions in Section 4.1.

In addition, we provide the optimal polynomial projection guarantee and universal approximation theorem (with DiT) of VLFM in Section 4.2 to confirm its approximating ability. Besides, Section 4.3 gives error bound of interpolation and extrapolation, and Section 4.4 gives the supplementary property that VLFM's timescale robustness, which indicates its theoretical advantages.

4.1 DIFFUSION TRANSFORMER (DIT)

Diffusion Transformer Peebles & Xie (2023) is a framework that utilizes Transformers Vaswani et al. (2017) as the backbone for Diffusion Models Ho et al. (2020); Song et al. (2020a). Specifically, a Transformer block consists of a multi-head self-attention layer and a feed-forward layer, with both layers having a skip connection. We use $\mathsf{TF}^{h,m,r} : \mathbb{R}^{n \times d_0} \to \mathbb{R}^{n \times d_0}$ to denote a Transformer block. Here h and m are the number of heads and head size in self-attention layer, and r is the hidden dimension in feed-forward layer. Let $X \in \mathbb{R}^{n \times d_0}$ be the model input. Then, we have the model output:

$$\mathsf{Attn}(X) := \sum_{i=1}^{h} \mathsf{Softmax}(X W_Q^i {W_K^i}^\top X^\top) \cdot X W_V^i {W_O^i}^\top + X,$$

where the projection weights $W_K^i, W_Q^i, W_V^i, W_O^i \in \mathbb{R}^{d_0 \times m}$. Moreover,

$$\mathsf{FF}(X) := \phi(X W_1 + \mathbf{1}_n b_1^\top) \cdot W_2^\top + \mathbf{1}_n b_2^\top + X.$$

where the projection weights $W_1, W_2 \in \mathbb{R}^{d_0 \times r}$, bias $b_1 \in \mathbb{R}^r, b_2 \in \mathbb{R}^{d_0}$, and ϕ is usually considered as the ReLU activated function.

In our work, we use Transformer networks with positional encoding $E \in \mathbb{R}^{n \times d_0}$. The transformer networks are then defined as the composition of Transformer blocks:

$$\mathcal{T}_P^{h,m,r} = \{ f_\mathcal{T} : \mathbb{R}^{n \times d_0} \to \mathbb{R}^{n \times d_0} \mid f_\mathcal{T} \text{ is a composition of blocks } \mathsf{TF}^{h,m,r}\text{'s} \}.$$

For instance, the following is a Transformer network consisting L blocks and positional encoding

$$f_\mathcal{T}(X) = \mathsf{FF}^{(L)} \circ \mathsf{Attn}^{(L)} \circ \cdots \mathsf{FF}^{(1)} \circ \mathsf{Attn}^{(1)}(X + E).$$

4.2 Approximation via DiT

Before we state the approximation theorem, we define a reshaped layer that transforms concatenated input in flow matching into a length-fixed sequence of vectors. It is denoted as $R : \mathbb{R}^{d+\ell+1} \to \mathbb{R}^{n \times d_0}$. Therefore, in the following, we give the theorem utilizing DiT to minimize training objective $\mathcal{L}(\theta)$ to arbitrary error.

Theorem 4.1 (Informal version of Theorem G.7)**.** *There exists a transformer network $f_\mathcal{T} \in \mathcal{T}_P^{2,1,4}$ defining function $F_\theta(z, c, t) := f_\mathcal{T}(R([z^\top, c^\top, t]^\top))$ with parameters θ that satisfies $\mathcal{L}(\theta) \le \epsilon$ for any error $\epsilon > 0$.*

Proof sketch of Theorem 4.1. Please refer to the proof of Theorem G.7 for the detailed analysis. □

4.3 Interpolation and Extrapolation

Now, we theoretically discuss the approximating error of our VLFM in processing interpolation and extrapolation. It is considered a recovery of the original idea data from limited sub-sampled observations. This analysis is achieved by splitting the error into three parts, which are: 1) approximating error ϵ_1 for HiPPO-LegS approximating the original data; 2) Gaussian error ϵ_2 for the boundary of Gaussian vector z; 3) interpolation and extrapolation error ϵ_3 that represents the training and predicting the difference between using original idea data V and limited sub-sampled observations $\Phi \widetilde{V}$. We state the results as follows:

Lemma 4.2 (Informal version of Lemma H.3)**.** *Denote failure probability $\delta \in (0, 0.1)$. Let the flow $\psi_t(\widetilde{u})$ defined in Eq. (3). Denote $G := [g(\Delta t), g(2\Delta t), \cdots, g(T)]^\top \in \mathbb{R}^{\frac{T}{\Delta t} \times s}$ and $\lambda^* := \lambda_{\min}(G) > 0$ as the minimum eigenvalue of G. Choosing $s = O(\frac{\Delta t}{T} \log((\frac{\Delta t}{T})^{1.5} \lambda^*))$. Denote $u_t = \mathcal{D}(V_t)$ for any $t \in [0, T]$. Especially, we define:*

- *Approximating error $\epsilon_1 := O(T^k s^{-k+1/2})$.*
- *Gaussian error $\epsilon_2 := O(\sqrt{d \log(d/\delta)})$.*
- *Interpolation and extrapolation error $\epsilon_3 := U d^{0.5} \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s)) / \lambda^*$.*

Then with a probability at least $1 - \delta$, we have:

$$\| \psi_t(\widetilde{u}) - u_t \|_2 \le \epsilon_1 + \epsilon_2 + \epsilon_3.$$

Proof. Proof sketch of Lemma 4.2 This proof follows from its formal version in Lemma H.3 □

Having Lemma 4.2, the concise bound for solving Eq. (2) could be given below:

Theorem 4.3 (Informal version of Theorem H.4)**.** *Following Theorem 4.1, denote failure probability* $\delta \in (0, 0.1)$ *and arbitrary error* $\epsilon_0 > 0$*. Then with a probability at least* $1 - \delta$*, the network in Theorem 4.1 satisfies Eq.* (2) *with* $p = 2$ *and*

$$\epsilon = \epsilon_0 + L_0(\epsilon_1 + \epsilon_2 + \epsilon_3).$$

Proof sketch of Theorem 4.3. Please refer to Theorem H.4 for complete proofs. □

Discussions. Following the results of Lemma 4.2 and Theorem 4.3, we thus derive few insights as follows:

- **Optimal choice of s: A trade-off between ϵ_1 and ϵ_3.** As shown in the conditions of Lemma 4.2, the larger value of the order of polynomials s helps to decrease approximating error in the training dataset while also ruining the generalization ability.
- **Stable visual decoder.** Theorem 4.3 shows a small value of L_0 (the stability and smoothness of visual decoder), which is important for the error of interpolation and extrapolation with an arbitrary frame rate.
- **Information.** Besides, a sub-linear factor $\sqrt{\frac{T}{\Delta t} - N}$, which stands for the obtained information about the continuous video, is vital as well for interpolation and extrapolation on data in distribution.

4.4 Timescale Robustness

Following Gu et al. (2020), we demonstrate that projection onto latent patches u_t is robust to timescales. Formally, the HiPPO-LegS operator is *timescale-equivariant*: dilating the input u does not change the approximation coefficients H_N. At the same time, this property is working in the case of the discretized form $\widetilde{u}$. We emphasize that it is crucial to use flow matching to model the latent patches, where whatever the sampling method and frame rate are, it will not greatly harm VLFM's performance. We give its formal statement below.

Lemma 4.4 (Proposition 3 of Gu et al. (2020), informal version of Lemma H.2)**.** *For any integer scale factor* $\beta > 0$*, the frames of video* $\widetilde{V}_\tau$ *is scaled to* $\widetilde{V}_{\beta\tau}$ *for each* $\tau \in [\frac{T}{\Delta t}]$*, it doesn't affect the result of* H_N*.*

Proof. This lemma follows from Proposition 3 in Gu et al. (2020). □

5 Experiments

In this section, we conduct experiments to evaluate the effectiveness of our approach. We first introduce our experimental setups in Section 5.1. Then, we demonstrate text-to-video generation using VLFM and VLFM's capability of generating videos in arbitrary frame rate in Section 5.2. Furthermore, we showcase the strong performance of interpolation and extrapolation of VLFM in Section 5.3. We also perform an ablation study to discuss the importance of the flow matching algorithm in Section 5.4.

5.1 Setup

In our experiments, we apply Stable Diffusion v1.5 Rombach et al. (2022) with DDIM scheduler Song et al. (2020a) as the visual decoder. Then, we use a DiT-XL-2 Peebles & Xie (2023) as the backbone for the Flow Matching algorithm Lipman et al. (2022); Liu et al. (2022), and the choice of hyper-parameters of $\sigma_t(\widetilde{u})$ is given by $\sigma_{\min} = 0.01$ and $\alpha = 10$. We optimize the DiT using Grams optimizer Cao et al. (2024). We sample and combine 7 data resources for comprehensive training and validation of our method. They are: OpenVid-1M Nan et al. (2024), UCF-101 Soomro et al. (2012), Kinetics-400 Kay et al. (2017), YouTube-8M Abu-El-Haija et al. (2016), InternVid Wang et al. (2023), MiraData Ju et al. (2024), and Pixabay Pixabay.

5.2 Text-to-Video Generation with Arbitrary Frame Rate

In this section, we recover several videos with different frame rates using VLFM with given video captions in the training dataset. We extract $T = 0.5$ for demonstrations as Figure 2. In detail, we choose three frame rates for generation $\{8, 12, 16\}$. As shown, our VLFM performs fairly on text-to-video generation while it requires very small resource that is equivalent to training a new flow matching text-to-image video, which ensures its efficiency. Moreover, we give more results that are generated by VLFM in Appendix C and D.

5.3 Interpolation and Extrapolation

In this section, we test the interpolation and extrapolation of VLFM. For the interpolation experiment, the model is trained with 24 FPS and evaluated to generate video with 48 FPS. For the extrapolation, the model is trained with the first video with $T = 2$ and evaluated to generate the whole video with $T = 8$. Referring the results in Figure 3, this demonstrates the strong performance of our VLFM under our mathematical guarantee of the error bound and its effectiveness.

5.4 Ablation Study

In this section, we compared training VLFM with the Flow Matching algorithm and directly used DiT to predict the latent patches to showcase the importance of utilizing flow matching in our VLFM. We compare VLFM with and without flow matching by training the model with 1000 steps and compare the PSNR (peak signal-to-noise ratio) before and after training for video recovery with given captions in the training dataset. We state the results in Table 1. Denote $\mathrm{MSE}(x, y)$ as the mean squared error function, the computation of the metric PSNR is given by ($x, y \in \mathbb{R}^{r\times r}$):

$$\mathrm{PSNR}(x, y) := 10 \log_{10}\left(\frac{r^2}{\mathrm{MSE}(x, y)}\right),$$

Table 1: PSNR comparison (the greater, the better) of Flow Matching and direct generation from DiT. We boldface the better scores.

Algorithm	Initial PSNR↑	Final PSNR↑
Flow Matching	**57.20**	**61.18**
Direct Predicting	9.81	53.77

6 Conclusion

This paper proposes *Video Latent Flow Matching* (VLFM) for efficient training of a time-varying flow to approximate the sequence of latent patches of the obtained video. This approach is confirmed to enjoy theoretical benefits, including 1) universal approximation theorem via applying Diffusion Transformer architecture and 2) optimal polynomial projections and timescale by introducing HiPPO-LegS. Furthermore, we provide the generalization error bound of VLFM that is trained only on the limited sub-sampled video to interpolate and extrapolate the whole ideal video. We evaluate our VLFM on Stable Diffusion v1.5 with DDIM scheduler and the DiT-XL-2 model with datasets OpenVid-1M, UCF-101, Kinetics-400, YouTube-8M, InternVid, MiraData, and Pixabay. The experimental results validated the potential of our approach to become a novel and efficient training form for text-to-video generation.

Limitations. Since the motivation of this paper focuses on simply and efficiently solving the main goal, it lacks enough exploring each design and how it affects the empirical performance, providing little insights for the follow-ups. Hence, we leave these comprehensive explorations, and its more concise theoretical working mechanism behind as future works. On the other hand, although VLFM simplifies the video modeling process, it necessitates additional computational consumption concerning the combination of the visual decoder part and the flow matching part at the inference stage. We also leave such exploration to a more efficient inference method as a future direction.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful suggestions.

REFERENCES

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, context-free grammar. *arXiv preprint arXiv:2305.13673*, 2023.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pp. 242–252. PMLR, 2019.

Josh Alman and Zhao Song. Fast attention requires bounded entries. *Advances in Neural Information Processing Systems*, 36, 2024a.

Josh Alman and Zhao Song. The fine-grained complexity of gradient computation for training large language models. *arXiv preprint arXiv:2402.04497*, 2024b.

Josh Alman, Zhao Song, Ruizhe Zhang, and Danyang Zhuo. Bypass exponential time preprocessing: Fast neural network training via weight-data correlation preprocessing. *Advances in Neural Information Processing Systems*, 36, 2023.

Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *arXiv preprint arXiv:2304.08477*, 2023.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Heli Ben-Hamu, Samuel Cohen, Joey Bose, Brandon Amos, Aditya Grover, Maximilian Nickel, Ricky TQ Chen, and Yaron Lipman. Matching normalizing flows and probability paths on manifolds. *arXiv preprint arXiv:2207.04711*, 2022.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. *arXiv preprint arXiv:2304.02207*, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL `https://openai.com/research/video-generation-models-as-world-simulators`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Yang Cao, Xiaoyu Li, and Zhao Song. Grams: Gradient descent with adaptive momentum scaling. *arXiv preprint arXiv:2412.17107*, 2024.

Yang Cao, Bo Chen, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. Force matching with relativistic constraints: A physics-inspired approach to stable and efficient generative modeling. *arXiv preprint arXiv:2502.08150*, 2025.

Bo Chen, Xiaoyu Li, Yingyu Liang, Jiangxuan Long, Zhenmei Shi, and Zhao Song. Circuit complexity bounds for rope-based transformer architecture. *arXiv preprint arXiv:2411.07602*, 2024a.

Bo Chen, Xiaoyu Li, Yingyu Liang, Jiangxuan Long, Zhenmei Shi, and Zhao Song. Circuit complexity bounds for rope-based transformer architecture. *arXiv preprint arXiv:2411.07602*, 2024b.

Bo Chen, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Hsr-enhanced sparse attention acceleration. *arXiv preprint arXiv:2410.10165*, 2024c.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pp. 4672–4712. PMLR, 2023.

Timothy Chu, Zhao Song, and Chiwun Yang. How to protect copyright data in optimization of large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17871–17879, 2024.

Majid Daliri, Zhao Song, and Chiwun Yang. Unlocking the theory behind scaling 1-bit neural networks. *arXiv preprint arXiv:2411.01663*, 2024.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Yichuan Deng, Hang Hu, Zhao Song, Omri Weinstein, and Danyang Zhuo. Training overparametrized neural networks in sublinear time. *arXiv preprint arXiv:2208.04508*, 2022.

Yichuan Deng, Zhihang Li, and Zhao Song. Attention scheme inspired softmax regression. *arXiv preprint arXiv:2304.10411*, 2023a.

Yichuan Deng, Zhao Song, Shenghao Xie, and Chiwun Yang. Unmasking transformers: A theoretical approach to data recovery via attention weights. *arXiv preprint arXiv:2310.12462*, 2023b.

Yichuan Deng, Zhao Song, and Chiwun Yang. Attention is naturally sparse with gaussian distributed input. *arXiv preprint arXiv:2404.02690*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Yeqi Gao, Zhao Song, Weixin Wang, and Junze Yin. A fast optimization view: Reformulating single layer attention in llm based on tensor and svm trick, and solving it in matrix multiplication time. *arXiv preprint arXiv:2309.07418*, 2023.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *arXiv preprint arXiv:2407.15595*, 2024.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22930–22941, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022.

Jiaxi Gu, Shicong Wang, Haoyu Zhao, Tianyi Lu, Xing Zhang, Zuxuan Wu, Songcen Xu, Wei Zhang, Yu-Gang Jiang, and Hang Xu. Reuse and diffuse: Iterative denoising for text-to-video generation. *arXiv preprint arXiv:2309.03549*, 2023.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David P Woodruff, and Amir Zandieh. Hyperattention: Long-context attention in near-linear time. *arXiv preprint arXiv:2310.05869*, 2023.

Eric Heitz, Laurent Belcour, and Thomas Chambon. Iterative α-(de) blending: A minimalist deterministic diffusion model. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–8, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

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, 2022b.

Jerry Yao-Chieh Hu, Donglin Yang, Dennis Wu, Chenwei Xu, Bo-Yu Chen, and Han Liu. On sparse modern hopfield model. *Advances in Neural Information Processing Systems*, 36, 2023.

Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Robin Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. *arXiv preprint arXiv:2404.03828*, 2024a.

Jerry Yao-Chieh Hu, Thomas Lin, Zhao Song, and Han Liu. On computational limits of modern hopfield models: A fine-grained complexity analysis. *arXiv preprint arXiv:2402.04520*, 2024b.

Jerry Yao-Chieh Hu, Wei-Po Wang, Ammar Gilani, Chenyang Li, Zhao Song, and Han Liu. Fundamental limits of prompt tuning transformers: Universality, capacity and efficiency. *arXiv preprint arXiv:2411.16525*, 2024c.

Jerry Yao-Chieh Hu, Dennis Wu, and Han Liu. Provably optimal memory capacity for modern hopfield models: Transformer-compatible dense associative memories as spherical codes. *arXiv preprint arXiv:2410.23126*, 2024d.

Jerry Yao-Chieh Hu, Weimin Wu, Zhao Song, and Han Liu. On statistical rates and provably efficient criteria of latent diffusion transformers (dits). *arXiv preprint arXiv:2407.01079*, 2024e.

Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein backbone generation. *arXiv preprint arXiv:2405.20313*, 2024.

Hui Jiang. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *arXiv preprint arXiv:2407.06358*, 2024.

Nikolas Kantas, Arnaud Doucet, Sumeetpal S. Singh, Jan Maciejowski, and Nicolas Chopin. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3), 2015. ISSN 0883-4237. doi: 10.1214/14-sts511. URL `http://dx.doi.org/10.1214/14-STS511`.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. In *International Conference on Algorithmic Learning Theory*, pp. 597–619. PMLR, 2023.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36, 2024.

Chenyang Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Tianyi Zhou. Fourier circuits in neural networks: Unlocking the potential of large language models in mathematical reasoning and modular arithmetic. *arXiv preprint arXiv:2402.09469*, 2024a.

Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. A tighter complexity analysis of sparsegpt. *arXiv preprint arXiv:2408.12151*, 2024b.

Xiaoyu Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Fine-grained attention i/o complexity: Comprehensive analysis for backward passes. *arXiv preprint arXiv:2410.09397*, 2024c.

Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, Zhuoyan Xu, and Junze Yin. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. *arXiv preprint arXiv:2405.05219*, 2024a.

Yingyu Liang, Jiangxuan Long, Zhenmei Shi, Zhao Song, and Yufa Zhou. Beyond linear approximations: A novel pruning approach for attention matrix. *arXiv preprint arXiv:2410.11261*, 2024b.

Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Multi-layer transformers gradient can be approximated in almost linear time. *arXiv preprint arXiv:2408.13233*, 2024c.

Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Looped relu mlps may be all you need as practical programmable computers. *arXiv preprint arXiv:2410.09375*, 2024d.

Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Tensor attention training: Provably efficient learning of higher-order transformers. *arXiv preprint arXiv:2405.16411*, 2024e.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Jian-wei LIU, Jun-wen LIU, and Xiong-lin LUO. Research progress in attention mechanism in deep learning. *Chinese Journal of Engineering*, 43(11):1499–1511, 2021.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.

Alexander Munteanu, Simon Omlor, Zhao Song, and David Woodruff. Bounding the width of neural networks via coupled initialization a worst case analysis. In *International Conference on Machine Learning*, pp. 16083–16122. PMLR, 2022.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Pixabay. Pixabay - stunning royalty-free images & royalty-free stock. URL `https://pixabay.com`.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

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*, pp. 10684–10695, 2022.

Noam Rozen, Aditya Grover, Maximilian Nickel, and Yaron Lipman. Moser flow: Divergence-based generative modeling on manifolds. *Advances in Neural Information Processing Systems*, 34: 17669–17680, 2021.

Jiacheng Ruan and Suncheng Xiang. Vm-unet: Vision mamba unet for medical image segmentation. *arXiv preprint arXiv:2402.02491*, 2024.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3531–3539, 2021.

Zhenmei Shi, Yifei Ming, Xuan-Phi Nguyen, Yingyu Liang, and Shafiq Joty. Discovering the gems in early layers: Accelerating long-context llms with 1000x input token reduction. *arXiv preprint arXiv:2409.17422*, 2024.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Zhao Song, Shuo Yang, and Ruizhe Zhang. Does preprocessing help training over-parameterized neural networks? *Advances in Neural Information Processing Systems*, 34:22890–22904, 2021.

Zhao Song, Junze Yin, and Lichen Zhang. Solving attention kernel regression problem via pre-conditioner. In *International Conference on Artificial Intelligence and Statistics*, pp. 208–216. PMLR, 2024a.

Zhao Song, Lichen Zhang, and Ruizhe Zhang. Training multi-layer over-parametrized neural network in subquadratic time. *ITCS*, 2024b.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Suno-AI. Bark: Text-prompted generative audio model. URL `https://github.com/suno-ai/bark`.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Jan van den Brand, Binghui Peng, Zhao Song, and Omri Weinstein. Training (overparametrized) neural networks in near-linear time. *ITCS*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

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.

Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*, 2023.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Dennis Wu, Jerry Yao-Chieh Hu, Teng-Yun Hsiao, and Han Liu. Uniform memory retrieval with larger capacity for modern hopfield models. *arXiv preprint arXiv:2404.03827*, 2024a.

Weimin Wu, Maojiang Su, Jerry Yao-Chieh Hu, Zhao Song, and Han Liu. Transformers are deep optimizers: Provable in-context learning for deep model training. *arXiv preprint arXiv:2411.16549*, 2024b.

Zhaohu Xing, Tian Ye, Yijun Yang, Guang Liu, and Lei Zhu. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 578–588. Springer, 2024.

Zhuoyan Xu, Zhenmei Shi, and Yingyu Liang. Do large language models have compositional ability? an investigation into limitations and scalability. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024a.

Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. Towards few-shot adaptation of foundation models via multitask finetuning. *arXiv preprint arXiv:2402.15017*, 2024b.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? *arXiv preprint arXiv:1912.10077*, 2019.

Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. Kdeformer: Accelerating transformers via kernel density estimation. In *International Conference on Machine Learning*, pp. 40605–40623. PMLR, 2023.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

Appendix

CONTENTS

Roadmap. In the appendix, we first introduce related work in Appendix A. We supplement the missing results of Section 5.2 and Section 5.3 in Appendix B. Then, we present more experimental text-to-video generation results in Appendix C and more interpolation and extrapolation results in Appendix D. Next, we introduce the preliminary in Appendix E. Moreover, we illustrate Video Latent Flow Matching formally in Appendix F. In Appendix G, we demonstrate the Diffusion Transformer, and finally, in Appendix H, we present the interpolation and extrapolation of VLFM.

A Related Work

This section briefly reviews three topics that are closely related to this work: Text-to-Video Generation, Flow Matching, and Theory in Transformer-Based Models.

Text-to-Video Generation. Text-to-video generation Singer et al. (2022); Voleti et al. (2022); Blattmann et al. (2023) is a specialized form of conditional video generation that aims to synthesize high-quality videos from textual descriptions. Recent advancements in this field have predominantly leveraged diffusion models Song et al. (2020b); Ho et al. (2020), which iteratively refine video frames by learning to denoise samples from a normal distribution. This approach has proven effective in generating coherent and visually appealing videos. Training strategies for text-to-video models vary widely. One common approach involves adapting pre-trained text-to-image models by incorporating temporal modules, such as temporal convolutions and attention mechanisms, to establish inter-frame relationships Ge et al. (2023); An et al. (2023); Singer et al. (2022); Gu et al. (2023); Guo et al. (2023). For instance, PYoCo Ge et al. (2023) introduced a noise prior technique and utilized the pre-trained eDiff-I model Balaji et al. (2022) as a starting point. Alternatively, some methods build on Stable Diffusion Rombach et al. (2022), leveraging its accessibility and pre-trained capabilities to expedite convergence Blattmann et al. (2023); Zhou et al. (2022). However, this approach can sometimes result in suboptimal outcomes due to the inherent distributional differences between images and videos. Another strategy involves training models from scratch on combined image and video datasets Ho et al. (2022a), which can yield superior results while requiring intensive computationally.

Flow Matching. Flow Matching has emerged as a highly effective framework for generative modeling, demonstrating significant advancements across various domains, including video generation. Its simplicity and power have been validated in large-scale generation tasks such as image Esser et al. (2024), video Polyak et al. (2024); Jin et al. (2024), speech Le et al. (2024), audio Vyas et al. (2023), proteins Huguet et al. (2024), and robotics Black et al. (2024). Flow Matching originated from efforts to address the computational challenges associated with Continuous Normalizing Flows (CNFs), where early methods struggled with simulation inefficiencies Rozen et al. (2021); Ben-Hamu et al. (2022). Modern Flow Matching algorithms Lipman et al. (2022); Liu et al. (2022); Albergo & Vanden-Eijnden (2022); Neklyudov et al. (2023); Heitz et al. (2023); Tong et al. (2023); Cao et al. (2025) have since evolved to learn CNFs without explicit simulation, significantly improving scalability. Recent innovations, such as Discrete Flow Matching Campbell et al. (2024); Gat et al. (2024), have further expanded the applicability of this framework, making it a versatile tool for generative tasks.

Theory in Transformer-Based Models. Transformers have become a cornerstone in AI and are widely used in different areas, especially in NLP (Natural Language Process) and CV (Computer Vision). However, understanding the Transformers from a theoretical perspective remains an ongoing challenge. Several works have explored the theoretical foundations and computational complexities of the Transformers Tsai et al. (2019); Zandieh et al. (2023); Brand et al. (2023); Alman & Song (2024a); Song et al. (2024a); Chen et al. (2024b); Hu et al. (2024b); Munteanu et al. (2022); Song et al. (2024b); Allen-Zhu et al. (2019); Deng et al. (2022); van den Brand et al. (2021); Song et al. (2021); Alman et al. (2023); Deng et al. (2023b) focusing on areas such as efficient Transformers Han et al. (2023); Shi et al. (2024); Shen et al. (2021); LIU et al. (2021); Liang et al. (2024a;e;c); Li et al. (2024b); Liang et al. (2024b); Chen et al. (2024c); Li et al. (2024c); Hu et al. (2024e;d;a); Wu et al. (2024a); Hu et al. (2023); Alman & Song (2024b); Gao et al. (2023), optimization Deng et al. (2023a); Chu et al. (2024), and the analysis of emergent abilities Brown et al. (2020); Wei et al. (2022); Allen-Zhu & Li (2023); Jiang (2023); Xu et al. (2024b); Li et al. (2024a); Xu et al. (2024a); Chen et al. (2024a); Liang et al. (2024d); Hu et al. (2024c); Wu et al. (2024b); Deng et al. (2024). Notably, Zandieh et al. (2023); Daliri et al. (2024) introduced an algorithm with provable guarantees for approximation of Transformers, Keles et al. (2023) proved a lower bound for Transformers based on the Strong Exponential Time Hypothesis, and Alman & Song (2024a) provided both an algorithm and hardness results for static Transformers computation.

B Supplementary Results of Section 5.2 and Section 5.3

We give supplementary experiment results of Section 5.2 and Section 5.3 in Figure 2 and Figure 3 respectively.

C More Text-to-Video Generation Results

We give more text-to-video generation results with different frame rates to demonstrate the generative ability of our VLFM in Figure 4 and Figure 5.

D More Interpolation and Extrapolation Results

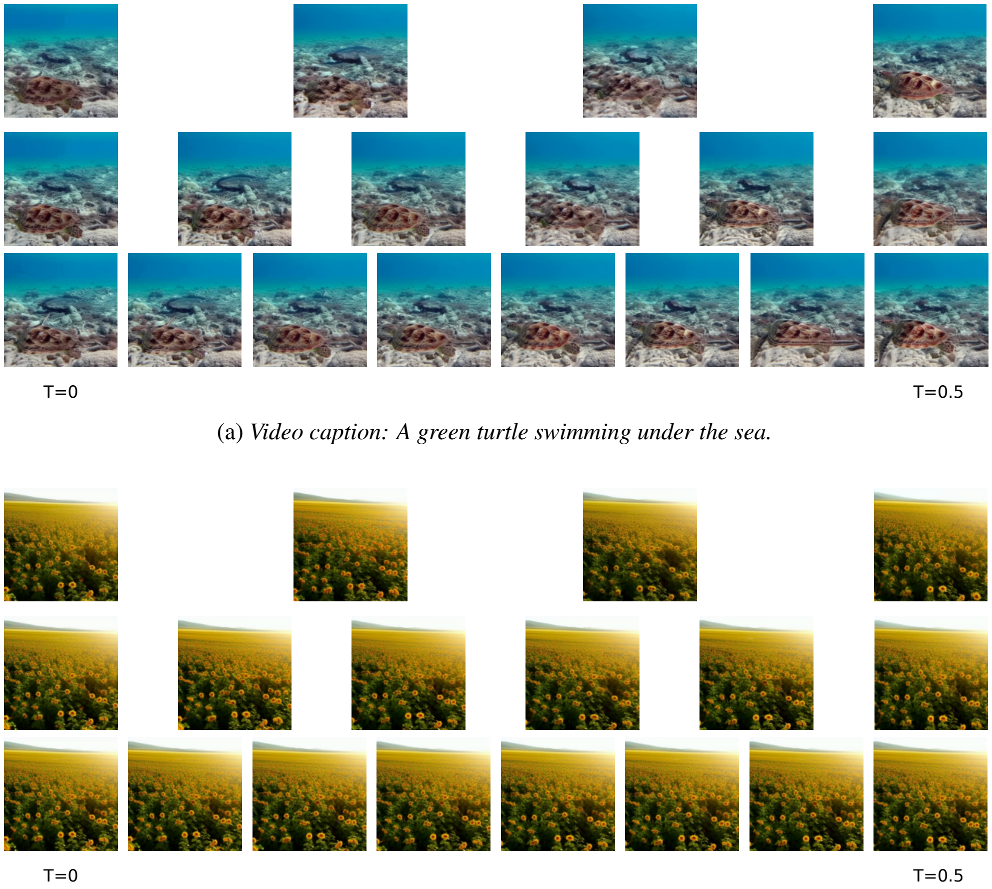
We give more results of interpolation and extrapolation of VLFM in Figure 6.

E Preliminary

In the preliminary section, we first introduce our notation in the appendix in Appendix E.1. Then, in Appendix E.2, we formally define the video-caption data and visual decoder. In Appendix E.3, we define the latent patches. Appendix E.4 makes some assumptions which we will use later. Finally, in Appendix E.5, we list some basic useful facts.

E.1 Notations

Notations. We use D to denote the flattened dimension of real-world images. We use d to represent the dimension of latent patches. We introduce d_0 as the dimension of Diffusion Transformers. We utilize $V : [0, T] \to \mathbb{R}^D$ to denote a video with T duration, where T is the longest time for each video. We omit $\nabla_t a(t)$ and $a'(t)$ to denote taking differentiation to some function $a(t)$ w.r.t. time t. We use integer s to denote the order of polynomials. The dimensional number of the text embedding vector is given by integer ℓ.



(a) *Video caption: A green turtle swimming under the sea.*

(b) *Video caption: Viewing countless sunflowers in a field from top.*

Figure 2: Generated videos with different frame rates $\{8, 12, 16\}$.

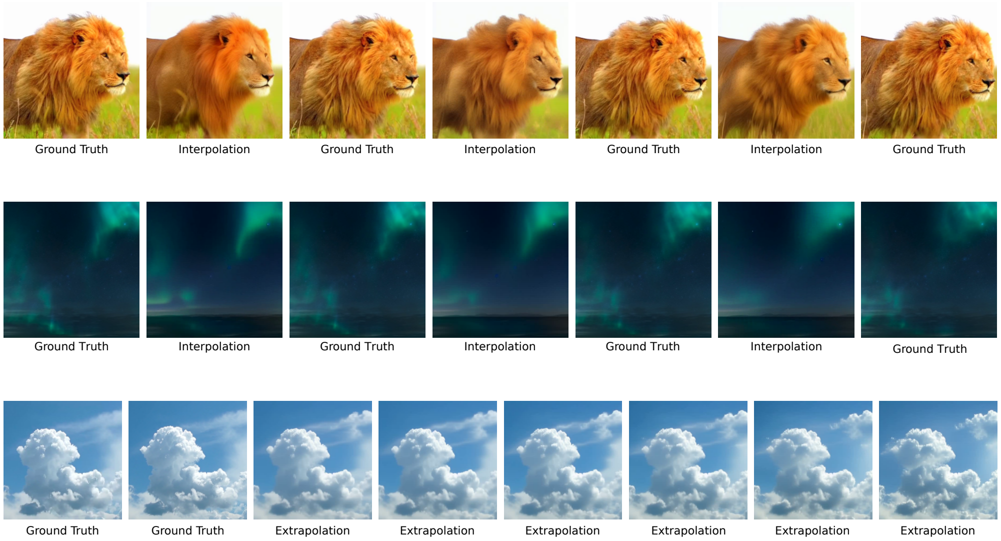
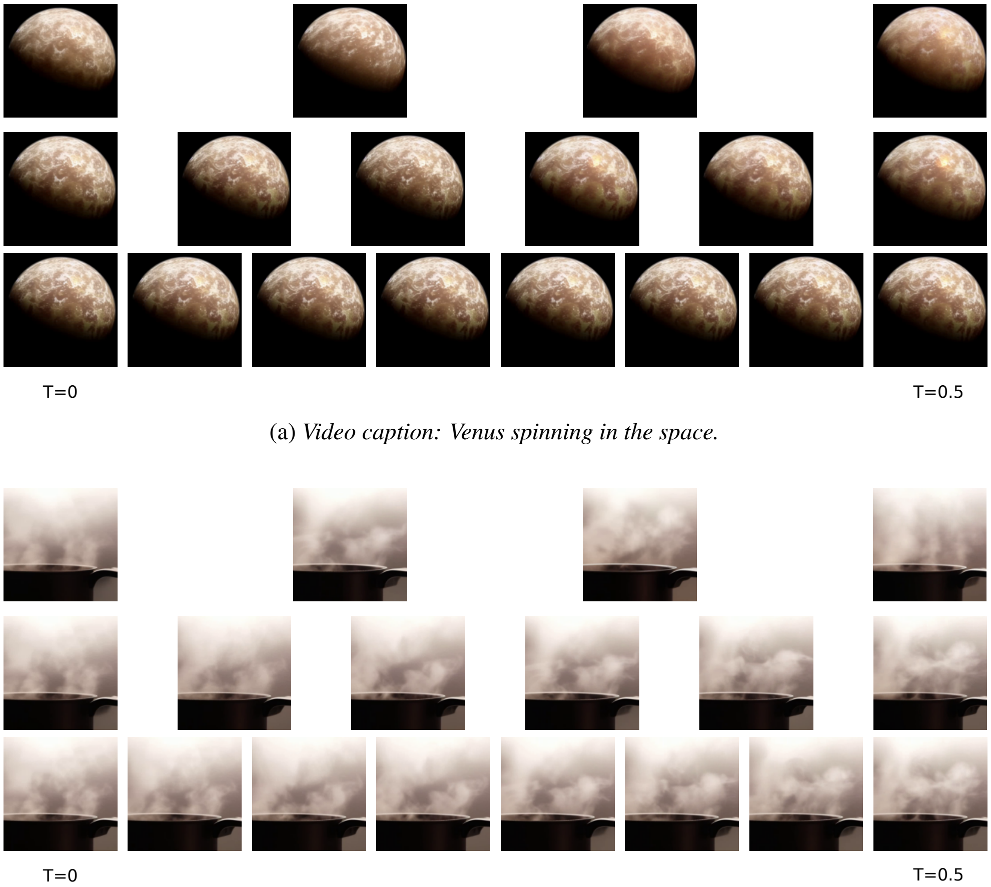


Figure 3: Interpolation and Extrapolation of VLFM.



(a) *Video caption: Venus spinning in the space.*

(b) *Video caption: Steam is coming out of a pot.*

Figure 4: Generated videos with different frame rates $\{8, 12, 16\}$.

E.2 Video-Caption Data

Definition E.1 (Video-caption data pairs and their distribution)**.** *We define a video caption distribution $(V, c) \sim \mathcal{V}_c$. Here, $V : [0, T] \to \mathbb{R}^D$ is considered as a function and $c \in \mathbb{R}^\ell$ is the corresponding text embedding vector.*

Definition E.2. *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1. We denote Δt as the minimal time unit of measurement in the real world (Planck time). For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of $V : [0, T] \to \mathbb{R}^D$, which is $\widetilde{V} \in \mathbb{R}^{\frac{T}{\Delta t} \times D}$, and its τ-th row $\forall \tau \in [\frac{T}{\Delta t}]$ is given by:*

$$\widetilde{V}_\tau := V_{\Delta t \cdot \tau} \in \mathbb{R}^D.$$

Definition E.3 (Obtained data in real-world cases)**.** *If the following conditions hold:*

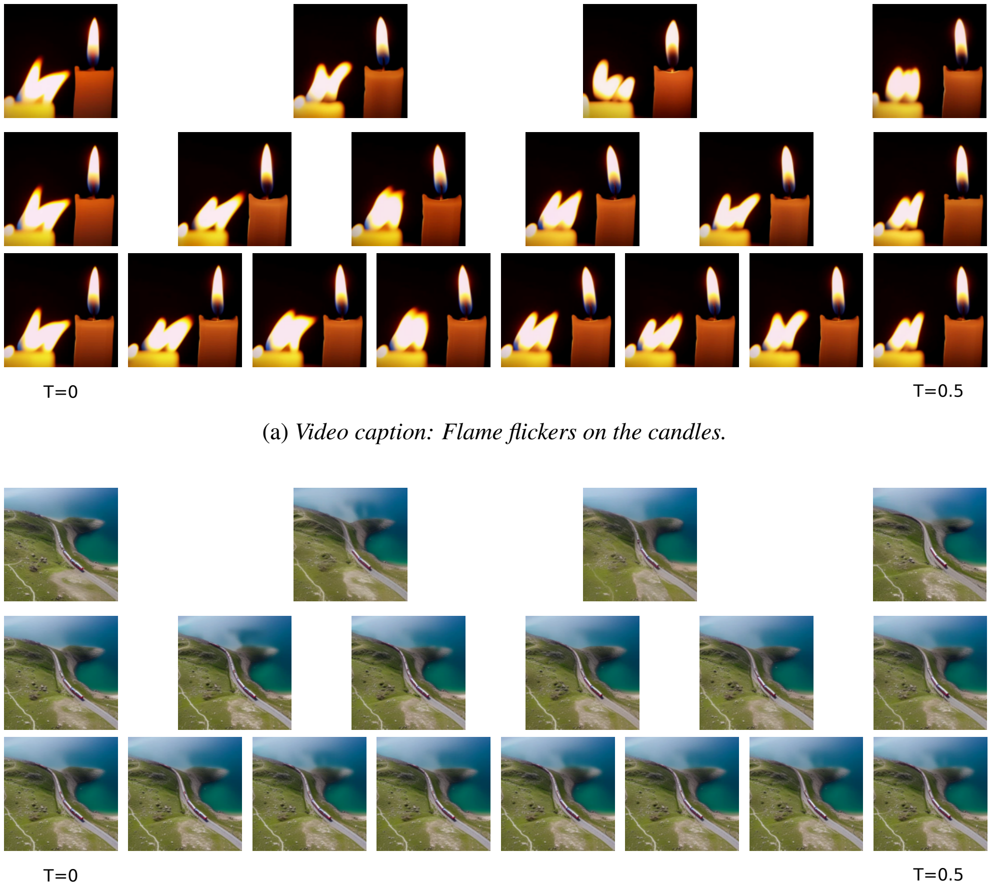
- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video $\widetilde{V}$ as Definition E.2.*

We define an observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$. The obtained data in real-world cases then is denoted as $\Phi\widetilde{V} \in \mathbb{R}^{N \times D}$.

Definition E.4 (Bijective Visual Decoder)**.** *We define the visual decoder $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ satisfies that:*

- *For any flattened image $V \in \mathbb{R}^D$, there is a unique $u \in \mathbb{R}^d$ such that $\mathcal{D}(u) = V$.*

Then we say $\mathcal{D}$ is bijective. Denote the reverse function of $\mathcal{D}$ as $\mathcal{D}^{-1} : \mathbb{R}^D \to \mathbb{R}^d$.



(a) *Video caption: Flame flickers on the candles.*

(b) *Video caption: A train is running through the rail road near the coast.*

Figure 5: Generated videos with different frame rates $\{8, 12, 16\}$.

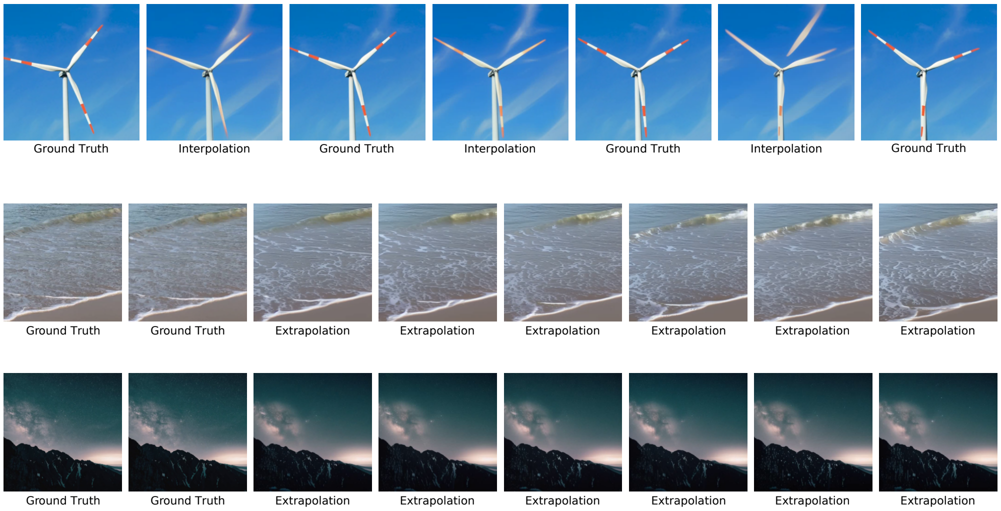


Figure 6: Interpolation and Extrapolation of VLFM.

E.3 Latent Patches Data

Definition E.5. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video $\widetilde{V}$ as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*

We define the ideal version (without observation matrix) of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$, and its τ-th $\forall \tau \in [\frac{T}{\Delta t}]$ row is defined as follows:

$$u_\tau := \mathcal{D}^{-1}(\widetilde{V}_\tau).$$

Definition E.6. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*

We define the real-world version (with observation matrix) of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$, and its τ-th $\forall \tau \in [N]$ row is defined as follows:

$$\widetilde{u}_\tau := \mathcal{D}^{-1}\Big((\Phi V)_\tau\Big).$$

E.4 Assumptions

Assumption E.7. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*

We assume u_τ is k-differentiable, there exists:

$$u_\tau^{(i)} = \lim_{\Delta t \to 0} \frac{u_{\tau+1}^{(i-1)} - u_\tau^{(i-1)}}{\Delta t}, \forall i \in [k], \tau \in [\frac{T}{\Delta t}],$$

where, we use $u_\tau^{(i)}$ to denote the i-th derivation of u.

Assumption E.8. *If the following conditions hold:*

- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*

We assume the visual decoder function $\mathcal{D}$ is L_0-smooth for constant $L_0 > 0$, such that:

$$\|\mathcal{D}(x) - \mathcal{D}(y)\|_2 \le L_0 \|x - y\|_2, \forall x, y \in \mathbb{R}^d.$$

Assumption E.9. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*

We assume each entry in latent patches u is bounded by a constant $U > 0$.

Assumption E.10. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$*

For any $(V, c) \sim \mathcal{V}_c$, we assume there exists a function $\mathcal{M} : [0, T] \times \mathbb{R}^\ell \to \mathbb{R}^D$ satisfies $V_t = \mathcal{M}_t(c)$.

E.5 Basic Facts

Fact E.11. *For a variable $x \sim \mathcal{N}(0, \sigma^2)$, then with probability at least $1 - \delta$, we have:*

$$|x| \leq C\sigma\sqrt{\log(1/\delta)}$$

Fact E.12. *For a PD matrix $A \in \mathbb{R}^{d_1 \times d_2}$ with a positive minimum eigenvalue $\lambda_{\min}(A) > 0$, the infinite norm of its pseudoinverse matrix $A^\dagger$ is given by:*

$$\|A^\dagger\|_\infty \leq \frac{1}{\lambda_{\min}(A)}.$$

Fact E.13. *For two matrices $A, B \in \mathbb{R}^{d_1 \times d_2}$, we have:*

$$\|A^\dagger - B^\dagger\| \leq C \max\{\|A^\dagger\|^2, \|B^\dagger\|^2\} \cdot \|A - B\|,$$

where $C > 0$ is some costant.

F Video Latent Flow Matching

This section, we first introduce the HiPPO Framework and LegS in Appendix F.1. Then, we formally define the video latent flow in Appendix F.2. Last, we introduce the training objective of VLFM in Appendix F.3.

F.1 HiPPO Framework and LegS

Definition F.1. *We define matrix $A \in \mathbb{R}^{s \times s}$ where its (i_1, i_2)-th entry $\forall i_1, i_2 \in [s]$ is given by:*

$$A_{i_1, i_2} = \begin{cases} \sqrt{(2i_1+1)(2i_2+1)}, & \textit{if } i_1 > i_2 \\ i_1 + 1, & \textit{if } i_1 = i_2 \\ 0, & \textit{if } i_1 < i_2 \end{cases}.$$

Definition F.2. *We define matrix $B \in \mathbb{R}^{s \times 1}$ where its i_1-th entry $\forall i_1 \in [s]$ is given by:*

$$B_{i_1} = \sqrt{2i_1 + 1}.$$

Definition F.3. *If the following conditions hold:*

- *Let matrix $A \in \mathbb{R}^{s \times s}$ be defined as Definition F.1.*
- *Let matrix $B \in \mathbb{R}^{s \times 1}$ be defined as Definition F.2.*

We initialize a matrix $H_0 = \mathbf{0}_{d\times s}$. Then we define:

$$H_\tau := H_{\tau-1}(I_s - \frac{1}{\tau}A)^\top + \frac{1}{\tau}\widetilde{u}_\tau B^\top, \forall \tau \in [N].$$

Definition F.4. *We define $g(t) := [\sqrt{\frac{1}{2}}P_0(t), \sqrt{\frac{3}{2}}P_1(t), \cdots, \sqrt{\frac{2s-1}{2}}P_{s-1}(t)]^\top \in \mathbb{R}^s$, where$P_i(t), \forall i \in [s]$ is some polynomials. Especially, $g(t)$ satisfies:*

- *Define $G := \begin{bmatrix} g(\Delta t)^\top \\ g(2\Delta t)^\top \\ \vdots \\ g(T)^\top \end{bmatrix}$, $\lambda_{\min}(G) > 0$. Here, $\lambda_{\min}$ is the function that outputs the minimal eigenvalue of the input matrix.*
- *$|G_{\tau,i}| \leq \exp(O(\frac{T}{\Delta t}s))$ for any $\tau \in [\frac{T}{\Delta t}], i \in [s]$.*

F.2 VIDEO LATENT FLOW

Definition F.5. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N\times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t}\times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N\times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d\times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ be defined as Definition F.4.*

We define the time-dependent mean of Gaussian distribution as follows:

$$\mu_t(\widetilde{u}) := H_N g(t) \in \mathbb{R}^d$$

Definition F.6. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N\times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t}\times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N\times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d\times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ be defined as Definition F.4.*
- *Denote $\sigma_{\min} > 0$.*

- *Given a hyper-parameter $\alpha > 0$.*

We define the time-dependent standard deviation as follows:

$$\sigma_t(\widetilde{u}) := (1 - \sigma_{\min}) \cdot [\sin^2(\pi \frac{N}{T} t) + \exp(-\alpha t)] + \sigma_{\min} \in \mathbb{R}_{\geq 0}.$$

Lemma F.7. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N \times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d \times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ be defined as Definition F.4.*
- *Let the time-dependent mean of Gaussian distribution $\mu_t(\widetilde{u})$ be defined as Definition F.5.*
- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*
- *Denote $\sigma_{\min} > 0$.*
- *Given a hyper-parameter $\alpha > 0$.*

Then for any $\alpha > 0$, we have:

$$|\frac{\sigma_t'(\widetilde{u})}{\sigma_t(\widetilde{u})}| \leq \frac{1 - \sigma_{\min}}{\sigma_{\min}}.$$

Proof. This result can be obtained following very simple algebras. □

Definition F.8. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N \times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d \times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ be defined as Definition F.4.*
- *Let the time-dependent mean of Gaussian distribution $\mu_t(\widetilde{u})$ be defined as Definition F.5.*
- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*

- *Denote* $\sigma_{\min} > 0$.
- *Sample* $z \sim \mathcal{N}(0, I_d)$.

We define the Video Latent Flow:

$$\psi_t(\widetilde{u}) := \sigma_t(\widetilde{u}) \cdot z + \mu_t(\widetilde{u}) \in \mathbb{R}^d.$$

F.3 Training Objective

Definition F.9. *If the following conditions hold:*

- *Given a video caption distribution* $\mathcal{V}_c$ *as Definition E.1.*
- *For any* $(V, c) \sim \mathcal{V}_c$*, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix* $\Phi : \{0,1\}^{N \times \frac{T}{\Delta t}}$ *be defined as Definition E.3.*
- *Let the visual decoder function* $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ *be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches* $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ *be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches* $\widetilde{u} \in \mathbb{R}^{N \times d}$ *be defined as Definition E.6.*
- *Let* $H_N \in \mathbb{R}^{d \times s}$ *be defined as Definition F.3.*
- *Let the function of polynomials* $g(t)$ *be defined as Definition F.4.*
- *Let the time-dependent mean of Gaussian distribution* $\mu_t(\widetilde{u})$ *be defined as Definition F.5.*
- *Let the time-dependent standard deviation* $\sigma_t(\widetilde{u})$ *be defined as Definition F.6.*
- *Denote* $\sigma_{\min} > 0$.
- *Sample* $z \sim \mathcal{N}(0, I_d)$.
- *Define a model function* $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ *with parameters* θ.

We define the training objective of Video Latent Flow Matching as follows:

$$\mathcal{L}(\theta) := \mathop{\mathbb{E}}_{z \sim \mathcal{N}(0, I_d), t \sim \mathsf{Uniform}[0,T], (V,c) \sim \mathcal{V}_c} [\|F_\theta(\psi_t(\widetilde{u}), c, t) - \frac{\mathrm{d}}{\mathrm{d}t}\psi_t(\widetilde{u})\|_2^2].$$

Theorem F.10. *If the following conditions hold:*

- *Given a video caption distribution* $\mathcal{V}_c$ *as Definition E.1.*
- *For any* $(V, c) \sim \mathcal{V}_c$*, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix* $\Phi : \{0,1\}^{N \times \frac{T}{\Delta t}}$ *be defined as Definition E.3.*
- *Let the visual decoder function* $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ *be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches* $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ *be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches* $\widetilde{u} \in \mathbb{R}^{N \times d}$ *be defined as Definition E.6.*
- *Let* $H_N \in \mathbb{R}^{d \times s}$ *be defined as Definition F.3.*
- *Let the function of polynomials* $g(t)$ *be defined as Definition F.4.*
- *Let the time-dependent mean of Gaussian distribution* $\mu_t(\widetilde{u})$ *be defined as Definition F.5.*

- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*
- *Denote $\sigma_{\min} > 0$.*
- *Sample $z \sim \mathcal{N}(0, I_d)$.*
- *Define a model function $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ with parameters θ.*
- *Let the training objective $\mathcal{L}(\theta)$ be defined as Definition F.9.*

Then the minimum solution for function F_θ that takes $z \sim N(0, I_d)$ and $t \sim \mathsf{Uniform}[0, T]$ is:

$$F_\theta(z, c, t) = \frac{\sigma_t'(\widetilde{u})}{\sigma_t(\widetilde{u})}(z - \mu_t(\widetilde{u})) + \mu_t'(\widetilde{u}).$$

Proof. This proof follows from Theorem 3 in Lipman et al. (2022). □

G Diffusion Transformer

In this section, we first define the Diffusion Transformer in Appendix G.1. Moreover, we introduce the Approximation via DiT in Appendix G.2.

G.1 Definitions

Definition G.1 (Multi-head self-attention)**.** *Given h-heads query, key, value and output projection weights $\{(W_Q^i, W_K^i, W_V^i, W_O^i)\}_{i=1}^h \subset \mathbb{R}^{d_0 \times 4m}$ with each weight is a $d_0 \times m$ shape matrix, for an input matrix $X \in \mathbb{R}^{n \times d_0}$, we define a multi-head self-attention computation as follows:*

$$\mathsf{Attn}(X) := \sum_{i=1}^h \mathsf{Softmax}(XW_Q^i {W_K^i}^\top X^\top) \cdot XW_V^i {W_O^i}^\top + X \in \mathbb{R}^{n \times d_0}.$$

Definition G.2 (Feed-forward)**.** *Given two projection weights $W_1, W_2 \in \mathbb{R}^{d_0 \times r}$ and two bias vectors $b_1 \in \mathbb{R}^r$ and $b_2 \in \mathbb{R}^{d_0}$, for an input matrix $X \in \mathbb{R}^{n \times d_0}$, we define a feed-forward computation as follows:*

$$\mathsf{FF}(X) := \phi(XW_1 + \mathbf{1}_n b_1^\top) \cdot W_2^\top + \mathbf{1}_n b_2^\top + X \in \mathbb{R}^{n \times d_0}.$$

Here, ϕ is an activation function and usually be considered as ReLU.

Definition G.3 (Transformer block)**.** *Given a set of model weights $\theta^{h,m,r} = \{\{(W_Q^i, W_K^i, W_V^i, W_O^i)\}_{i=1}^h,\ W_1, W_2, b_1, b_2\}$, the computation of a transformer block is given by the combination of multi-head self-attention computation (Definition G.1) and feed-forward computation (Definition G.2). Formally, for an input matrix $X \in \mathbb{R}^{n \times d_0}$, we define:*

$$\mathsf{TF}_{\theta^{h,m,r}}(X) := \mathsf{FF} \circ \mathsf{Attn}(X) \in \mathbb{R}^{n \times d_0}$$

Definition G.4 (Reshape Layer)**.** *We define the reshape network $R : \mathbb{R}^d \to \mathbb{R}^{n \times d_0}$.*

Definition G.5 (Complete transformer network)**.** *We consider a transformer network as a composition of a transformer block (Definition G.3) with model weight $\theta^{h,m,r}$, which is:*

$$\begin{aligned}
&\mathcal{T}^{h,m,r} \\
:= &\{\mathcal{F} : \mathbb{R}^{n \times d_0} \to \mathbb{R}^{n \times d_0} \\
&\mid \mathcal{F} \text{ is a composition of Transformer blocks } \mathsf{TF}_{\theta^{h,m,r}}\text{'s with positional embedding } E \in \mathbb{R}^{n \times d_0}\}
\end{aligned}$$

We especially say $\theta^{h,m,r}$ is the model weight that contains h heads, m hidden size for attention and r hidden size for feed-forward. See Example G.6 for further explanation of the sequence-to-sequence mapping $\mathcal{F}$.

Example G.6. *We here give an example for the sequence-to-sequence mapping $\mathcal{F}$ in Definition G.5: Denote L as the number of layers in some transformer network. For an input matrix $X \in \mathbb{R}^{n \times d}$, we use $E \in \mathbb{R}^{n \times d}$ to denote the positional encoding, we then define:*

$$\mathcal{F}(X) := \mathsf{TF}^L \circ \mathsf{TF}^{L-1} \circ \cdots \circ \mathsf{TF}^2 \circ \mathsf{TF}^1(X + E)$$

G.2 APPROXIMATION VIA DIT

Theorem G.7. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0,1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N \times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d \times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ be defined as Definition F.4.*
- *Let the time-dependent mean of Gaussian distribution $\mu_t(\widetilde{u})$ be defined as Definition F.5.*
- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*
- *Denote $\sigma_{\min} > 0$.*
- *Sample $z \sim \mathcal{N}(0, I_d)$.*
- *Define a model function $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ with parameters θ.*
- *Let the training objective $\mathcal{L}(\theta)$ be defined as Definition F.9.*

Then there exists a transformer network $f_{\mathcal{T}} \in \mathcal{T}_P^{2,1,4}$ defining function $F_\theta(z, c, t) := f_{\mathcal{T}}(R([z^\top, c^\top, t]^\top))$ with parameters θ that satisfies $\mathcal{L}(\theta) \leq \epsilon$ for any error $\epsilon > 0$.

Proof. Following Assumption E.10, we first denote $\widetilde{V}_\tau = \widetilde{\mathcal{M}}_\tau(c)$ for any $\tau \in [\frac{T}{\Delta t}]$ to discretize function $\mathcal{M}$. Then we have:

$$\widetilde{u}_\tau = \mathcal{D}^{-1}\Big((\Phi\widetilde{\mathcal{M}}(c))_\tau\Big). \tag{4}$$

where this step follows from Definition E.3 and Definition E.4.

Besides, we also have:

$$\begin{aligned}
\mu_t(\widetilde{u}) &= H_N g(t) \\
&= \Big(H_{N-1}(I_s - \frac{1}{N}A)^\top + \frac{1}{N}\widetilde{u}_N B^\top\Big)g(t) \\
&= \bigg(H_{N-2}\Big((I_s - \frac{1}{N-1}A)^\top + \frac{1}{N-1}\widetilde{u}_N B^\top\Big)(I_s - \frac{1}{N}A)^\top + \frac{1}{N}\widetilde{u}_N B^\top\bigg)g(t) \\
&= \bigg(H_0 \prod_{\tau=1}^{N}(I_s - \frac{1}{\tau}A)^\top + \sum_{\tau=1}^{N}\Big(\prod_{\tau'=1}^{\tau-1}(I_s - \frac{1}{\tau'}A)^\top\Big) \cdot \frac{1}{N+1-\tau}\widetilde{u}_{N+1-\tau}B^\top\bigg)g(t) \quad (5)
\end{aligned}$$

where these steps follow from Definition F.5 and simple algebras.

Recall $F_\theta(z, c, t) := f_{\mathcal{T}}(R([z^\top, c^\top, t]^\top))$, we choose $n = 1$, then there is a target function given by:

$$
\begin{aligned}
& f_{\mathcal{T}}([z^\top, c^\top, t]) \\
= & \frac{\sigma_t'(\widetilde{u})}{\sigma_t(\widetilde{u})}(z - \Bigg(H_0 \prod_{\tau=1}^{N}(I_s - \frac{1}{\tau}A)^\top + \sum_{\tau=1}^{N}\Big(\prod_{\tau'=1}^{\tau-1}(I_s - \frac{1}{\tau'}A)^\top\Big) \cdot \frac{1}{N+1-\tau}\widetilde{u}_{N+1-\tau}B^\top\Bigg)g(t)) \\
& + \Bigg(H_0 \prod_{\tau=1}^{N}(I_s - \frac{1}{\tau}A)^\top + \sum_{\tau=1}^{N}\Big(\prod_{\tau'=1}^{\tau-1}(I_s - \frac{1}{\tau'}A)^\top\Big) \cdot \frac{1}{N+1-\tau}\widetilde{u}'_{N+1-\tau}B^\top\Bigg)g(t) \\
& + \Bigg(H_0 \prod_{\tau=1}^{N}(I_s - \frac{1}{\tau}A)^\top + \sum_{\tau=1}^{N}\Big(\prod_{\tau'=1}^{\tau-1}(I_s - \frac{1}{\tau'}A)^\top\Big) \cdot \frac{1}{N+1-\tau}\widetilde{u}_{N+1-\tau}B^\top\Bigg)g'(t) \\
= & \frac{\sigma_t'(\widetilde{u})}{\sigma_t(\widetilde{u})}(z \\
& - \Bigg(H_0 \prod_{\tau=1}^{N}(I_s - \frac{1}{\tau}A)^\top + \sum_{\tau=1}^{N}\Big(\prod_{\tau'=1}^{\tau-1}(I_s - \frac{1}{\tau'}A)^\top\Big) \cdot \frac{1}{N+1-\tau}\mathcal{D}^{-1}\Big((\Phi\widetilde{\mathcal{M}}(c))_{N+1-\tau}\Big)B^\top\Bigg)g(t)) \\
& + \Bigg(H_0 \prod_{\tau=1}^{N}(I_s - \frac{1}{\tau}A)^\top + \sum_{\tau=1}^{N}\Big(\prod_{\tau'=1}^{\tau-1}(I_s - \frac{1}{\tau'}A)^\top\Big) \cdot \frac{1}{N+1-\tau}\Big(\mathcal{D}^{-1}\Big((\Phi\widetilde{\mathcal{M}}(c))_{N+1-\tau}\Big)\Big)'B^\top\Bigg)g(t) \\
& + \Bigg(H_0 \prod_{\tau=1}^{N}(I_s - \frac{1}{\tau}A)^\top + \sum_{\tau=1}^{N}\Big(\prod_{\tau'=1}^{\tau-1}(I_s - \frac{1}{\tau'}A)^\top\Big) \cdot \frac{1}{N+1-\tau}\mathcal{D}^{-1}\Big((\Phi\widetilde{\mathcal{M}}(c))_{N+1-\tau}\Big)B^\top\Bigg)g'(t)
\end{aligned}
$$

where the first step follows the combination of Theorem F.10 and Eq. (5), and the differentiablity of $\widetilde{u}_\tau$ is ensure by Assumption E.7, the second step follows from Eq. (4).

Following Theorem 2 and Theorem 3 in Yun et al. (2019), we thus complete the proof by obtaining the theorem result. □

H INTERPOLATION AND EXTRAPOLATION

This section first introduce properties of HiPPO-LegS in Appendix H.1. Also, we bound the error of VLFM in Appendix H.2.

H.1 HIPPO-LEGS PROPERTIES

Lemma H.1 (Proposition 6 in Gu et al. (2020)). *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N \times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d \times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ be defined as Definition F.4.*
- *Let the time-dependent mean of Gaussian distribution $\mu_t(\widetilde{u})$ be defined as Definition F.5.*
- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*

- *Denote $\sigma_{\min} > 0$.*
- *Sample $z \sim \mathcal{N}(0, I_d)$.*
- *Define a model function $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ with parameters θ.*
- *Let the training objective $\mathcal{L}(\theta)$ be defined as Definition F.9.*
- *Let Assumptions E.7, Assumption E.8, Assumption E.10 and Assumption E.9 hold.*

Then we have:

$$\|\mu_{\tau \cdot \Delta t}(\widetilde{u}) - \widetilde{u}_\tau\|_2 = O(t^k s^{-k+1/2})$$

Proof. This lemma is a re-statement of Proposition 6 in Gu et al. (2020). □

Lemma H.2 (Proposition 3 in Gu et al. (2020))**.** *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N \times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d \times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ be defined as Definition F.4.*
- *Let the time-dependent mean of Gaussian distribution $\mu_t(\widetilde{u})$ be defined as Definition F.5.*
- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*
- *Denote $\sigma_{\min} > 0$.*
- *Sample $z \sim \mathcal{N}(0, I_d)$.*
- *Define a model function $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ with parameters θ.*
- *Let the training objective $\mathcal{L}(\theta)$ be defined as Definition F.9.*
- *Let Assumptions E.7, Assumption E.8, Assumption E.10 and Assumption E.9 hold.*

For any integer scale factor $\beta > 0$, the frames of video $\widetilde{V}_\tau$ is scaled to $\widetilde{V}_{\beta\tau}$, it doesn't affect the result of H_N (Definition F.3).

Proof. This lemma is a re-statement of Proposition 3 in Gu et al. (2020). □

H.2 Error Bounds

Lemma H.3. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*

- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N \times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d \times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ and matrix G be defined as Definition F.4.*
- *Denote $1/\lambda^* := \lambda_{\min}(G) > 0$.*
- *Let the time-dependent mean of Gaussian distribution $\mu_t(\widetilde{u})$ be defined as Definition F.5.*
- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*
- *Denote $\sigma_{\min} > 0$.*
- *Sample $z \sim \mathcal{N}(0, I_d)$.*
- *Define a model function $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ with parameters θ.*
- *Let the training objective $\mathcal{L}(\theta)$ be defined as Definition F.9.*
- *Let Assumptions E.7, Assumption E.8, Assumption E.10 and Assumption E.9 hold.*
- *$\delta \in (0, 1)$.*
- *Choosing $s = O(\frac{\Delta t}{T} \log((\frac{\Delta t}{T})^{1.5}/1/\lambda^*))$.*

Particularly, we define:

- *$\epsilon_1 := O(T^k s^{-k+1/2})$.*
- *$\epsilon_2 := O(\sqrt{d \log(d/\delta)})$.*
- *$\epsilon_3 := 1/\lambda^* U d^{0.5} \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s))$.*

Then with a probability at least $1 - \delta$, we have:

$$\|\psi_t(\widetilde{u}) - u_t\|_2 \leq \epsilon_1 + \epsilon_2 + \epsilon_3.$$

Proof. We have:

$$\begin{aligned}
\|\psi_t(\widetilde{u}) - u_t\|_2 &= \|\sigma_t(\widetilde{u}) \cdot z + \mu_t(\widetilde{u}) - u_t\|_2 \\
&\leq \|\sigma_t(\widetilde{u}) \cdot z\|_2 + \|\mu_t(\widetilde{u}) - u_t\|_2 \\
&\leq \|z\|_2 + \|\mu_t(\widetilde{u}) - u_t\|_2 \\
&\leq O(\sqrt{d \log(d/\delta)}) + \|\mu_t(\widetilde{u}) - u_t\|_2 \\
&= \epsilon_2 + \|\mu_t(\widetilde{u}) - u_t\|_2
\end{aligned}$$

where the first step follows from Definition F.8, the second step follows from triangle inequality, the third step follows from $\sigma_t(\widetilde{u}) \leq 1, \forall t \in [0, T]$ by some simple algebras and Definition F.6, the fourth step follows from the union bound of Gaussian tail bound (Fact E.11), the last step follows from the definition of ϵ_2.

Then we get:

$$\begin{aligned}
\|\mu_t(\widetilde{u}) - u_t\|_2 &= \|H_N g(t) - u_t\|_2 \\
&= \|(M \cdot G)^\dagger (M \cdot u) \cdot g(t) - u_t\|_2 \\
&\leq \|(M \cdot G)^\dagger (M \cdot u) \cdot g(t) - G^\dagger u \cdot g(t)\|_2 + O((\frac{T}{\Delta t})^k s^{-k+1/2}) \\
&\leq \|((M \cdot G)^\dagger (M \cdot u) - G^\dagger u\|_2 \cdot \|g(t)\|_2 + O((\frac{T}{\Delta t})^k s^{-k+1/2}) \\
&= \|((M \cdot G)^\dagger (M \cdot u) - G^\dagger u\|_2 \cdot \|g(t)\|_2 + \epsilon_1
\end{aligned}$$

where the first step follows from Definition F.5, the second step follows from optimal error of solving $\|MGH - Mu\|_2^2$, pseudoinverse matrix $(M \cdot G)^\dagger \in \mathbb{R}^{d \times \frac{T}{\Delta t}}$ and defining a mask $M = \mathrm{diag}(m)$ where $m := \{0,1\}^{\frac{T}{\Delta t}}$ and $\langle m, \mathbf{1}_{\frac{T}{\Delta t}} \rangle = N$, the third step follows from the optimal error of solving $\|GH - u\|_2^2$, pesdueo-inverse matrix $G^\dagger \in \mathbb{R}^{d \times \frac{T}{\Delta}}$ and Lemma H.1, the fourth step follows from Cauchy–Schwarz inequality and the last step follows from the definition of ϵ_2.

Next, we can show that:

$$\begin{aligned}
\|(M \cdot G)^\dagger (M \cdot u) - G^\dagger u\|_2 &= \|(M \cdot G)^\dagger (M \cdot u) - G^\dagger (M \cdot u) + G^\dagger (M \cdot u) - G^\dagger u\|_2 \\
&\leq \|(M \cdot G)^\dagger (M \cdot u) - G^\dagger (M \cdot u)\|_2 + \|G^\dagger (M \cdot u) - G^\dagger u\|_2 \\
&\leq \|(M \cdot G)^\dagger - G^\dagger\| \|M \cdot u\|_2 + \|G^\dagger\| \|(M \cdot u) - u\|_2
\end{aligned}$$

where the first step follows from simple algebras, the second step follows from triangle inequality, the last step follows from Cauchy–Schwarz inequality.

We first give:

$$\|G^\dagger\| \leq 1/\lambda^* \sqrt{\frac{T}{\Delta t} \cdot s} \tag{6}$$

where this step follows from Definition F.4, Fact E.12 and the definition of ℓ_2 norm.

And:

$$\|u\|_2 \leq U \sqrt{\frac{T}{\Delta t} \cdot d}$$

where this step follows from Assumption E.9 and the definition of ℓ_2 norm.

Also:

$$\|G\| \leq \sqrt{\frac{T}{\Delta t} \cdot s} \exp(O(\frac{T}{\Delta t} \cdot s)) \tag{7}$$

where this step follows from Definition F.4 and the definition of ℓ_2 norm.

Besides, we have:

$$\begin{aligned}
\|(M \cdot G)^\dagger - G^\dagger\| \leq &\ \max\{\|G^\dagger\|, \|M \cdot G^\dagger\|\} \cdot \|(M \cdot G)^\dagger - G^\dagger\| \\
\leq &\ \frac{1/{\lambda^*}^2 (\frac{T}{\Delta t} s)^{1.5} \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s))}{1 - 1/\lambda^* \frac{T}{\Delta t} s \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s))}
\end{aligned}$$

where the first step follows from Fact E.13, simple algebras, and Cauchy–Schwarz inequality, the second step follows from Eq. (6), Eq. (7), Definition F.4 and simeple algebras.

Combining all results, we get:

$$\begin{aligned}
&\|((M \cdot G)^\dagger (M \cdot u) - G^\dagger u\|_2 \\
\leq &\frac{1/{\lambda^*}^2 (\frac{T}{\Delta t} s)^{1.5} \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s))}{1 - 1/\lambda^* \frac{T}{\Delta t} s \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s))} \cdot U \sqrt{\frac{T}{\Delta t} N d} + 1/\lambda^* \sqrt{\frac{T}{\Delta t} - N} \cdot U \sqrt{\frac{T}{\Delta t} \cdot d} \\
\leq &\, 1/\lambda^* U d^{0.5} \sqrt{\frac{T}{\Delta t} (\frac{T}{\Delta t} - N)} \cdot \Big(\frac{1/\lambda^* (\frac{T}{\Delta t})^{1.5} N^{0.5} s^{1.5} \cdot \exp(O(\frac{T}{\Delta t} s))}{1 - 1/\lambda^* (\frac{T}{\Delta t})^{1.5} s \cdot \exp(O(\frac{T}{\Delta t} s))} + 1 \Big) \\
\leq &\, 1/\lambda^* U d^{0.5} \sqrt{\frac{T}{\Delta t} (\frac{T}{\Delta t} - N)} \cdot \frac{1}{1 - 1/\lambda^* (\frac{T}{\Delta t})^{1.5} s \cdot \exp(O(\frac{T}{\Delta t} s))} \\
\leq &\, O\Big(1/\lambda^* U d^{0.5} \sqrt{\frac{T}{\Delta t} (\frac{T}{\Delta t} - N)} \Big)
\end{aligned}$$

where the second and third steps follow from simple algebras, the last step follows from plugging the choice of s.

Finally, we have:

$$\begin{aligned}
\|((M \cdot G)^\dagger (M \cdot u) - G^\dagger u\|_2 \cdot \|g(t)\|_2 \leq &\, O\Big(1/\lambda^* U d^{0.5} \sqrt{\frac{T}{\Delta t} (\frac{T}{\Delta t} - N)} \Big) \cdot \sqrt{s} \exp(O(\frac{T}{\Delta t} s)) \\
\leq &\, 1/\lambda^* U d^{0.5} \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s)) \\
= &\, \epsilon_3
\end{aligned}$$

these steps follow from simple algebras, Definition F.4 and the definition of ϵ_3. □

Theorem H.4. *If the following conditions hold:*

- *Given a video caption distribution $\mathcal{V}_c$ as Definition E.1.*
- *For any $(V, c) \sim \mathcal{V}_c$, we define the discretized form of video as Definition E.2.*
- *Let the observation matrix $\Phi : \{0, 1\}^{N \times \frac{T}{\Delta t}}$ be defined as Definition E.3.*
- *Let the visual decoder function $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^D$ be defined as Definition E.4.*
- *Let the ideal version of the sequence of latent patches $u \in \mathbb{R}^{\frac{T}{\Delta t} \times d}$ be defined as Definition E.5.*
- *Let the real-world version of the sequence of latent patches $\widetilde{u} \in \mathbb{R}^{N \times d}$ be defined as Definition E.6.*
- *Let $H_N \in \mathbb{R}^{d \times s}$ be defined as Definition F.3.*
- *Let the function of polynomials $g(t)$ and matrix G be defined as Definition F.4.*
- *Denote $1/\lambda^* := \lambda_{\min}(G) > 0$.*
- *Let the time-dependent mean of Gaussian distribution $\mu_t(\widetilde{u})$ be defined as Definition F.5.*
- *Let the time-dependent standard deviation $\sigma_t(\widetilde{u})$ be defined as Definition F.6.*
- *Denote $\sigma_{\min} > 0$.*
- *Sample $z \sim \mathcal{N}(0, I_d)$.*
- *Define a model function $F_\theta : \mathbb{R}^d \times \mathbb{R}^\ell \times [0, T] \to \mathbb{R}^d$ with parameters θ.*
- *Let the training objective $\mathcal{L}(\theta)$ be defined as Definition F.9.*
- *Let Assumptions E.7, Assumption E.8, Assumption E.10 and Assumption E.9 hold.*

- $\delta \in (0, 1)$.

Particularly, we define:

- $\epsilon_1 := O(T^k s^{-k+1/2})$.
- $\epsilon_2 := O(\sqrt{d \log(d/\delta)})$.
- $\epsilon_3 := 1/\lambda^* U d^{0.5} \sqrt{\frac{T}{\Delta t} - N} \cdot \exp(O(\frac{T}{\Delta t} s))$.

Then with a probability at least $1 - \delta$*, we have:*

$$\|\mathcal{D}(z + \int_0^t F_\theta(z, c, t')\mathrm{d}t') - u_t\|_2 \leq \epsilon_0 + L_0(\epsilon_1 + \epsilon_2 + \epsilon_3).$$

Proof. This proof follows from the combination of Assumption E.8, Theorem G.7 and Lemma H.3. □