Time-Correlated Video Bridge Matching

Anonymous Authors[1]

Abstract

Diffusion models excel in noise-to-data generation tasks, providing a mapping from a Gaussian distribution to a more complex data distribution. However they struggle to model translations between complex distributions, limiting their effectiveness in data-to-data tasks. While Bridge Matching models address this by finding the translation between data distributions, their application to time-correlated data sequences remains unexplored. This is a critical limitation for video generation and manipulation tasks, where maintaining temporal coherence is particularly important. To address this gap, we propose Time-Correlated Video Bridge Matching (TCVBM), a framework that extends BM to time-correlated data sequences in the video domain. TCVBM explicitly models inter-sequence dependencies within the diffusion bridge, directly incorporating temporal correlations into the sampling process. We compare our approach to classical methods based on bridge matching and diffusion models for three video-related tasks: frame interpolation, image-to-video generation, and video super-resolution. TCVBM achieves superior performance across multiple quantitative metrics, benchmark datasets and human evaluation. The source code will be publicly available upon publication.

1. Introduction

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b) have emerged as a powerful paradigm for generative modeling, achieving remarkable results in high-fidelity data synthesis (Saharia et al., 2022; Rombach et al., 2022; Arkhipkin et al., 2024; Labs, 2024; Arkhipkin et al., 2025a). By iteratively denoising samples from a Gaussian distribution, these models excel at producing diverse and realistic outputs. However, despite their widespread adoption, diffusion models exhibit a critical limitation: they struggle to model translations between complex-structured data distributions. This shortcoming hinders their effectiveness in data-to-data tasks, where smooth bridging of the distributions is essential.

In contrast, Bridge Matching (BM) offers a principled solution to this problem by explicitly constructing a translation between arbitrary data distributions (Peluchetti, 2023b;a; Liu et al., 2022; Zhou et al., 2023). These methods learn vector field that connects source and target distributions, demonstrating strong performance in image-to-image tasks (Shi et al., 2023; Liu et al., 2023). However, the translation between independent data sequences, the components of which are time-correlated, remains unaddressed. At the same time, this is exactly what video data is, where one data sample is a sequence of correlated frames. While existing bridge matching methods assume to operate with video data samples without considering their internal structure in the prior (Wang et al., 2025b), this omission can lead to a decrease in temporal consistency in the generated video.

Contribution. To address this gap, we propose Time Correlated Video Bridge Matching (TCVBM), a framework that extends Bridge Matching to time-dependent video data. Furthermore, TCVBM utilizes a unique feature of the Bridge Matching framework to incorporate inductive bias via designing a prior process to promote better consistency between frames (Figure 1). We evaluate TCVBM on three video-related tasks: frame interpolation, image-to-video generation, and video super resolution. A comparison with other diffusion approaches and task-specific methods demonstrates that TCVBM provides better temporal consistency, reconstruction quality and achieves higher human preferences.

2. Related Works

2.1. Bridge Models

Despite the success of diffusion models in generative tasks (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b), their reliance on Gaussian noise as a prior lacks meaningful structural information about the data. In contrast, models that match velocity fields using pre-defined

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

transport maps can achieve competitive performance (Lipman et al., 2023). Bridge Matching offers a particularly flexible framework, outperforming standard diffusion for tasks like image restoration, translation, and reconstruction (Delbracio & Milanfar, 2024; Zhou et al., 2024a; Liu et al., 2023). However, applying Bridge Matching to correlated sequential data, such as video, remains largely unexplored. A recent extension to image-to-video generation (Wang et al., 2025b) overlooked inherent temporal dependencies in the prior structure. Our approach addresses this by designing an interpolant that explicitly models the linear correlations between video frames.

2.2. Temporal Modeling for Video Data

Architectural Approach. Advances in video generation often adapt pre-trained image models by adding temporal modules like 3D convolutions or temporal attention layers (Ho et al., 2022; Blattmann et al., 2023; Arkhipkin et al., 2023). This architectural specialization continues with Diffusion Transformer-based video models (Chen et al., 2023; Ma et al., 2025). A key challenge is the computational cost of attention, addressed by techniques such as sparse attention and adaptive masking (Zhang et al., 2025b; Xi et al., 2025; Mikhailov et al., 2026). However, these works doesn't consider modeling temporal dependencies within the generative dynamics and SDE prior.

Prior Modification. Other prior works on video generation have considered incorporating cross-correlation between video frames through a modification of the diffusion prior (Ge et al., 2024; Chang et al., 2025; Liu & Vahdat, 2025). However, until now, the Bridge Matching paradigm, which has proven successful in data-to-data tasks, has not considered incorporating correlation through a modification of the prior for working with video sequences.

Additional related works about specific video manipulation tasks can be found in Appendix A.

3. Background on Bridge Matching

We briefly review the Bridge Matching framework (Peluchetti, 2023b;a; Liu et al., 2022; Shi et al., 2023), which constructs diffusion processes for data translation, given a distribution of clean data $p(\mathbf{x}_0)$ and corrupted data $p(\mathbf{x}_T)$ on $\mathbb{R}^D$. The goal is to model a stochastic process that transitions from $\mathbf{x}_0 \sim p(\mathbf{x}_0)$ to $\mathbf{x}_T \sim p(\mathbf{x}_T \mid \mathbf{x}_0)$, while incorporating a prior dynamics.

Consider a coupling $p(\mathbf{x}_0, \mathbf{x}_T) = p(\mathbf{x}_0)\, p(\mathbf{x}_T \mid \mathbf{x}_0)$, and let the prior process be defined by the stochastic differential equation (SDE):

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)\, dt + g(t)\, d\mathbf{W}_t, \tag{1}$$

where $f(\mathbf{x}_t, t)$ is a drift function, $g(t)$ is a time-dependent noise scale, and $\mathbf{W}_t$ is a standard Wiener process. For a fixed starting point $\mathbf{x}_s$, we denote the marginal of the prior process at time t by $q(\mathbf{x}_t \mid \mathbf{x}_s)$.

Bridge Distribution. Given a pair $(\mathbf{x}_0, \mathbf{x}_{t'})$ from the prior, the posterior distribution of the process at time $t < t'$, denoted as $q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_{t'})$, is referred to as the *bridge distribution*. Using Bayes' rule, it is expressed as:

$$q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_{t'}) = \frac{q(\mathbf{x}_{t'} \mid \mathbf{x}_t, \mathbf{x}_0)\, q(\mathbf{x}_t \mid \mathbf{x}_0)}{q(\mathbf{x}_{t'} \mid \mathbf{x}_0)}.$$

Bridge Matching Dynamics. Bridge Matching aims to construct a stochastic process that interpolates between $\mathbf{x}_T$ and $\mathbf{x}_0$ using a reverse-time SDE:

$$d\mathbf{x}_t = \left\{ f(\mathbf{x}_t, t) - g^2(t)\, v^*(\mathbf{x}_t, t) \right\} dt + g(t)\, d\bar{\mathbf{W}}_t,$$

where $\bar{\mathbf{W}}_t$ is a standard Wiener process under time reversal $t \leftarrow T - t$, and dt denotes a negative infinitesimal timestep.

Learning Objective. The drift function $v^*(\mathbf{x}_t, t)$ is approximated using the following optimization objective:

$$\min_{\phi} \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_T, t} \left[\left\| v_\phi(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0) \right\|^2 \right], \tag{2}$$

where $\mathbf{x}_0 \sim p(\mathbf{x}_0)$, $\mathbf{x}_T \sim p(\mathbf{x}_T \mid \mathbf{x}_0)$, and $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)$ is sampled from the bridge distribution. Time t is sampled uniformly from the interval $[0, T]$.

This formulation provides a principled way to learn drift functions that guide the translation of corrupted data samples from $p(\mathbf{x}_T)$ to clean data samples from $p(\mathbf{x}_0)$ through learned diffusion processes.

4. Method

In this section, we introduce our proposed *Time-Correlated Video Bridge Matching (TCVBM)* method for modeling video data sequences. The core idea is to incorporate temporal correlations directly into the prior diffusion process, enabling better coherence and reconstruction of sequential data (Figure 1). We provide formal derivations and defer all proofs to the supplementary material section B.

4.1. Time-Correlated Prior Process

We consider sequences of length N, represented as

$$\mathbf{X} = (\mathbf{x}^1, \ldots, \mathbf{x}^N), \tag{3}$$

where each $\mathbf{x}^n \in \mathbb{R}^D$ for $n = 1, \ldots, N$. We aim to define a prior diffusion process that imposes an inductive bias toward temporal smoothness across elements.

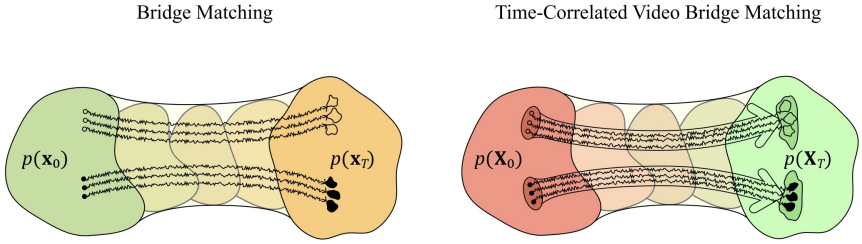


Figure 1. Comparison of the Bridge Matching and Time-Correlated Video Bridge Matching methods. The frames of the same video in the distributions $p(\mathbf{x}_0)$ and $p(\mathbf{X}_0)$ are indicated by dots of the same color. While Bridge Matching method makes the transition from one distribution to another without considering the relationship between frames, our approach treats one video as a single $\mathbf{X}_0$ data sequence and constructs the transition taking into account the internal correlation between video frames.

Column-wise independence across features. To model high-dimensional data efficiently, we assume that the D feature dimensions evolve independently but share the same temporal dynamics. For each feature index $d = 1, \ldots, D$, we define the time-dependent trajectory

$$\mathbf{x}_t^{(d)} = \begin{bmatrix} x_t^{(d,1)} & \ldots & x_t^{(d,N)} \end{bmatrix}^\top \in \mathbb{R}^N,$$

which evolves by a stochastic differential equation (SDE):

$$d\mathbf{x}_t^{(d)} = \left(\mathbf{A}\mathbf{x}_t^{(d)} + \mathbf{b}^{(d)}\right) dt + g(t)\, d\mathbf{W}_t^{(d)},$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a symmetric, invertible matrix encoding temporal correlations, $\mathbf{b}^{(d)} \in \mathbb{R}^N$ is a drift correction term, and $\mathbf{W}_t^{(d)}$ is a standard Wiener process.

Matrix form of the prior. Equivalently, the full intermediate sequence $\mathbf{X}_t \in \mathbb{R}^{N\times D}$ evolves as:

$$d\mathbf{X}_t = (\mathbf{A}\mathbf{X}_t + \mathbf{b})dt + g(t)\, d\mathbf{W}_t,$$

where $\mathbf{b} = [\mathbf{b}^{(1)} \ldots \mathbf{b}^{(D)}] \in \mathbb{R}^{N\times D}$, and $\mathbf{W}_t \in \mathbb{R}^{N\times D}$ is a matrix of independent Wiener processes across columns. In all expressions involving covariance and scores, formulas are applied column-wise. For example,

$$\mathbf{\Sigma}_t^{-1}(\mathbf{X}_t - \boldsymbol{\mu}_t) \in \mathbb{R}^{N\times D}$$

denotes applying $\mathbf{\Sigma}_t^{-1} \in \mathbb{R}^{N\times N}$ independently to each column of $\mathbf{X}_t - \boldsymbol{\mu}_t$. Further, unless otherwise stated, we will assume a time-independent noise scale $g(t) = \sqrt{\epsilon}$ for simplicity.

We now derive the transition and bridge distributions for this prior, which are essential for bridge matching.

Proposition 1 (Correlated Process Score)**.** *Let $\mathbf{X}_t$ follow the linear SDE:*

$$d\mathbf{X}_t = (\mathbf{A}\mathbf{X}_t + \mathbf{b})\, dt + \sqrt{\epsilon}\, d\mathbf{W}_t, \quad \mathbf{X}_0 \sim \delta_{\mathbf{X}_0}, \tag{4}$$

then the marginal distribution of $\mathbf{X}_t$ is Gaussian:

$$q(\mathbf{X}_t|\mathbf{X}_0) = \mathcal{N}(\mathbf{X}_t \mid \boldsymbol{\mu}_{t|0}(\mathbf{X}_0), \mathbf{\Sigma}_{t|0}), \tag{5}$$

with

$$\boldsymbol{\mu}_{t|0}(\mathbf{X}_0) = e^{\mathbf{A}t}\mathbf{X}_0 + \left(e^{\mathbf{A}t} - I\right)\mathbf{A}^{-1}\mathbf{b}, \tag{6}$$

$$\mathbf{\Sigma}_{t|0} = \epsilon\, \frac{e^{2\mathbf{A}t} - I}{2}\mathbf{A}^{-1}. \tag{7}$$

The score function is then given by

$$\nabla_{\mathbf{X}_t} \log q(\mathbf{X}_t|\mathbf{X}_0) = -\mathbf{\Sigma}_{t|0}^{-1}(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)). \tag{8}$$

To perform bridge matching, one also needs to be able to sample from $q(\mathbf{X}_t|\mathbf{X}_0, \mathbf{X}_T)$.

Proposition 2 (Correlated Bridge Distribution)**.** *Let $\mathbf{X}_t$ follow the same SDE as in Proposition 1. Then, given fixed endpoints $\mathbf{X}_0$ and $\mathbf{X}_{t'}$, the posterior (bridge) distribution of $\mathbf{X}_t$ is Gaussian:*

$$q(\mathbf{X}_t|\mathbf{X}_0, \mathbf{X}_{t'}) = \mathcal{N}(\mathbf{X}_t \mid \boldsymbol{\mu}_{t|0,t'}, \mathbf{\Sigma}_{t|0,t'}), \tag{9}$$

where

$$\boldsymbol{\mu}_{t|0,t'} = \boldsymbol{\mu}_{t|0}(\mathbf{X}_0) + \mathbf{\Sigma}_{t|0}\mathbf{\Sigma}_{t'|0}^{-1}(\mathbf{X}_{t'} - \boldsymbol{\mu}_{t'|0}(\mathbf{X}_0)), \tag{10}$$

$$\mathbf{\Sigma}_{t|0,t'} = \mathbf{\Sigma}_{t|0} - \mathbf{\Sigma}_{t|0}\mathbf{\Sigma}_{t'|0}^{-1}\mathbf{\Sigma}_{t|0}. \tag{11}$$

Together, Propositions 1 and 2 provide closed-form expressions required to implement bridge matching under the time-correlated prior.

4.2. Time-Correlated Video Bridge Matching

Training. To train a bridge matching model, we follow the general framework of Bridge Matching described in section 3. We assume access to clean samples $\mathbf{X}_0 \sim p_0(\mathbf{X}_0)$, and a degradation process $p(\mathbf{X}_T|\mathbf{X}_0)$, together forming a coupling $p(\mathbf{X}_0, \mathbf{X}_T) = p_0(\mathbf{X}_0)p(\mathbf{X}_T|\mathbf{X}_0)$.

Algorithm 1 Training

Require: data from coupling $p_0(\mathbf{X}_0)p_T(\mathbf{X}_T|\mathbf{X}_0)$ and coefficients $\mathbf{A}$, $\mathbf{b}$ and ϵ for prior (1).

repeat
  $t \sim \mathcal{U}([0,1]), \mathbf{X}_0 \sim p_0(\mathbf{X}_0), \mathbf{X}_T \sim p(\mathbf{X}_T \mid \mathbf{X}_0)$
  $\mathbf{X}_t \sim q\big(\mathbf{X}_t \mid \mathbf{X}_0, \mathbf{X}_T\big)$ (9)
  Take gradient descent step on $\mathbf{X}_0^\phi(\mathbf{X}_t, t)$ (13)
until convergence

We aim to minimize the squared error between the predicted score function $v_\phi(\mathbf{X}_t, t)$ and the score of prior process $\nabla_{\mathbf{X}_t} \log p(\mathbf{X}_t|\mathbf{X}_0)$, averaged over bridge samples $\mathbf{X}_t \sim p(\mathbf{X}_t|\mathbf{X}_0, \mathbf{X}_T)$:

$$\min_\phi \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\left\|v_\phi(\mathbf{X}_t,t) + \boldsymbol{\Sigma}_{t|0}^{-1}(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0))\right\|^2\right], \tag{12}$$

where $t \sim \text{Uniform}(0, T)$.

This objective can be simplified by reparameterizing the score function in terms of an intermediate predictor:

Proposition 3 (Reparameterization of the drift function)**.** *The minimizer $v^*(\mathbf{X}_t, t)$ of the objective (12) can be expressed as:*

$$v^*(\mathbf{X}_t, t) = -\boldsymbol{\Sigma}_{t|0}^{-1}\left(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^*(\mathbf{X}_t, t))\right),$$

where $\widehat{\mathbf{X}}_0^*(\mathbf{X}_t, t)$ is the solution to the regression problem:

$$\min_\phi \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\|\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t,t) - \mathbf{X}_0\|^2\right]. \tag{13}$$

Thus, learning the score function reduces to learning a predictor for the clean data $\mathbf{X}_0$.

We parameterize the predictor $\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)$ with a neural network and train it using the regression loss in (13). The training procedure is summarized in Algorithm 1.

Inference. At inference time, given a corrupted sequence $\mathbf{X}_T \sim p(\mathbf{X}_T)$, we perform iterative denoising using the learned predictor and the time-correlated bridge distribution. Given a schedule $0 = t_0 < t_1 < \cdots < t_N = T$, we iteratively refine the estimate of $\mathbf{X}_0$ by sampling from posterior:

$$\mathbf{X}_{t_{n-1}} \sim p(\mathbf{X}_{t_{n-1}} \mid \widehat{\mathbf{X}}_0, \mathbf{X}_{t_n}),$$

where $\widehat{\mathbf{X}}_0 = \widehat{\mathbf{X}}_0^\phi(\mathbf{X}_{t_n}, t_n)$ obtained by using prediction of the trained model. This process is detailed in Algorithm 2.

4.3. Choice of the Prior Process for Video Manipulation Tasks

To encourage smooth transitions between consecutive elements of the sequence, we define the prior matrix $\mathbf{A}$ of the size $N \times N$ with a tridiagonal structure:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{bmatrix}.$$

Here, $\mathbf{A}$ promotes temporal correlations across adjacent elements, and the per-element prior is:

$$d\mathbf{x}_t^n = \left((\mathbf{x}_t^{n-1} - \mathbf{x}_t^n) + (\mathbf{x}_t^{n+1} - \mathbf{x}_t^n)\right) dt + \sqrt{\epsilon}\, d\mathbf{W}_t, \tag{14}$$

where $n = 2, \ldots, N-1$. This formulation naturally encourages a linear relationship between the elements of the sequence. From the point of view of this approximation, each frame $\mathbf{x}_t^n$ should remain close to the average of its neighbors $\mathbf{x}_t^{n-1}$ and $\mathbf{x}_t^{n+1}$. It is particularly well-suited for video-related tasks, where the frames of one video are correlated with each other, and the prior process enforces consistency and smoothness between them.

Depending on the video manipulation task, we suggest the following options for vector $\mathbf{b}$ and equation 14:

Frame Interpolation. In this case, we consider the sequence of length $N+2$, represented as

$$\widetilde{\mathbf{X}} = (\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^N, \mathbf{x}^{N+1}),$$

where the endpoints $\mathbf{x}^0$ and $\mathbf{x}^{N+1}$ are fixed as the initial and final frames of a video clip, between which it is necessary to make interpolation. The middle part of the video will be defined in the same way as in the equation 3:

$$\mathbf{X} = (\mathbf{x}^1, \ldots, \mathbf{x}^N),$$

and all statements for $\mathbf{X}_t$ from paragraphs 4.1 and 4.2 remain valid. Considering in equation 14 for all t and $n = 1, \ldots, N$ $\mathbf{x}_t^0 = \mathbf{x}^0$ and $\mathbf{x}_t^{N+1} = \mathbf{x}^{N+1}$, we can define the vector $\mathbf{b}$ as:

$$\mathbf{b} = \left[\mathbf{x}^0, 0, \ldots 0, \mathbf{x}^{N+1}\right]^T, \quad \mathbf{b} \in \mathbb{R}^{N \times D},$$

i.e. $\mathbf{b}$ enforces the boundary conditions from the known fixed endpoints $\mathbf{x}^0$ and $\mathbf{x}^{N+1}$.

Algorithm 2 Inference

Require: Input $\mathbf{X}_T \sim p_T(\mathbf{X}_T)$, trained model $\widehat{\mathbf{X}}_0^\phi(\cdot,\cdot)$, time schedule $\{t_n\}_{n=0}^N$

Set $\mathbf{X}_{t_N} \leftarrow \mathbf{X}_T$
for $n = N$ **to** 1 **do**
  Predict $\widehat{\mathbf{X}}_0 \leftarrow \widehat{\mathbf{X}}_0^\phi(\mathbf{X}_{t_n}, t_n)$
  Sample $\mathbf{X}_{t_{n-1}} \sim p(\mathbf{X}_{t_{n-1}} \mid \widehat{\mathbf{X}}_0, \mathbf{X}_{t_n})$ using (9)
end for
return $\mathbf{X}_0 = \mathbf{X}_{t_0}$

Image-to-Video Generation. This task corresponds to the case of one fixed left point of the video sequence, i.e.:

$$\widetilde{\mathbf{X}} = (\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^N), \quad \widetilde{\mathbf{X}} \in \mathbb{R}^{(N+1)\times D}$$

$$\mathbf{b} = \left[\mathbf{x}^0, 0, \dots 0\right]^T, \quad \mathbf{b} \in \mathbb{R}^{N\times D},$$

and in the equation 14 for all n and t $\mathbf{x}_t^0 = \mathbf{x}^0$ and $\mathbf{x}_t^{N+1} = 0$.

Video Super Resolution. This is the simplest case, where $\widetilde{\mathbf{X}} = \mathbf{X} \in \mathbb{R}^{N\times D}$, i.e. the endpoints are not fixed, and the vector $\mathbf{b}$ is equal to zero.

5. Experiments

5.1. Video Manipulation Task

To test the extension of our framework, we choose three common video manipulation tasks, following the Section 4.3:

Frame Interpolation. A sequence of N frames is fed into the network as input, with the first and last frames from the dataset fixed. At the output, $N-2$ middle frames are generated, which are then compared with $N-2$ corresponding video frames from the dataset.

Image-to-Video Generation. The task is to generate the remaining $N-1$ frames based on the first frame.

Video Super Resolution. The network accepts N low resolution frames from the dataset, and N frames with the dataset resolution are expected at the output.

5.2. Proof-of-concept Experiments on Moving MNIST Dataset

First, we compare TCVBM with three classical generative methods, namely DDPM (Ho et al., 2020), DDIM (Song et al., 2021a), and Bridge Matching with Brownian Bridge (BM) (Ibe, 2013, Chapter 9), on the Moving MNIST dataset (Srivastava et al., 2015), which contains $10,000$ video sequences each 20 frames long and showing 2 moving digits in a 64×64 resolution. For each method, we use the same setup, except for the loss function and sampling procedure in the forward and reverse processes, which are determined by the specific paradigm of each algorithm. In these experiments, we do not use a composition of our approach with methods involving temporal convolutions or temporal attention to test only the change in diffusion prior.

5.2.1. Implementation Details

For proof-of-concept experimental setup we take our own small and simple U-Net model based on several residual blocks with $2D$ convolutions. The model has approximately 8.7 million parameters. For each experiment, we extract sub-sequences from the videos, consisting of $N = 10$ consecutive frames, and concatenate them into a 10-channel input for the neural network. We use $9,500$ training sequences and 500 validation sequences, with a random split for each seed value. We train each model for $150,000$ iterations with a batch size of 128 and an ema rate of 0.999. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with betas set to 0.9 and 0.95, a weight decay of 10^{-4}, and a learning rate of 3×10^{-5}. All experiments are conducted using a single NVIDIA A100 GPU.

Frame Interpolation. The DDPM and DDIM methods aim to solve the generation problem for middle frames from random noise in the reverse process. For BM and TCVBM methods, we consider two ways to initialize the middle input frames: linear interpolation between the two fixed boundary frames, or noise samples from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The second approach produced better results for BM and TCVBM, so the results presented here further are for noise-based initialization method. For more details, please see the Appendix D.1.

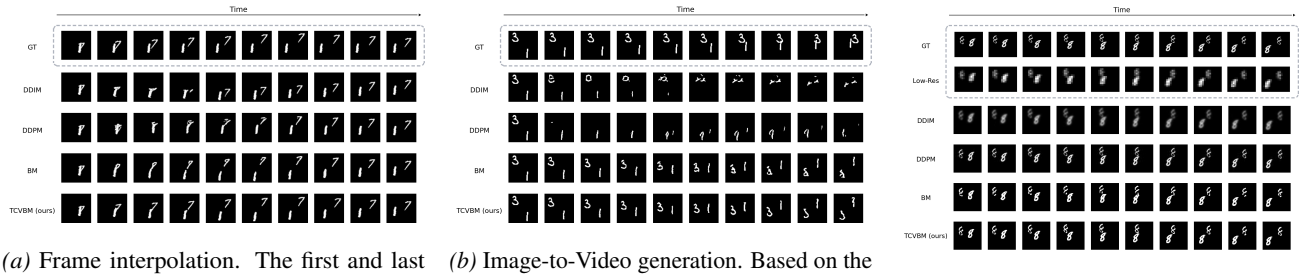
Image-to-Video Generation. We provide the first frame as input data, along with 9 Gaussian samples from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, or 9 copies of the first frame. The second option demonstrated the best results for all methods, so further we consider this initialization variant. For more information, please see the Appendix D.2.

Video Super Resolution. We consider starting with resolutions of 16×16 or 32×32 for all compared algorithms, but due to the simplicity of the task, we did not see any significant advantages for our method using the second option. Therefore, the further results are reported for the starting resolution of 16×16. We also consider two initialization options: feeding the network with only low resolution data or low resolution data concatenated with Gaussian noise. Both approaches produced similar results, and further we use noise-free variant. More details can be found in the Appendix D.3.

5.2.2. Drift Coefficient, Efficiency and Non-constant Correlation

Below, we present the results for the BM and TCVBM methods with ϵ equal to 0.1 and a coefficient at $\mathbf{A}$ set to 1 for the TCVBM model, i.e., $\widetilde{\mathbf{A}} = \alpha\mathbf{A}$ and $\widetilde{\mathbf{b}} = \alpha\mathbf{b}$, where $\alpha = 1$. Additionally, we conduct a series of experiments to find optimal hyperparameters ϵ and α. The results of these experiments can be found in Appendix E.

We present an analysis of the computational complexity of our method in the Appendix F.



(a) Frame interpolation. The first and last frames are fixed and come from the dataset. The task is to generate the intermediate 8 frames. TCVBM provides a smoother and more consistent transition from frame to frame and a better detalization.

(b) Image-to-Video generation. Based on the first frame, the method generate the next 9 frames. TCVBM makes it possible to achieve better appearance of the generations and a lower quality degradation with increasing sequence length.

(c) Video super resolution. The resolution is increased from 16×16 to 64×64. As can be seen, TCVBM outperforms other methods.

Figure 2. Generation results on the MovingMNIST dataset.

Table 1. Results on the MovingMNIST dataset.

Method	Frame interpolation				Image-to-video generation				Video super resolution			
	FVD↓	LPIPS↓	PSNR↑	SSIM↑	FVD↓	LPIPS↓	PSNR↑	SSIM↑	FVD↓	LPIPS↓	PSNR↑	SSIM↑
DDIM	33.61	0.105	15.80	0.760	77.72	0.294	**10.88**	**0.603**	334.70	0.514	17.31	0.613
DDPM	32.40	0.117	14.94	0.741	75.86	0.311	10.72	0.595	607.88	0.236	20.19	0.582
BM	34.32	0.079	17.10	0.794	49.32	0.271	10.63	0.579	**53.52**	**0.020**	22.64	0.954
TCVBM	**30.54**	**0.077**	**17.28**	**0.813**	**44.96**	**0.258**	10.75	0.591	59.49	**0.020**	**22.67**	**0.970**

Furthermore, we explored the dynamic correlation approach in which the coefficient α_t depends on time and increases as the reverse process progresses, i.e. when $t \to 0$. However, our experiments showed that this approach does not provide an advantage over TCVBM when the α is time-independent. The theoretical background and experimental findings are presented in Appendix G.

5.3. Large Scale Experiments

Second, we compare our method with other approaches, including the Bridge Matching-based method and specialized models for larger standard benchmark datasets with complex realistic motion.

Frame Interpolation. Here we use two datasets: the UCF-101 action dataset (Soomro et al., 2012), which contains 9,537 training and 3,783 test video clips at a resolution of 256×256, and the Vimeo-90k septuplet dataset (Xue et al., 2019), which consists of 91,701 7-frame sequences with a fixed resolution of 448×256. We train models in two variants: only on the UCF-101 training dataset ($N = 10$) and on a combined set of septuplet from UCF-101 and the Vimeo90k training dataset ($N = 5$). For frame interpolation, we fine-tune the pre-trained CogVideoX-2B model (Yang et al., 2025) over 10 thousand iterations with batch size 4 and gradient accumulation 4 on one NVIDIA A100 GPU. We use the same AdamW optimizer configuration like in experiments on MovingMNIST dataset.

Image-to-Video Generation. Here we train and test our method on the train-test split of the UCF-101 dataset. In particular, we compare our method with FrameBridge (Wang et al., 2025b). For both methods, we use the architecture of the Latte-S/2 model (Ma et al., 2025). We train TCVBM and FrameBridge from scratch using Xavier initialization over 400 thousand iterations with a batch size 20.

6. Results

Qualitative Comparison. As can be seen in Figure 2 and Figure 3, TCVBM produces more consistent and accurate results compared to other generative methods. By considering the mutual correlation between frames, our approach ensures a smoother transition and information transfer between them, which enhances the overall method stability.

Quantitative Evaluation. First, we perform quantitative evaluation for proof-of-concept experiments on 500 videos from our MovingMNIST validation set using four metrics: FVD (Unterthiner et al., 2019), LPIPS (Zhang et al., 2018), PSNR, and SSIM. Table 1 presents the results. As can be seen, in most cases TCVBM outperforms other approaches in most cases.

It is worth noting that, in terms of metrics, TCVBM and BM demonstrate similar results for the video super resolution task. We explain this by noting that low-resolution video already contains significant inter-frame correlation, which provides a strong foundation for BM's performance, as for

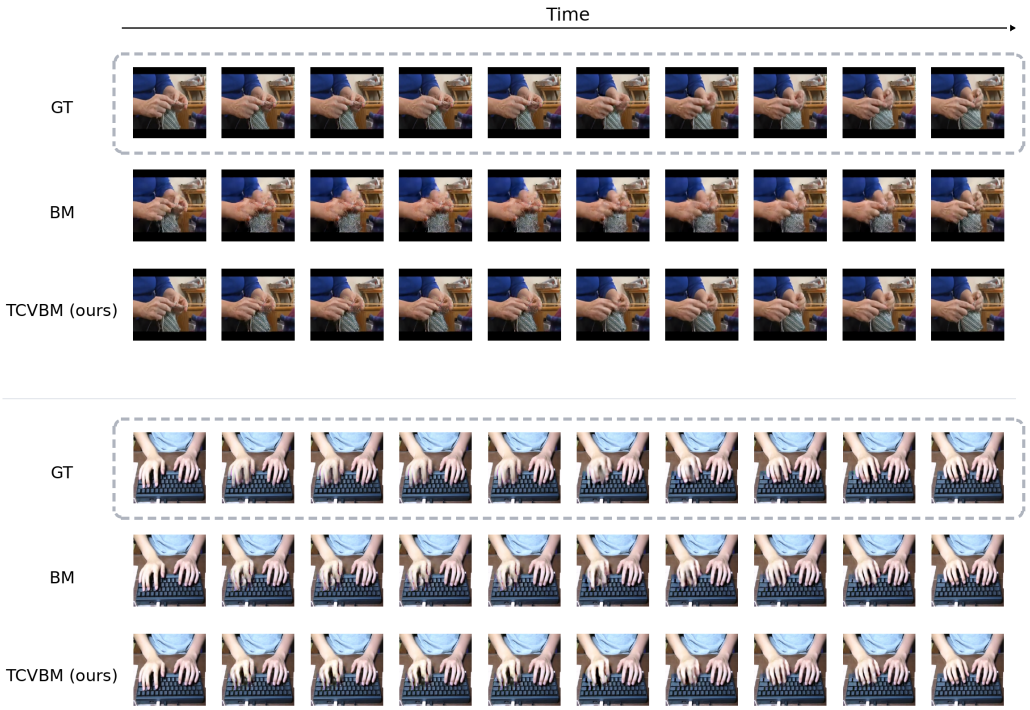


Figure 3. Frame interpolation results on the UCF-101 test dataset, trained on UCF-101 train set. TCVBM retains more structural information by linking to boundary frames, providing improved detail and dynamics.

Table 2. Frame interpolation results on the UCF-101 test set for the model trained on the UCF-101 train set.

Method	FVD↓	LPIPS↓	PSNR↑	SSIM↑
BM	688.26	**0.0583**	34.51	0.8753
TCVBM	**611.48**	0.0588	**35.72**	**0.9074**

successful data-to-data translation method. Even though our method does not demonstrate a clear advantage in this specific task, the results confirm that inter-frame correlation plays a key role in video modeling.

Additional results providing the standard deviation over launches with different random seeds are presented in the Appendix C.

Tables 2, 3 and 4 present the results of metric evaluations on the large scale datasets for frame interpolation and image-to-video generation. For interpolation, we employ the metrics outlined earlier; for image-to-video generation, we compare with other methods using FVD, Inception Score (Saito et al., 2017), and PIC-score (Xing et al., 2023). The results demonstrate that our approach successfully scales to videos with more complex dynamics, offering a competitive advantage across a range of generative tasks.

Human Evaluation. Evaluating generative video is a complex task. Automatic metrics are not always indicative of true quality. We conducted a side-by-side human evaluation study with five assessors. For frame interpolation and image-to-video generation tasks, we compared our method with the Bridge Matching-based approach. We used 40 generated videos for each of the two tasks using the frames from the UCF-101 test dataset. Each of the 40 pairs was evaluated in terms of visual quality and temporal coherence of the transition between frames. The assessor could choose one of the two videos or give preference to both. The results for models trained on the UCF-101 dataset are presented in Table 5. Human evaluation results on the Moving MNIST dataset, including three types of generative tasks, can be found in the Appendix C.2.

7. Discussion

Potential Impact. The proposed Time-Correlated Video Bridge Matching (TCVBM) framework is designed for generative modeling and manipulation with sequential video data. Unlike traditional diffusion and bridge matching methods, which often ignore the intrinsic temporal structure of data, TCVBM explicitly models inter-sequence correlations within the diffusion prior. This principle are applicable not only to video but also to other types of sequences, such as

Table 3. Frame interpolation results for the model trained on the train sets of UCF-101 Vimeo-90K.

Method	UCF-101				Vimeo-90K		
	PSNR↑	SSIM↑	LPIPS↓	FVD↓	PSNR↑	SSIM↑	LPIPS↓
RIFE (Huang et al., 2022)	25.06	0.7489	0.044	552	25.83	0.804	0.098
LDMVFI (Danier et al., 2024)	23.47	0.7509	**0.036**	**369**	25.26	0.809	**0.094**
BM	20.59	0.7134	0.052	678	21.51	0.729	0.113
TCVBM	**25.63**	**0.7513**	0.041	441	**25.84**	**0.811**	0.097

Table 4. Image-to-Video generation results on the UCF-101 dataset. Here we use the same metrics as in (Wang et al., 2025b) and re-train FrameBridge for a fair comparison.

Methos	FVD↓	IS↑	PIC↑
FrameBridge (BM) (Wang et al., 2025b)	603	36	0.6857
I2VGen-XL (Zhang et al., 2023)	571	–	0.5313
SparseCtrl (Guo et al., 2023a)	722	19	0.4818
ExtDM (Zhang et al., 2024)	649	21	–
TCVBM	**467**	**59**	**0.7441**

Table 5. Human evaluation results on the 40 videos from UCF-101 test dataset.

Method	Frame interpolation		Image-to-video generation	
	FrameQuality	TemporalConsistency	FrameQuality	TemporalConsistency
BM	18%	13%	35%	29%
TCVBM	43%	51%	38%	37%
Both	39%	36%	27%	34%

audio signals or time series, where temporal consistency is important. The flexibility in defining prior process parameters, such as tridiagonal matrices for local correlations, allows for the method to be adapted to specific applications.

Limitations. This work has several limitations. First, the current form of the diffusion prior produces smooth videos in which the main content is largely preserved throughout the video's length. This could be useful for local interpolation, video enhancement, and the generation of specific synthetic data with preserved quasi-periodic structure. Extending this method to more complex scenarios involving scene changes, rapid appearance or disappearance of objects, and autoregressive long videos generation is the subject of future research.

Secondly, the reliance on a predefined tridiagonal matrix prior is likely insufficient for capturing the complex nature of temporal dependencies in complex real-world data, particularly long-range and non-linear dynamics. Consequently, the exploration of more sophisticated interpolants – potentially modeling correlations in a feature space via trainable adapters rather than directly between frames-is a critical direction for future work.

Impact Statement

This paper presents work that contributes to the development of video generation methods. This work is a continuation of many studies on this topic and does not pose any specific risks.

References

Arkhipkin, V., Shaheen, Z., Vasilev, V., Dakhova, E., Kuznetsov, A., and Dimitrov, D. Fusionframes: Efficient architectural aspects for text-to-video generation pipeline, 2023. URL https://arxiv.org/abs/2311.13073.

Arkhipkin, V., Viacheslav, V., Andrei, F., Igor, P., Julia, A., Nikolai, G., Anna, A., Evelina, M., Bukashkin, A., Konstantin, K., Andrey, K., and Denis, D. Kandinsky 3: Text-to-image synthesis for multifunctional generative framework. In Hernandez Farias, D. I., Hope, T., and Li, M. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 475–485, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-demo.48. URL https://aclanthology.org/2024.emnlp-demo.48/.

Arkhipkin, V., Korviakov, V., Gerasimenko, N., Parkhomenko, D., Vasilev, V., Letunovskiy, A., Vaulin, N., Kovaleva, M., Kirillov, I., Novitskiy, L., Koposov, D., Kiselev, N., Varlamov, A., Mikhailov, D., Polovnikov, V., Shutkin, A., Agafonova, J., Vasiliev, I., Kargapoltseva, A., Dmitrienko, A., Maltseva, A., Averchenkova, A., Kim, O., Nikulina, T., and Dimitrov, D. Kandinsky 5.0: A family of foundation models for image and video generation, 2025a. URL https://arxiv.org/abs/2511.14993.

Arkhipkin, V., Shaheen, Z., Vasilev, V., Dakhova, E., Sobolev, K., Kuznetsov, A., and Dimitrov, D. Improvey-ourvideos: Architectural improvements for text-to-video generation pipeline. *IEEE Access*, 13:1986–2003, 2025b. doi: 10.1109/ACCESS.2024.3522510.

Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Chang, P., Tang, J., Gross, M., and Azevedo, V. C. How i warped your noise: a temporally-correlated noise prior for diffusion models, 2025. URL https://arxiv.org/abs/2504.03072.

Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., and Li, Z. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.

Chen, J., Feng, B. Y., Cai, H., Wang, T., Burner, L., Yuan, D., Fermuller, C., Metzler, C. A., and Aloimonos, Y. Repurposing pre-trained video diffusion models for event-based video interpolation, 2025. URL https://arxiv.org/abs/2412.07761.

Danier, D., Zhang, F., and Bull, D. Ldmvfi: video frame interpolation with latent diffusion models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i2.27912. URL https://doi.org/10.1609/aaai.v38i2.27912.

Delbracio, M. and Milanfar, P. Inversion by direct iteration: An alternative to denoising diffusion for image restoration, 2024. URL https://arxiv.org/abs/2303.11435.

Ge, S., Nah, S., Liu, G., Poon, T., Tao, A., Catanzaro, B., Jacobs, D., Huang, J.-B., Liu, M.-Y., and Balaji, Y. Preserve your own correlation: A noise prior for video diffusion models, 2024. URL https://arxiv.org/abs/2305.10474.

Guo, X., Zheng, M., Hou, L., Gao, Y., Deng, Y., Wan, P., Zhang, D., Liu, Y., Hu, W., Zha, Z., Huang, H., and Ma, C. I2v-adapter: A general image-to-video adapter for diffusion models, 2024. URL https://arxiv.org/abs/2312.16693.

Guo, Y., Yang, C., Rao, A., Agrawala, M., Lin, D., and Dai, B. Sparsectrl: Adding sparse controls to text-to-video diffusion models, 2023a. URL https://arxiv.org/abs/2311.16933.

Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., and Dai, B. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning, 2023b.

Gushchin, N., Li, D., Selikhanovych, D., Burnaev, E., Baranchuk, D., and Korotin, A. Inverse bridge matching distillation, 2025. URL https://arxiv.org/abs/2502.01362.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., and Salimans, T. Imagen video: High definition video generation with diffusion models, 2022. URL https://arxiv.org/abs/2210.02303.

Huang, Z., Zhang, T., Heng, W., Shi, B., and Zhou, S. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Ibe, O. *Markov processes for stochastic modeling*. Newnes, 2013.

Jain, S., Watson, D., Tabellion, E., Hołyński, A., Poole, B., and Kontkanen, J. Video interpolation with diffusion models, 2024. URL https://arxiv.org/abs/2404.01203.

Kalluri, T., Pathak, D., Chandraker, M., and Tran, D. Flavr: Flow-agnostic video representations for fast frame interpolation. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2070–2081, 2023. doi: 10.1109/WACV56688.2023.00211.

Labs, B. F. Flux. https://github.com/black-forest-labs/flux, 2024.

Lee, H., Kim, T., Chung, T.-y., Pak, D., Ban, Y., and Lee, S. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Liu, C. and Vahdat, A. On equivariance and fast sampling in video diffusion models trained with warped noise, 2025. URL https://arxiv.org/abs/2504.09789.

Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. I2sb: image-to-image schrödinger bridge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Liu, X., Wu, L., Ye, M., and qiang liu. Let us build bridges: Understanding and extending diffusion generative models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. URL https://openreview.net/forum?id=0ef0CRKC9uZ.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ma, X., Wang, Y., Chen, X., Jia, G., Liu, Z., Li, Y.-F., Chen, C., and Qiao, Y. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025.

Mikhailov, D., Letunovskiy, A., Kovaleva, M., Arkhipkin, V., Korviakov, V., Polovnikov, V., Vasilev, V., Sidorova, E., and Dimitrov, D. Nabla: Neighborhood adaptive block-level attention for efficient video generation. *IEEE Access*, 14:64655–64665, 2026. doi: 10.1109/ACCESS.2026.3686867.

Niklaus, S. and Liu, F. Softmax splatting for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Park, J., Lee, C., and Kim, C.-S. Asymmetric bilateral motion estimation for video frame interpolation. In *International Conference on Computer Vision*, 2021.

Peluchetti, S. Diffusion bridge mixture transports, schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023a.

Peluchetti, S. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023b.

Petersen, K. B., Pedersen, M. S., et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Reda, F., Kontkanen, J., Tabellion, E., Sun, D., Pantofaru, C., and Curless, B. Film: Frame interpolation for large motion. In *European Conference on Computer Vision (ECCV)*, 2022.

Ren, W., Yang, H., Zhang, G., Wei, C., Du, X., Huang, S., and Chen, W. Consisti2v: Enhancing visual consistency for image-to-video generation. *arXiv preprint arXiv:2402.04324*, 2024.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding, 2022. URL https://arxiv.org/abs/2205.11487.

Saito, M., Matsumoto, E., and Saito, S. Temporal generative adversarial nets with singular value clipping, 2017. URL https://arxiv.org/abs/1611.06624.

Shen, L., Liu, T., Sun, H., Ye, X., Li, B., Zhang, J., and Cao, Z. Dreammover: Leveraging the prior of diffusion models for image interpolation with large motion. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XV*, pp. 336–353, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72632-3. doi: 10.1007/978-3-031-72633-0_19. URL https://doi.org/10.1007/978-3-031-72633-0_19.

Shi, X., Huang, Z., Wang, F.-Y., Bian, W., Li, D., Zhang, Y., Zhang, M., Cheung, K. C., See, S., Qin, H., et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. *SIGGRAPH 2024*, 2024.

Shi, Y., Bortoli, V. D., Campbell, A., and Doucet, A. Diffusion schrödinger bridge matching. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qy07OHsJT5.

Shi, Z., Xu, X., Liu, X., Chen, J., and Yang, M.-H. Video frame interpolation transformer. In *CVPR*, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. URL https://arxiv.org/abs/1212.0402.

Srivastava, N., Mansimov, E., and Salakhutdinov, R. Unsupervised learning of video representations using lstms. *CoRR*, abs/1502.04681, 2015. URL http://arxiv.org/abs/1502.04681.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. Towards accurate generative models of video: A new metric & challenges, 2019. URL https://arxiv.org/abs/1812.01717.

Voleti, V., Jolicoeur-Martineau, A., and Pal, C. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. In *(NeurIPS) Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.09853.

Wang, X., Zhou, B., Curless, B., Kemelmacher-Shlizerman, I., Holynski, A., and Seitz, S. Generative inbetweening: Adapting image-to-video models for keyframe interpolation. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=ykD8a9gJvy.

Wang, Y., Chen, Z., Xiaoyu, C., Wei, Y., Zhu, J., and Chen, J. Framebridge: Improving image-to-video generation with bridge models. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=iYmV2xRSNW.

Wu, T., Si, C., Jiang, Y., Huang, Z., and Liu, Z. Freeinit: Bridging initialization gap in video diffusion models. *arXiv preprint arXiv:2312.07537*, 2023.

Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., Chen, J., Stoica, I., Keutzer, K., and Han, S. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity, 2025. URL https://arxiv.org/abs/2502.01776.

Xing, J., Xia, M., Zhang, Y., Chen, H., Wang, X., Wong, T.-T., and Shan, Y. Dynamicrafter: Animating open-domain images with video diffusion priors, 2023.

Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, February 2019. ISSN 1573-1405. doi: 10.1007/s11263-018-01144-2. URL http://dx.doi.org/10.1007/s11263-018-01144-2.

Yang, X., He, C., Ma, J., and Zhang, L. Motion-guided latent diffusion for temporally consistent real-world video super-resolution, 2024. URL https://arxiv.org/abs/2312.00853.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., Yin, D., Zhang, Y., Wang, W., Cheng, Y., Xu, B., Gu, X., Dong, Y., and Tang, J. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025. URL https://arxiv.org/abs/2408.06072.

Zhang, G., Zhu, Y., Cui, Y., Zhao, X., Ma, K., and Wang, L. Motion-aware generative frame interpolation, 2025a. URL https://arxiv.org/abs/2501.03699.

Zhang, P., Chen, Y., Su, R., Ding, H., Stoica, I., Liu, Z., and Zhang, H. Fast video generation with sliding tile attention. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=U74MOXPEJd.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhang, S., Wang, J., Zhang, Y., Zhao, K., Yuan, H., Qing, Z., Wang, X., Zhao, D., and Zhou, J. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models, 2023.

Zhang, Z., Hu, J., Cheng, W., Paudel, D., and Yang, J. Extdm: Distribution extrapolation diffusion model for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2024.

Zhou, L., Lou, A., Khanna, S., and Ermon, S. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhou, L., Lou, A., Khanna, S., and Ermon, S. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2024a.

Zhou, S., Yang, P., Wang, J., Luo, Y., and Loy, C. C. Upscale-A-Video: Temporal-consistent diffusion model for real-world video super-resolution. In *CVPR*, 2024b.

A. Additional Related Works

Frame Interpolation aims to synthesize middle frames from two inputs, ensuring smoothness and consistency. Traditional methods use optical flow (Niklaus & Liu, 2020; Lee et al., 2020; Park et al., 2021; Huang et al., 2022) or convolutional features with attention (Kalluri et al., 2023; Shi et al., 2022; Reda et al., 2022). Diffusion-based interpolation began with bidirectional masking (Voleti et al., 2022) and advanced through conditional generation (Danier et al., 2024), cascaded refinement (Jain et al., 2024), adapted image-to-video models (Wang et al., 2025a), and large-motion techniques (Shen et al., 2024). However, these methods lack explicit modeling of inter-frame correlations. Event-based approaches (Chen et al., 2025; Zhang et al., 2025a) add motion cues but also use standard diffusion without capturing temporal dependencies.

Image-to-Video Generation creates a video from an input image, requiring consistent and accurate motion. Diffusion models have significantly advanced this field, producing high-quality results (Zhang et al., 2023; Xing et al., 2023; Shi et al., 2024; Guo et al., 2023b; 2024; Arkhipkin et al., 2025b;a). However, the standard noise-to-data diffusion process risks losing essential information from the input image. Some approaches address this within the diffusion framework (Ren et al., 2024; Wu et al., 2023). A recent extension of bridge matching to image-to-video generation, although taking into account temporal correlation between frames, did not modify the interpolant itself (Wang et al., 2025b). In our solution, we address this gap and develop a new prior that explicitly takes into account the correlations between the components of random vectors representing video data samples.

Video Super Resolution reconstructs high-resolution videos from low-resolution inputs. Diffusion models are applied for their strong generative prior, which synthesizes realistic details to overcome degradation. A key challenge is ensuring temporal coherence within the inherently stochastic diffusion process. Recent methods address this by introducing explicit spatiotemporal constraints, such as temporal layer integration and motion-guided losses (Zhou et al., 2024b; Yang et al., 2024). Meanwhile, bridge-matching methods have shown promise for image super-resolution (Liu et al., 2023; Gushchin et al., 2025). In this work, we extend bridge matching to video super resolution, proposing a novel approach that explicitly models temporal coherence between frames.

B. Proof of Propositions

Proof of Proposition 1. Consider the linear SDE

$$d\mathbf{X}_t = (\mathbf{A}\mathbf{X}_t + \mathbf{b})dt + \sqrt{\epsilon}\, d\mathbf{W}_t, \quad \mathbf{X}_0 \sim \delta_{\mathbf{X}_0},$$

with $\mathbf{A} \in \mathbb{R}^{D\times D}$ symmetric and invertible, $\mathbf{b} \in \mathbb{R}^D$, and a D-dimensional standard Wiener process $\mathbf{W}_t$.

Conditional mean. Let $\mathbf{\Phi}(t) := e^{\mathbf{A}t}$ and define $\mathbf{Y}_t := (\mathbf{\Phi}(t))^{-1}\mathbf{X}_t = e^{-\mathbf{A}t}\mathbf{X}_t$ (note $\mathbf{Y}_0 = \mathbf{X}_0$).

$$d\mathbf{Y}_t = d\big(e^{-\mathbf{A}t}\mathbf{X}_t\big) = e^{-\mathbf{A}t}\, d\mathbf{X}_t + d(e^{-\mathbf{A}t})\, \mathbf{X}_t =$$
$$e^{-\mathbf{A}t}\big[(\mathbf{A}\mathbf{X}_t + \mathbf{b})\, dt + \sqrt{\epsilon}\, d\mathbf{W}_t\big] - \mathbf{A}e^{-\mathbf{A}t}\mathbf{X}_t\, dt = e^{-\mathbf{A}t}\mathbf{b}\, dt + \sqrt{\epsilon}\, e^{-\mathbf{A}t} d\mathbf{W}_t,$$

In the integral form:

$$\mathbf{Y}_t = \mathbf{X}_0 + \int_0^t e^{-\mathbf{A}s}\mathbf{b}\, ds + \sqrt{\epsilon}\int_0^t e^{-\mathbf{A}s}\, d\mathbf{W}_s.$$

Multiplying by $\mathbf{\Phi}(t) = e^{\mathbf{A}t}$ yields:

$$\mathbf{X}_t = e^{\mathbf{A}t}\mathbf{X}_0 + \int_0^t e^{\mathbf{A}(t-s)}\mathbf{b} ds + \sqrt{\epsilon}\int_0^t e^{\mathbf{A}(t-s)} d\mathbf{W}_s. \tag{15}$$

Hence, the conditional mean is:

$$\boldsymbol{\mu}_{t|0}(\mathbf{X}_0) = e^{\mathbf{A}t}\mathbf{X}_0 + \Big(\int_0^t e^{\mathbf{A}(t-s)} ds\Big)\mathbf{b} = e^{\mathbf{A}t}X_0 + \big(e^{\mathbf{A}t} - I\big)\mathbf{A}^{-1}\mathbf{b}, \tag{16}$$

Conditional variance.

$$\mathbf{I}_t := \int_0^t e^{A(t-s)}\, d\mathbf{W}_s, \qquad \text{so that} \qquad \mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0) = \sqrt{\epsilon}\, \mathbf{I}_t.$$

$$\mathrm{Cov}(\mathbf{X}_t \mid \mathbf{X}_0) = \mathbb{E}\big[(\mathbf{X}_t - \boldsymbol{\mu}_{t|0})(\mathbf{X}_t - \boldsymbol{\mu}_{t|0})^\top \mid \mathbf{X}_0\big] = \epsilon\,\mathrm{Cov}(\mathbf{I}_t).$$

In turn:

$$\mathrm{Cov}(\mathbf{I}_t) = \mathbb{E}\left[\left(\int_0^t e^{\mathbf{A}(t-s)}\,d\mathbf{W}_s\right)\left(\int_0^t e^{\mathbf{A}(t-r)}\,d\mathbf{W}_r\right)^\top\right].$$

By Itô isometry:

$$\mathbb{E}\left[\int_0^t G_s\,d\mathbf{W}_s\right] = 0, \qquad \mathrm{Cov}\left(\int_0^t G_s\,d\mathbf{W}_s, \int_0^t H_s\,dW_s\right) = \int_0^t G_s H_s^\top\,ds.$$

Taking $G_s = H_s = e^{\mathbf{A}(t-s)}$ gives

$$\mathrm{Cov}(\mathbf{I}_t) = \int_0^t e^{\mathbf{A}(t-s)}\,e^{\mathbf{A}^\top(t-s)}\,ds.$$

Because $\mathbf{A}$ is symmetric, $e^{\mathbf{A}^\top u} = e^{\mathbf{A}u}$, hence

$$\mathrm{Cov}(\mathbf{I}_t) = \int_0^t e^{2\mathbf{A}(t-s)}\,ds = \frac{1}{2}\big(e^{2\mathbf{A}t} - I\big)\mathbf{A}^{-1}.$$

Consequently,

$$\boldsymbol{\Sigma}_{t|0} = \mathrm{Cov}(\mathbf{X}_t \mid \mathbf{X}_0) = \frac{\epsilon}{2}\big(e^{2\mathbf{A}t} - \mathbf{I}\big)\mathbf{A}^{-1}.$$

Since $\mathbf{X}_t \mid \mathbf{X}_0$ is Gaussian with mean $\mu_{t|0}$ and covariance $\Sigma_{t|0}$, its score is

$$\nabla_{\mathbf{X}_t} \log q(\mathbf{X}_t \mid \mathbf{X}_0) = -\,\boldsymbol{\Sigma}_{t|0}^{-1}\big(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\big).$$

This completes the proof. □

Proof of Proposition 2. **Step 1: Joint law from the prior.** From (15) and (16) in the proof of Proposition 1:

$$\mathbf{X}_u = \boldsymbol{\mu}_{u|0}(\mathbf{X}_0) + \sqrt{\epsilon}\int_0^u e^{\mathbf{A}(u-s)}d\mathbf{W}_s,$$

$$\boldsymbol{\mu}_{u|0}(\mathbf{X}_0) = e^{\mathbf{A}u}\mathbf{X}_0 + \big(e^{\mathbf{A}u} - \mathbf{I}\big)\mathbf{A}^{-1}\mathbf{b}.$$

Thus, conditionally on $\mathbf{X}_0$,

$$\mathbb{E}[\mathbf{X}_t \mid \mathbf{X}_0] = \boldsymbol{\mu}_{t|0}(\mathbf{X}_0), \qquad \mathbb{E}[\mathbf{X}_{t'} \mid \mathbf{X}_0] = \boldsymbol{\mu}_{t'|0}(\mathbf{X}_0),$$
$$\boldsymbol{\Sigma}_{t|0} = \frac{\epsilon}{2}\big(e^{2\mathbf{A}t} - \mathbf{I}\big)\mathbf{A}^{-1}, \qquad \boldsymbol{\Sigma}_{t'|0} = \frac{\epsilon}{2}\big(e^{2\mathbf{A}t'} - \mathbf{I}\big)\mathbf{A}^{-1}.$$

For the cross-covariance, using Itô isometry and independence of increments, for $t < t'$,

$$\boldsymbol{\Sigma}_{t,t'|0} = \mathrm{Cov}(\mathbf{X}_t, \mathbf{X}_{t'} \mid \mathbf{X}_0) = \epsilon\int_0^t e^{\mathbf{A}(t-s)}\,e^{\mathbf{A}(t'-s)}\,ds = \epsilon\int_0^t e^{\mathbf{A}(t+t'-2s)}\,ds = \frac{\epsilon}{2}\,\mathbf{A}^{-1}\Big(e^{\mathbf{A}(t+t')} - e^{\mathbf{A}(t'-t)}\Big).$$

Collecting blocks, we have the joint Gaussian (conditionally on X_0)

$$\begin{bmatrix}\mathbf{X}_t \\ \mathbf{X}_{t'}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{t|0}(\mathbf{X}_0) \\ \boldsymbol{\mu}_{t'|0}(\mathbf{X}_0)\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{t|0} & \boldsymbol{\Sigma}_{t,t'|0} \\ \boldsymbol{\Sigma}_{t,t'|0} & \boldsymbol{\Sigma}_{t'|0}\end{bmatrix}\right).$$

Step 2: Conditioning to obtain the bridge. For a joint Gaussian $\begin{bmatrix}x \\ y\end{bmatrix}$ with blocks $(\mu_x, \mu_y, \Sigma_{xx}, \Sigma_{yy}, \Sigma_{xy})$, the conditional $x \mid y$ is (Petersen et al., 2008, Section 8.1.3):

$$x \mid y \sim \mathcal{N}\big(\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\big).$$

Applying this with $x = \mathbf{X}_t$, $y = \mathbf{X}_{t'}$ and the blocks above gives:

$$\boldsymbol{\mu}_{t|0,t'} = \boldsymbol{\mu}_{t|0}(\mathbf{X}_0) + \boldsymbol{\Sigma}_{t,t'|0}\,\boldsymbol{\Sigma}_{t'|0}^{-1}\big(\mathbf{X}_{t'} - \boldsymbol{\mu}_{t'|0}(\mathbf{X}_0)\big),$$
$$\boldsymbol{\Sigma}_{t|0,t'} = \boldsymbol{\Sigma}_{t|0} - \boldsymbol{\Sigma}_{t,t'|0}\,\boldsymbol{\Sigma}_{t'|0}^{-1}\boldsymbol{\Sigma}_{t,t'|0}.$$

Here

$$\mathbf{\Sigma}_{t|0} = \tfrac{\epsilon}{2}(e^{2\mathbf{A}t} - \mathbf{I})\mathbf{A}^{-1}, \quad \mathbf{\Sigma}_{t'|0} = \tfrac{\epsilon}{2}(e^{2\mathbf{A}t'} - \mathbf{I})\mathbf{A}^{-1}, \quad \mathbf{\Sigma}_{t,t'|0} = \tfrac{\epsilon}{2}A^{-1}\big(e^{\mathbf{A}(t+t')} - e^{\mathbf{A}(t'-t)}\big).$$

This completes the proof. □

Proof of Proposition 3. Consider the following bijective reparameterization:

$$v_\phi(\mathbf{X}_t, t) = -\mathbf{\Sigma}_{t|0}^{-1}\left(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t))\right)$$

and substitute it in the optimization problem:

$$\min_\phi \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\left\|v_\phi(\mathbf{X}_t, t) + \mathbf{\Sigma}_{t|0}^{-1}(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0))\right\|^2\right],$$

$$\min_\phi \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\left\|-\mathbf{\Sigma}_{t|0}^{-1}\left(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t))\right) + \mathbf{\Sigma}_{t|0}^{-1}(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0))\right\|^2\right] =$$
$$\min_\phi \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\left\|\mathbf{\Sigma}_{t|0}^{-1}\left((\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)) - \left(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t))\right)\right)\right\|^2\right] =$$
$$\min_\phi \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\left\|\mathbf{\Sigma}_{t|0}^{-1}\left(\boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)) - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\right)\right\|^2\right] =$$
$$\min_\phi \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\left(\boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)) - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\right)^\top (\mathbf{\Sigma}_{t|0}^\top)^{-1}\mathbf{\Sigma}_{t|0}^{-1}\left(\boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)) - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\right)\right]$$

Taking the gradient of this objective with respect to ϕ, we obtain:

$$\mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[2(\mathbf{\Sigma}_{t|0}^\top)^{-1}\mathbf{\Sigma}_{t|0}^{-1}\left(\boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)) - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\right)\right] = 0$$

Since $\mathbf{\Sigma}_{t|0}^{-1}$ is positive definite we can multiply by $\frac{1}{2}\mathbf{\Sigma}_{t|0}(\mathbf{\Sigma}_{t|0}^\top)$ and get:

$$\mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)) - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\right] = 0$$

Then for each $\mathbf{X}_t$ consider conditional mean:

$$\mathbb{E}_{\mathbf{X}_t,t}\left[\mathbb{E}_{\mathbf{X}_0|\mathbf{X}_t}\left[\boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)) - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\right]\right] = 0$$

$$\mathbb{E}_{\mathbf{X}_0|\mathbf{X}_t}\left[\boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_t, t)) - \boldsymbol{\mu}_{t|0}(\mathbf{X}_0)\right] = 0$$

From (16) we have:

$$\mu_{t|0}(\mathbf{X}_0) = e^{\mathbf{A}t}\mathbf{X}_0 + \Big(\int_0^t e^{\mathbf{A}(t-s)}ds\Big)\mathbf{b} = e^{\mathbf{A}t}X_0 + \big(e^{\mathbf{A}t} - I\big)\mathbf{A}^{-1}\mathbf{b},$$

Then (note that $e^{\mathbf{A}t}$ is invertible and we can multiplu both sides on e^{-At}):

$$\mathbb{E}_{\mathbf{X}_0|\mathbf{X}_t}\left[e^{\mathbf{A}t}\mathbf{X}_0^\phi(\mathbf{X}_t, t) + \big(e^{\mathbf{A}t} - I\big)\mathbf{A}^{-1}\mathbf{b} - e^{\mathbf{A}t}\mathbf{X}_0 + \big(e^{\mathbf{A}t} - I\big)\mathbf{A}^{-1}\mathbf{b}\right] = 0$$
$$\mathbb{E}_{\mathbf{X}_0|\mathbf{X}_t}\left[e^{\mathbf{A}t}(\mathbf{X}_0^\phi(\mathbf{X}_t, t) - \mathbf{X}_0)\right] = 0$$
$$\mathbb{E}_{\mathbf{X}_0|\mathbf{X}_t}\left[\mathbf{X}_0^\phi(\mathbf{X}_t, t) - \mathbf{X}_0\right] = 0$$
$$\mathbf{X}_0^\phi(\mathbf{X}_t, t) = \mathbb{E}_{\mathbf{X}_0|\mathbf{X}_t}\left[\mathbf{X}_0\right]$$

Hence, optimal $\mathbf{X}_0^* = \mathbb{E}_{\mathbf{X}_0|\mathbf{X}_t}[\mathbf{X}_0]$, which in turn is the minimizer of MSE problem:

$$\min_{\phi} \mathbb{E}_{\mathbf{X}_0,\mathbf{X}_t,t}\left[\|\widehat{\mathbf{X}}_0^{\phi}(\mathbf{X}_t,t) - \mathbf{X}_0\|^2\right].$$

By substituting in to v_ϕ we have:

$$v^*(\mathbf{X}_t,t) = -\mathbf{\Sigma}_{t|0}^{-1}\left(\mathbf{X}_t - \boldsymbol{\mu}_{t|0}(\widehat{\mathbf{X}}_0^*(\mathbf{X}_t,t))\right)$$

This completes the proof. □

C. Additional Quantitative Results

C.1. Confidence Intervals

Here, we present the results of a quantitative comparison between DDPM, DDIM, Bridge Matching (BM), and our proposed method, TCVBM. The values of the standard deviation are provided, based on 3 runs of each method with different random seeds.

Table 6. Frame interpolation quantitative results with standard deviation. The best values in column are bold, second best values are underlined.

Metric	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
DDIM	34.664 ± 5.80	0.120 ± 0.070	15.843 ± 0.120	0.766 ± 0.011
DDPM	<u>33.612 ± 1.494</u>	0.107 ± 0.009	14.509 ± 0.427	0.714 ± 0.024
BM	34.766 ± 0.398	<u>0.078 ± 0.001</u>	<u>17.265 ± 0.390</u>	<u>0.789 ± 0.005</u>
TCVBM (ours)	**31.491 ± 4.035**	**0.071 ± 0.019**	**17.451 ± 0.459**	**0.825 ± 0.044**

Table 7. Image-to-Video generation quantitative results with standard deviation. The best values in column are bold, second best values are underlined.

Metric	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
DDIM	335.51 ± 241.12	0.402 ± 0.092	10.205 ± 0.514	0.513 ± 0.069
DDPM	250.52 ± 134.99	0.383 ± 0.054	10.333 ± 0.275	0.530 ± 0.046
BM	<u>48.54 ± 0.56</u>	<u>0.268 ± 0.005</u>	<u>10.627 ± 0.053</u>	<u>0.582 ± 0.004</u>
TCVBM (ours)	**45.32 ± 0.91**	**0.260 ± 0.002**	**10.710 ± 0.028**	**0.589 ± 0.001**

Table 8. Video super resolution quantitative results with standard deviation. The best values in column are bold, second best values are underlined.

Metric	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
DDIM	336.808 ± 5.175	0.520 ± 0.006	17.226 ± 0.103	0.600 ± 0.016
DDPM	614.288 ± 6.289	0.237 ± 0.001	20.152 ± 0.050	0.577 ± 0.004
BM	**29.710 ± 20.683**	**0.026 ± 0.005**	<u>21.412 ± 1.040</u>	<u>0.941 ± 0.012</u>
TCVBM (ours)	<u>32.762 ± 23.153</u>	<u>0.029 ± 0.004</u>	**21.431 ± 1.419**	**0.941 ± 0.011**

C.2. Human Evaluation on Moving MNIST

We conducted an additional user study on 40 random generated examples using initial frames from the MovingMNIST test dataset. The results are presented in Table 9.

D. Initialization Experiments

In this section, we explore options for initializing or representing input data for bridge-based methods used in our work, namely for Bridge Matching with Browninan Bridge (BM) and Time-Correlated Video Bridge Matching (TCVBM). The

Table 9. Human evaluation results on the MovingMNIST dataset.

Method	Frame interpolation		Image-to-video generation		Video super resolution	
	FrameQuality	TemporalConsistency	FrameQuality	TemporalConsistency	FrameQuality	TemporalConsistency
BM	17%	20.5%	28%	18.5%	11.5%	13.5%
TCVBM	36%	52.5%	51%	59.5%	19%	30.5%
Both	47%	27%	21%	22%	69.5%	56%

interest in exploring the effect of input data initialization on the quality of model performance stems primarily from the assumption that bridge-based approaches are better suited for data-to-data translation tasks.

D.1. Frame Interpolation

As input data for the network, we explored two options: filling in intermediate frames with Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and using linear interpolation between fixed boundary frames $\mathbf{x}^0$ and x^N, i.e.:

$$\mathbf{x}^n_{input} = \frac{n\mathbf{x}^0 + (N-n)x^N}{N}, \quad n = 1, \dots, N-1.$$

Table 10 compares the results of these initialization methods. As can be seen, filling intermediate frames with noise from a normal distribution produces better results than the initial linear interpolation.

Table 10. Analysis of the impact of initialization of input video data for bridge-based methods in the task of frame interpolation. Using noise from a normal distribution shows a clear advantage. The best values in column are bold, second best values are underlined.

Initialization method	**Method**	**FVD** ↓	**LPIPS** ↓	**PSNR** ↑	**SSIM** ↑
Linear interpolation	BM	34.804	0.109	15.439	0.756
	TCVBM	<u>31.944</u>	0.092	16.275	0.782
Gaussian noise from $\mathcal{N}(\mathbf{0}, \mathbf{1})$	BM	34.315	<u>0.079</u>	<u>17.103</u>	<u>0.794</u>
	TCVBM	**30.542**	**0.077**	**17.280**	**0.813**

D.2. Image-to-Video Generation

Here we compare the following two types of initial initialization: duplicating the first frame in place of the frames to be generated (static video) and using random noise everywhere except the first frame. Static video initialization is superior to the noise option for both models (Table 11).

Table 11. Comparison of two types of initial initialization of input data for image-to-video generation.

Initialization method	**Method**	**FVD** ↓	**LPIPS** ↓	**PSNR** ↑	**SSIM** ↑
Static video	BM	49.32	0.271	10.63	0.579
	TCVBM	**44.96**	**0.258**	**10.75**	**0.591**
Gaussian noise from $\mathcal{N}(\mathbf{0}, \mathbf{1})$	BM	52.57	0.287	10.61	0.568
	TCVBM	<u>48.61</u>	<u>0.263</u>	<u>10.68</u>	<u>0.587</u>

D.3. Video Super Resolution

Table 12. Comparison of two types of initial initialization of input data for video super resolution. We perform this comparison for low-resolution 32×32.

Initialization method	**Method**	**FVD** ↓	**LPIPS** ↓	**PSNR** ↑	**SSIM** ↑
Low-resolution video	BM	<u>9.501</u>	<u>0.014</u>	24.888	<u>0.972</u>
	TCVBM	**9.496**	**0.012**	24.970	**0.973**
Low-resolution video	BM	9.556	**0.012**	<u>24.892</u>	**0.973**
concatenated with noise from $\mathcal{N}(\mathbf{0}, \mathbf{1})$	TCVBM	9.646	**0.012**	**24.988**	**0.973**

E. Hyperparameters Searching

Here we compare different values of hyperparameters, namely the noise scaling value ϵ and the coefficient α, which determines the degree of impact of the matrix $\mathbf{A}$ as $\widetilde{\mathbf{A}} := \alpha\mathbf{A}$. Tables 13 and 14 show that in the case of frame interpolation and image-to-video generation, it is impossible to identify a clear dependence of the generation quality on the hyperparameters used, however, a sufficient amount of noise and not large values for the α coefficient are optimal. The results for video super resolution in Table 15 demonstrate that small values of ϵ and α are optimal for this task, which does not contradict the results for frame interpolation.

Table 13. The results of TCVBM training with various hyperparameters ϵ and α for frame interpolation. The best values in column are bold, second best values are underlined.

ϵ	α	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
0.1	0.1	35.572	0.085	16.40	0.797
0.1	1	36.542	0.089	16.37	0.797
0.1	10	<u>29.792</u>	0.086	16.56	0.801
1	0.1	**27.342**	0.084	16.65	0.803
1	1	30.542	**0.077**	16.86	0.813
1	10	31.432	<u>0.080</u>	17.12	0.817
10	0.1	37.662	0.084	**17.24**	**0.819**
10	1	54.542	0.093	<u>17.17</u>	<u>0.818</u>
10	10	54.562	0.100	<u>17.17</u>	0.769

Table 14. The results of TCVBM training with various hyperparameters ϵ and α for image-to-video generation. The best values in column are bold, second best values are underlined.

ϵ	α	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
0.1	0.1	55.46	0.2670	**10.80**	0.587
0.1	1	51.62	0.2578	10.74	0.583
0.1	10	59.57	**0.2571**	10.67	0.585
1	0.1	51.80	0.2615	10.70	0.590
1	1	44.96	<u>0.2575</u>	<u>10.75</u>	<u>0.591</u>
1	10	51.27	0.2613	10.60	0.583
10	0.1	44.90	0.2610	10.67	0.588
10	1	**44.22**	0.2584	10.70	**0.591**
10	10	<u>44.58</u>	0.2597	10.58	0.583

Table 15. The results of TCVBM training with various hyperparameters ϵ and α for video super resolution. We perform this comparison for low-resolution 32×32.

ϵ	α	FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
0.1	0.1	**9.496**	**0.012**	**24.970**	**0.973**
0.1	1	<u>10.413</u>	<u>0.013</u>	<u>24.358</u>	<u>0.969</u>
1	0.1	13.226	0.019	23.004	0.959
1	1	15.023	0.022	22.120	0.949
10	0.1	18.458	0.033	20.085	0.921
10	1	20.746	0.039	19.283	0.906

F. Computational Cost Analysis

As described in Algorithm 2, inference consists of two alternating steps: (i) computing $\widehat{\mathbf{X}}_0^\phi(\mathbf{X}_{t_n}, t_n)$ and (ii) sampling from the Gaussian distribution given in Eq. 9. Sampling can be done using reparameterization:

$$\mathbf{X}_{t'} = \mu_{t|0,t'}(\mathbf{X}_0) + (\mathbf{\Sigma}^{1/2}_{t|0,t'})^\top \mathbf{Z}, \quad \mathbf{Z} \sim \mathcal{N}(0, I),$$

which requires the evaluation of the mean and covariance:

$$\mu_{t|0,t'} = \mu_{t|0}(\mathbf{X}_0) + \mathbf{\Sigma}_{t|0}\mathbf{\Sigma}^{-1}_{t'|0}(\mathbf{X}_{t'} - \mu_{t'|0}(\mathbf{X}_0)),$$

$$\mathbf{\Sigma}_{t|0,t'} = \mathbf{\Sigma}_{t|0} - \mathbf{\Sigma}_{t|0}\mathbf{\Sigma}^{-1}_{t'|0}\mathbf{\Sigma}_{t|0}.$$

In turn,

$$\mu_{t|0}(\mathbf{X}_0) = e^{\mathbf{A}t}\mathbf{X}_0 + \left(e^{\mathbf{A}t} - I\right)\mathbf{A}^{-1}\mathbf{b}, \quad \mathbf{\Sigma}_{t|0} = \epsilon\frac{e^{2\mathbf{A}t} - I}{2}\mathbf{A}^{-1}.$$

In total, almost all these operations can be cached except for 5 matrix multiplications: $(\mathbf{\Sigma}^{1/2}_{t|0,t'})^\top\mathbf{Z}$, $e^{\mathbf{A}t}\mathbf{X_0}$, $(\mathbf{\Sigma}_{t|0}\mathbf{\Sigma}^{-1}_{t'|0})\mathbf{X}_{t'}$, $e^{\mathbf{A}t'}\mathbf{X_0}$, $(\mathbf{\Sigma}_{t|0}\mathbf{\Sigma}^{-1}_{t'|0})(e^{\mathbf{A}t'}\mathbf{X_0})$. All these multiplications are for tensors of size $F \times F$ (F is the number of frames in the video) with a tensor of size $F \times C \times H \times W$, where C is the number of channels, H is the height, and W is the width of the video. This requires $O(F^2CHW)$ operations. For comparison, a single convolutional layer with C input channels, C_{out} output channels, and kernel size $K \times K$ applied to all F frames requires $O(FCHWK^2C_{\text{out}})$ operations. Since typically $F \sim 10{-}100$, $K^2 \sim 10$, and $C_{\text{out}} \sim 10$, we have

$$\frac{F^2CHW}{FCHWK^2C_{\text{out}}} = \frac{F}{K^2C_{\text{out}}} \sim 0.1 - 1.$$

Thus, each of these 5 operations is comparable to requiring less computation than 1 convolution layer. Since a neural network requires many convolutional layers and additional nonlinear processing, the resulting overhead of our bridge update is comparable to only a few convolutional layers and therefore remains relatively small in practice.

G. Dynamical Correlation

G.1. Theory

Consider prior SDE with an additional multiplicative function $f(t)$ depending only on time t:

$$d\mathbf{X}_t = f(t)(\mathbf{A}\mathbf{X}_t + \mathbf{b})dt + \sqrt{\epsilon}d\mathbf{W}_t.$$

The derivation of formulas for this prior follows the same principles used for $f(t) = 1$ in Appendix B.

Define:

$$F(t) = \int_0^t f(\tau)\,d\tau, \qquad \mathbf{\Phi}(t) = e^{\mathbf{A}\,F(t)}.$$

Then $\frac{d}{dt}\mathbf{\Phi}(t) = f(t)\,\mathbf{A}\,\mathbf{\Phi}(t)$, $\mathbf{\Phi}(0) = I$, and $\frac{d}{dt}\mathbf{\Phi}(t)^{-1} = -\,f(t)\,\mathbf{\Phi}(t)^{-1}\mathbf{A}$.

Conditional mean. Consider $\mathbf{Y}_t := \mathbf{\Phi}(t)^{-1}\mathbf{X}_t$. By Itô's rule:

$$d\mathbf{Y}_t = \mathbf{\Phi}(t)^{-1}d\mathbf{X}_t + d(\mathbf{\Phi}(t)^{-1})\,\mathbf{X}_t = \mathbf{\Phi}(t)^{-1}f(t)(\mathbf{A}\mathbf{X}_t + \mathbf{b})\,dt + \sqrt{\epsilon}\,\mathbf{\Phi}(t)^{-1}d\mathbf{W}_t - f(t)\,\mathbf{\Phi}(t)^{-1}A\mathbf{X}_t\,dt =$$
$$= \mathbf{\Phi}(t)^{-1}f(t)\mathbf{b}\,dt + \sqrt{\epsilon}\,\mathbf{\Phi}(t)^{-1}d\mathbf{W}_t.$$

Integrating from 0 to t gives

$$\mathbf{Y}_t = \mathbf{X}_0 + \int_0^t \mathbf{\Phi}(s)^{-1}f(s)\,\mathbf{b}\,ds + \sqrt{\epsilon}\int_0^t \mathbf{\Phi}(s)^{-1}d\mathbf{W}_s,$$

and thus

$$\mathbf{X}_t = \mathbf{\Phi}(t)\mathbf{X}_0 + \mathbf{\Phi}(t)\int_0^t \mathbf{\Phi}(s)^{-1}f(s)\,\mathbf{b}\,ds + \sqrt{\epsilon}\,\mathbf{\Phi}(t)\int_0^t \mathbf{\Phi}(s)^{-1}d\mathbf{W}_s.$$

Taking expectation and using $\mathbf{\Phi}(t)\mathbf{\Phi}(s)^{-1} = e^{\mathbf{A}(F(t)-F(s))}$,

$$\boldsymbol{\mu}_{t|0}(\mathbf{X}_0) = \mathbb{E}[\mathbf{X}_t|\mathbf{X}_0] = e^{\mathbf{A}F(t)}\mathbf{X}_0 + \int_0^t e^{\mathbf{A}(F(t)-F(s))}f(s)\,\mathbf{b}\,ds.$$

With the change of variables $u = F(s)$ (so $du = f(s)\,ds$) this equals

$$\int_0^{F(t)} e^{\mathbf{A}(F(t)-u)}\,du\,\mathbf{b} = \Big[-\,e^{\mathbf{A}(F(t)-u)}\mathbf{A}^{-1}\Big]_{u=0}^{F(t)}\mathbf{b} = \big(e^{\mathbf{A}F(t)} - I\big)\mathbf{A}^{-1}\mathbf{b}.$$

Therefore

$$\boldsymbol{\mu}_{t|0}(\mathbf{X}_0) = e^{\mathbf{A}F(t)}\mathbf{X}_0 + \big(e^{\mathbf{A}F(t)} - I\big)\mathbf{A}^{-1}\mathbf{b}.$$

The mean $\mu_{t|0}(X_0)$ of the process starting from a given $\mathbf{X}_0$ is given by:

$$\boldsymbol{\mu}_{t|0}(\mathbf{X}_0) = e^{\mathbf{A}F(t)}\mathbf{X}_0 + \left(e^{\mathbf{A}F(t)} - I\right)\mathbf{A}^{-1}\mathbf{b}.$$

Conditional variance. Let $\mathbf{Z}_t := \mathbf{X}_t - \boldsymbol{\mu}_{t|0}$ be the centered process. Subtracting the mean SDE from the original SDE yields

$$d\mathbf{Z}_t = f(t)\,\mathbf{A}\,\mathbf{Z}_t\,dt + \sqrt{\epsilon}\,d\mathbf{W}_t.$$

Define the covariance $\boldsymbol{\Sigma}_{t|0} := \mathbb{E}[\mathbf{Z}_t\mathbf{Z}_t^\top]$. Using Itô's rule for $\mathbf{Z}_t\mathbf{Z}_t^\top$,

$$d(\mathbf{Z}_t\mathbf{Z}_t^\top) = (d\mathbf{Z}_t)\mathbf{Z}_t^\top + \mathbf{Z}_t(d\mathbf{Z}_t)^\top + d\mathbf{Z}_t(d\mathbf{Z}_t)^\top,$$

where

$$(d\mathbf{Z}_t)\,\mathbf{Z}_t^\top = f(t)\mathbf{A}\mathbf{Z}_t\mathbf{Z}_t^\top\,dt + \sqrt{\epsilon}\,d\mathbf{W}_t\,\mathbf{Z}_t^\top, \tag{17}$$

$$\mathbf{Z}_t\,(d\mathbf{Z}_t)^\top = f(t)\mathbf{Z}_t\mathbf{Z}_t^\top\mathbf{A}^\top\,dt + \sqrt{\epsilon}\,\mathbf{Z}_t\,d\mathbf{W}_t^\top, \tag{18}$$

$$(d\mathbf{Z}_t)(d\mathbf{Z}_t)^\top = \epsilon\,d\mathbf{W}_t d\mathbf{W}_t^\top = \epsilon\,I\,dt. \tag{19}$$

Hence

$$d(\mathbf{Z}_t\mathbf{Z}_t^\top) = f(t)\big(\mathbf{A}\mathbf{Z}_t\mathbf{Z}_t^\top + \mathbf{Z}_t\mathbf{Z}_t^\top\mathbf{A}^\top\big)\,dt + \sqrt{\epsilon}\,d\mathbf{W}_t\,\mathbf{Z}_t^\top + \sqrt{\epsilon}\,\mathbf{Z}_t\,d\mathbf{W}_t^\top + \epsilon\,I\,dt.$$

and taking expectations gives

$$\frac{d}{dt}\boldsymbol{\Sigma}_{t|0} = f(t)\,\mathbf{A}\,\boldsymbol{\Sigma}_{t|0} + f(t)\,\boldsymbol{\Sigma}_{t|0}\mathbf{A}^\top + \epsilon\,I, \quad \boldsymbol{\Sigma}_{t|0} = 0.$$

To get the cross-covariance, we use:

$$\mathbf{Z}_t = \sqrt{\epsilon}\int_0^t e^{\mathbf{A}(F(t)-F(s))}\,d\mathbf{W}_s, \qquad \mathbf{Z}_{t'} = \sqrt{\epsilon}\int_0^{t'} e^{\mathbf{A}(F(t')-F(u))}\,d\mathbf{W}_u,$$

The Itô isometry yields

$$\boldsymbol{\Sigma}_{t,t'|0} := \mathrm{Cov}(\mathbf{Z}_t, \mathbf{Z}_{t'}^\top) = \epsilon\int_0^t e^{\mathbf{A}(F(t)-F(s))}\,e^{\mathbf{A}(F(t')-F(s))^\top}\,ds.$$

For symmetric $\mathbf{A}$, $e^{(\cdot)^\top} = e^{(\cdot)}$ and, since these exponentials commute (all are $e^{\mathbf{A}(\cdot)}$),

$$e^{\mathbf{A}(F(t)-F(s))}\,e^{\mathbf{A}(F(t')-F(s))} = e^{\mathbf{A}(F(t')-F(t))}\,e^{2\mathbf{A}(F(t)-F(s))}.$$

Hence

$$\boldsymbol{\Sigma}_{t,t'|0} = e^{\mathbf{A}(F(t')-F(t))}\,\epsilon\int_0^t e^{2\mathbf{A}(F(t)-F(s))}\,ds = e^{\mathbf{A}\,[F(t')-F(t)]}\,\boldsymbol{\Sigma}_{t|0}.$$

Summary. Thus, all three components: mean $\boldsymbol{\mu}_{t|0}(\mathbf{X}_0)$, variance $\boldsymbol{\Sigma}_{t|0}$, and cross-covariance $\boldsymbol{\Sigma}_{t,t'|0}$ are derived and can be used further in the same way as in the original case of $f(t) = 1$.

G.2. Experimental Results

The continuous and time-decreasing function $f(t)$ sets the increasing values of the matrix $\mathbf{A}$ in the inverse diffusion process when $t \to 0$. Thus, the correlation of frames with each other in the generated video increases in the last steps of the inference. We conduct a series of experiments to investigate the effect of dynamic correlation and the choice of the function $f(t)$. We use the same experimental setting for Moving MNIST dataset as described in Section 5.2.1 of the main paper, with a number of optimizer steps set to $120,000$. The frame interpolation results are presented in Table 16. As can be seen, our experiments do not demonstrate the advantages of using dynamic correlation compared to using constant values of the matrix $\mathbf{A}$. However, we observe significant differences in the quality of the results depending on the function $f(t)$. This demonstrates a complex structure of the relationship between video frames in the diffusion process, which our framework provides in the constant linear approximation.

Table 16. The results of the selection of the function $f(t)$, which determines the inverse dependence of the values of the matrix $\mathbf{A}$ on time t. The best values in column are bold, second best values are underlined.

$f(t)$	**FVD** ↓	**LPIPS** ↓	**PSNR** ↑	**SSIM** ↑
$1-t$	49.569	0.115	<u>13.938</u>	<u>0.752</u>
$1-2t$	398.823	0.202	11.760	0.684
$1-0.5t$	427.942	0.226	12.063	0.620
$2\times(1-t)$	409.495	0.195	12.239	0.624
$0.5\times(1-t)$	269.899	0.182	12.297	0.676
$(1-t)^2$	<u>43.651</u>	<u>0.112</u>	13.916	0.751
e^{-t}	1910.002	0.973	3.600	0.002
e^{-2t}	1216.250	0.629	7.243	0.119
e^{-4t}	108.052	0.142	13.186	0.688
e^{-8t}	50.078	0.128	13.321	0.726
Constant ($f(t)=\alpha=1$)	**36.309**	**0.097**	**14.515**	**0.772**