VideoTitans: Scalable Video Prediction with Integrated Short- and Long-term Memory

Young-Jae Park
Department of AI Convergence
GIST
youngjae.park@gm.gist.ac.kr

Minseok Seo
School of Electrical Engineering
KAIST
minseok.seo@kaist.ac.kr

Hae-Gon Jeon[*]
Department of Artificial Intelligence
Yonsei University
earboll@yonsei.ac.kr

Abstract

Accurate video forecasting enables autonomous vehicles to anticipate hazards, robotics and surveillance systems to predict human intent, and environmental models to issue timely warnings for extreme weather events. However, existing methods remain limited: transformers rely on global attention with quadratic complexity, making them impractical for high-resolution, long-horizon video prediction, while convolutional and recurrent networks suffer from short-range receptive fields and vanishing gradients, losing key information over extended sequences. To overcome these challenges, we introduce *VideoTitans*, the first architecture to adapt the gradient-driven *Titans* memory—originally designed for language modelling to video prediction. VideoTitans integrates three core ideas: (i) a sliding-window attention core that scales linearly with sequence length and spatial resolution, (ii) an episodic memory that dynamically retains only informative tokens based on a gradient-based *surprise* signal, and (iii) a small set of persistent tokens encoding task-specific priors that stabilize training and enhance generalization. Extensive experiments on Moving-MNIST, Human3.6M, TrafficBJ and WeatherBench benchmarks show that VideoTitans consistently reduces computation (FLOPs) and achieves competitive visual fidelity compared to state-of-the-art recurrent, convolutional, and efficient-transformer methods. Comprehensive ablations confirm that each proposed component contributes significantly.

1 Introduction

Accurate video forecasting enables proactive decision-making in critical real-world systems such as autonomous driving [22, 24, 73], city-scale surveillance [15, 40, 41], robotics [29, 25, 52], and weather forecasting [18, 42, 43]. Predicting future video frames demands a delicate balance: the model must precisely capture rapid, subtle changes between frames [6], yet retain memory of important events and scene dynamics over extended time horizons. Traditional approaches rely heavily on convolutional [28, 61, 54, 69] or recurrent architectures [66, 71], which handle local dynamics effectively but face challenges due to limited receptive fields [7, 47] and vanishing gradients [64, 33], severely restricting their performance on long sequences.

[*] Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Recent transformer-based architectures address these limitations by employing global self-attention [1, 32, 9, 21] to capture long-range dependencies. However, this comes at the prohibitive cost of quadratic computational complexity [76, 3], making them infeasible for realistic, high-resolution, long-sequence applications. Attempts to circumvent this computational bottleneck—such as hierarchical window attention [38], low-rank approximation [34], or external memory [12]—introduce rigid architectural constraints and task-specific heuristics, limiting their generalizability and flexibility across diverse forecasting scenarios.

An alternative approach emerges from recent advances in natural-language processing. Titans [2], a gradient-driven episodic memory module, selectively commits information to memory only when its loss gradient signals substantial "surprise"—a mechanism motivated by the way humans tend to remember unexpected or novel events [35]. This memory mechanism naturally aligns with video prediction tasks, where redundant frame sequences dominate, punctuated by critical rare events such as sudden object movements or abrupt camera motions. However, adapting Titans directly to video forecasting is non-trivial: handling high-resolution frames substantially inflates memory complexity, standard transformer attention remains a computational bottleneck, and visual forecasting benefits significantly from learned, static priors which episodic memory alone cannot provide.

In this paper, we introduce *VideoTitans*, the first architecture to successfully adapt the Titans gradient-driven memory to the dense video forecasting domain. VideoTitans uniquely integrates three core components into a unified, computationally efficient framework: (i) a lightweight sliding-window attention core whose complexity grows linearly with sequence length and spatial resolution, (ii) a gradient-based episodic memory that selectively encodes surprising patch tokens, and (iii) a small set of persistent tokens that inject input-agnostic, reusable priors into the prediction pipeline. The interplay of these modules is seamlessly coordinated by a single gating mechanism, ensuring the predictor remains end-to-end differentiable without reliance on manually tuned heuristics.

We conduct extensive evaluations across diverse and challenging benchmarks—Moving-MNIST [53], Human3.6M [26], TrafficBJ [74], and WeatherBench [45]—demonstrating that VideoTitans consistently reduces computational load while delivering superior visual fidelity in long-range forecasts compared to state-of-the-art recurrent, convolutional, and efficient-transformer methods. Our comprehensive ablation studies further verify that each proposed component is critical to achieving this performance. To facilitate further research and ensure full reproducibility, we will publicly release our source code, trained checkpoints, and demonstration videos.

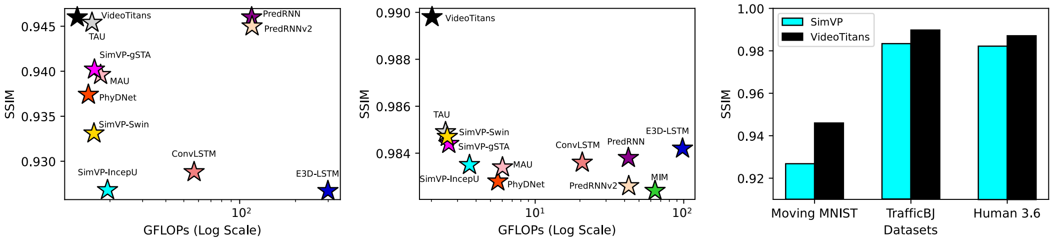
Contributions

- We demonstrate that gradient-driven Titans memory can be applied to long, high-resolution video sequences without incurring quadratic computational growth, providing the first cross-domain evidence of its effectiveness beyond language.
- We present a unified memory–attention architecture that balances efficiency and temporal coverage by combining sliding-window attention, episodic memory and persistent priors behind a single gating mechanism.
- Extensive experiments on Moving-MNIST, Human3.6M, TrafficBJ and WeatherBench show consistent reductions in computation and improvements in long-range visual fidelity over state-of-the-art recurrent, convolutional, and efficient-transformer baselines, while ablation studies confirm that every component of VideoTitans is indispensable.

2 Related Works

2.1 Memory in RNNs and Transformers

Recurrent neural networks (RNNs) [70, 36, 55] and their variants, such as Long Short-Term Memory (LSTM) [23, 48, 77] and Gated Recurrent Units (GRU) [10, 11, 13], have been widely used for modeling sequential dependencies in video prediction [37, 39, 16, 71]. These architectures utilize internal memory to retain past information, enabling them to capture long-range dependencies [56, 31]. However, they suffer from vanishing gradients and sequential processing constraints, limiting their scalability to long video sequences [44]. While various enhancements have been proposed to improve memory retention, RNN-based approaches remain computationally inefficient for high-dimensional spatio-temporal modeling [27, 5].



(a) MMNIST: GFLOPs-SSIM (b) TrafficBJ: GFLOPs-SSIM (c) SSIM Comparison

Figure 1: Performance and efficiency comparisons among video prediction models. Compared to other video prediction models on benchmark datasets, VideoTitans achieves lower FLOPs(G) while delivering superior or comparable performance.

Transformer-based architectures have gained prominence for their ability to model long-term dependencies through self-attention mechanisms, enabling parallelized sequence processing [62, 60, 20]. However, standard attention scales quadratically with sequence length, making it impractical for long video sequences. To address the issue, memory-augmented transformers incorporate external memory to store and retrieve key representations, reducing computational overhead while preserving global contextual information [72, 4, 30]. Despite these improvements, challenges remain in retrieval efficiency and adaptability to dynamic dependencies [59, 63]. Our work builds upon these advancements by introducing a neural long-term memory module that selectively retains critical past information, improving both efficiency and predictive robustness for video forecasting.

2.2 Video Prediction

Video prediction involves forecasting future frames based on past observed frames by modeling intricate spatio-temporal dependencies. ConvLSTM [51] introduced convolutional recurrent units to jointly capture spatial and temporal contexts but struggled with long-term stability. PredRNN [64] and its variants [65, 68] significantly improved temporal modeling by incorporating additional spatio-temporal memory units but came with considerable computational overhead. E3D-LSTM [66] further enhanced performance by integrating 3D convolutions, yet remained computationally demanding. PhyDNet [19] leveraged physical constraints to better represent motion dynamics but was limited in modeling highly complex scenarios.

SimVP [17] significantly simplified the prediction model by employing spatial-temporal separable architectures, balancing performance with computational efficiency. Building upon this, SimVP-meta [58] extended SimVP by integrating recurrent, convolutional, and transformer-based architectures into a unified meta-model framework, greatly advancing the field of video prediction. Following this work, the autoregressive-based [50] model has further enriched the literature with promising directions.

Inspired by these developments, our paper introduces, for the first time, a novel Titans-based [2] architecture—*VideoTitans*—specifically designed to enhance both long-term and short-term memory capabilities, thereby addressing critical challenges in video prediction tasks.

3 Preliminaries

Memory and Sequence Modeling. Sequential modeling tasks, such as video prediction, typically involve handling long-term temporal dependencies. Recurrent neural networks (RNNs) encode these dependencies into a compressed hidden state but often lose essential information over longer sequences. On the other hand, Transformers explicitly model dependencies using attention, but this comes with quadratic complexity, limiting their applicability to very long sequences such as videos.

Neural Memory and Adaptive Forgetting. Recent architectures introduce explicit neural memory modules that dynamically store historical context beyond the immediate attention window. These modules update memory state $\mathbf{M}_t$ through recursive formulations such as:

$$\mathbf{M}_t = f(\mathbf{M}_{t-1}, \mathbf{x}_t), \tag{1}$$

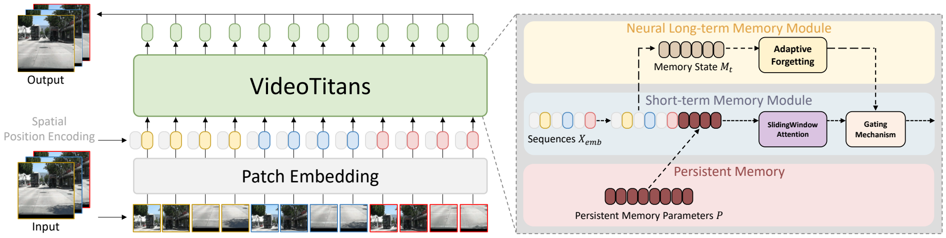


Figure 2: Overview of the VideoTitans framework, which integrates neural long-term memory, sliding window attention, and persistent memory to enhance video prediction. The model dynamically adapts to both short-term and long-term dependencies through a gradient-based surprise mechanism. The final prediction is obtained by combining short-term attention and memory states via a gating mechanism.

where f represents the memory update function that selectively encodes relevant information while employing adaptive forgetting mechanisms to discard redundant details. Such adaptive mechanisms are crucial to managing long-term dependencies without memory overflow.

Titans Framework. Titans proposes a neural memory module specifically designed to dynamically learn, memorize, and retrieve crucial information. Titans define a gradient-based surprise mechanism to determine the importance of events:

$$\mathbf{M}_t = \mathbf{M}_{t-1} + \mathbf{S}_t, \quad \mathbf{S}_t = \eta_t \mathbf{S}_{t-1} - \theta_t \nabla \ell(\mathbf{M}_{t-1}; \mathbf{x}_t), \tag{2}$$

where $\mathbf{S}_t$ captures both historical (past) and immediate (momentary) surprise. This allows Titans to dynamically adapt and selectively encode important events, balancing short-term attention and long-term memorization effectively.

Inspired by this approach, we introduce *VideoTitans*, adapting the Titans memory framework to video forecasting tasks, allowing efficient modeling of both local and global temporal dependencies inherent in video data.

4 Methodology

4.1 Problem Definition.

Given an input video sequence $\mathbf{X} \in \mathbb{R}^{B \times T \times C \times H \times W}$ consisting of T observed frames, the goal of video forecasting is to accurately predict subsequent future video frames $\mathbf{Y} \in \mathbb{R}^{B \times \hat{T} \times C \times H \times W}$. Here, B represents the batch size, T denotes the number of observed frames, $\hat{T}$ is the number of future frames to predict, C corresponds to the number of channels, and H, W indicate the spatial dimensions (height and width) of each frame. Formally, the task can be defined as learning a function:

$$\hat{\mathbf{Y}} = \mathcal{F}(\mathbf{X}; \theta), \tag{3}$$

where the model $\mathcal{F}$, parameterized by θ, aims to learn complex spatio-temporal dependencies from past video frames and leverage them to generate precise, high-fidelity predictions for future frames. The challenge is that video data inherently contains both short-term dynamics (local correlations between consecutive frames) and long-term dependencies (slowly evolving or periodic patterns across multiple frames), making the accurate modeling of both short-term and long-term temporal relationships critical for reliable predictions.

4.2 VideoTitans

Input Embedding and Positional Encoding. The input video sequence (B, T, C, H, W) is reshaped into $(B \times T, C, H, W)$. Each frame is embedded into spatial patches with positional encoding (PE):

$$(B \times T, C, H, W) \to (B \times T, \text{embed\_dim}, H/16, W/16).$$

We then permute the embeddings into a form suitable for temporal modeling: $(B \times T, \text{embed\_dim}, H/16, W/16) \to (B, H/16 \times W/16, T \times \text{embed\_dim})$.

	MovingMNIST				TrafficBJ				Human3.6			
Method	MSE (↓)	MAE (↓)	SSIM (↑)	FLOPs (G)	MSE (↓)	MAE (↓)	SSIM (↑)	FLOPs (G)	MSE (↓)	MAE (↓)	SSIM (↑)	FLOPs (G)
ConvLSTM [51]	29.7990	90.6396	0.9288	56.8	0.3358	15.3175	0.9836	20.74	125.5210	1566.7080	0.9813	347.0
E3D-LSTM [66]	38.5383	83.6387	0.9267	298.9	0.3427	14.9824	0.9842	98.19	143.3489	1442.4492	0.9803	542.0
MIM [67]	$\underline{22.5508}$	$\underline{69.9673}$	**0.9488**	179.2	0.3130	14.9387	0.9824	64.10	111.8432	1463.4142	0.9830	1051.0
PhyDNet [19]	28.1955	78.6397	0.9374	15.3	0.3622	15.5315	0.9828	5.60	125.7428	1614.7234	0.9804	**19.1**
PredRNN [64]	23.9667	72.8222	0.9460	116.0	0.3194	15.3077	0.9838	42.40	113.1855	1458.3422	0.9831	704.0
PredRNNv2 [68]	24.1136	73.7252	0.9450	116.6	0.3834	15.5528	0.9826	42.63	114.8799	1484.8729	0.9827	708.0
MAU [8]	26.8564	78.2186	0.9396	17.8	0.3268	15.2582	0.9834	6.02	127.3176	1577.0112	0.9812	105.0
TAU [57]	24.6029	71.9298	0.9454	16.0	$\underline{0.3108}$	$\underline{14.9341}$	$\underline{0.9849}$	2.49	113.3487	**1390.6997**	$\underline{0.9839}$	182.0
SimVP-IncepU [17]	32.1478	89.0498	0.9268	19.4	0.3282	15.4554	0.9835	3.61	115.8376	1511.4755	0.9822	197.0
SimVP-gSTA [58]	26.6926	77.1883	0.9402	16.5	0.3247	15.0290	0.9844	2.62	**108.0713**	1444.5731	0.9833	$\underline{74.6}$
SimVP-Swin [58]	29.6991	84.0507	0.9331	16.4	0.3127	15.0689	0.9847	2.56	133.2034	1599.7281	0.9799	188.0
SimVP-Uniformer [58]	30.3827	85.8719	0.9308	16.5	0.3268	15.1653	0.9844	2.71	116.3079	1497.6663	0.9824	211.0
SimVP-ViT [58]	35.1473	95.8649	0.9140	16.9	0.3171	15.1532	0.9841	2.80	136.3321	1603.5026	0.9796	239.0
SimVP-Poolformer [58]	31.7882	88.4830	0.9271	$\underline{14.1}$	0.3273	15.3947	0.9840	$\underline{2.06}$	118.4458	1484.1716	0.9827	156.0
VideoTitans	**21.3265**	**65.2124**	$\underline{0.9463}$	**13.33**	**0.3099**	**14.8220**	**0.9898**	**1.90**	$\underline{109.4821}$	$\underline{1401.3456}$	**0.9871**	128.52

Table 1: A performance comparison of VideoTitans with other approaches is conducted on three standard benchmark datasets for future frame prediction. VideoTitans consistently achieves competitive results across Moving MNIST, TrafficBJ, and Human 3.6, which differ in characteristics.

Neural Long-term Memory Module. We introduce an adaptive neural long-term memory module based on a gradient-based surprise mechanism. The memory state update rule at time t is:

$$\mathbf{M}_t = (1 - \alpha_t)\mathbf{M}_{t-1} + \mathbf{S}_t, \tag{4}$$

where the surprise score $\mathbf{S}_t$ is computed by:

$$\mathbf{S}_t = \eta_t \mathbf{S}_{t-1} - \theta_t \nabla \ell(\mathbf{M}_{t-1}; \mathbf{x}_t). \tag{5}$$

Here, parameters α_t, η_t, and θ_t control adaptive forgetting, surprise decay, and momentary surprise integration, respectively, enabling the selective memorization of crucial historical information.

Sliding Window Attention. To precisely model short-term temporal dependencies, sliding window attention is applied to embedded input sequences:

$$\mathbf{Y}_S = \text{SlidingWindowAttention}(\mathbf{X}_{emb}), \tag{6}$$

where $\mathbf{X}_{emb}$ denotes the reshaped spatial embeddings.

Persistent Memory. To encode task-specific and context-independent information, we incorporate persistent memory parameters $\mathbf{P}$. These parameters are concatenated to the embedded input as follows:

$$\mathbf{X}_{new} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{N_p}] || || \mathbf{X}_{emb}. \tag{7}$$

Decoding and Frame Reconstruction. The final prediction is obtained by decoding the combined representations of short-term attention and neural long-term memory through a gating mechanism:

$$\hat{\mathbf{Y}} = \text{Decoder}\left(\sigma(\mathbf{Y}_S \otimes \mathbf{M}_t)\right). \tag{8}$$

The decoded output $\hat{\mathbf{Y}}$ is reshaped back to the original video dimensions $(B, \hat{T}, C, H, W)$.

5 Experiments

In this section, we present extensive evaluations of our proposed VideoTitans architecture on widely adopted benchmarks for future frame prediction. Additionally, we analyze the effectiveness of each component of VideoTitans through comprehensive ablation studies.

Dataset	N_{train}	N_{test}	(C, H, W)	T	T'
Moving MNIST [53]	10,000	10,000	(1, 64, 64)	10	10
TrafficBJ [74]	20,461	500	(2, 32, 32)	4	4
Human3.6 [26]	73,404	8,582	(3, 128, 128)	4	4
Weatherbench [45]	2,167	706	(1/2, 32, 64)	12	12

Table 2: Summary of datasets used. N_{train}/N_{test} are sample counts. (C, H, W): input shape. T/T': input/predicted frames.

Datasets We evaluate VideoTitans on four widely-used datasets for future frame prediction, summarized in Table 2: Moving MNIST (MMNIST) [53], TrafficBJ [74], Human 3.6 [26], and WeatherBench [45]. Moving MNIST consists of synthetically generated video sequences depicting two digits moving randomly within a constrained grid, making it ideal for evaluating models on simple nonlinear temporal dynamics. TrafficBJ contains real-world traffic flow data collected in Beijing, capturing complex urban spatio-temporal dynamics. Human 3.6M comprises high-resolution motion capture data of human activities, challenging the model to accurately capture subtle and intricate human motions. WeatherBench includes global meteorological variables such as temperature (t2m), wind fields (uv10), and total cloud cover (tcc), testing the model's capacity to handle complex global-scale spatio-temporal interactions.

Evaluation Metric We use widely adopted metrics to assess prediction quality and evaluate four different datasets. Specifically, MMNIST, TrafficBJ, and Human3.6 are evaluated using the Mean Square Error (MSE), Mean Absolute Error (MAE), and Structural Similarity Index Measure (SSIM). The weatherbench is evaluated with MSE, MAE, and Root Mean Square Error (RMSE).

Implementation Details Following [58], we optimize VideoTitans using the Adam optimizer and train with the Mean Squared Error (MSE) loss. We set the batch size to 8 for all experiments. The learning rate is adaptively adjusted using the ReduceLROnPlateau scheduler with patience of 10 epochs. Initial learning rates are selected from the set $\{10^{-2}, 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$, and the best-performing value is used for each dataset. The total number of training epochs varies depending on the dataset complexity and size. All experiments are implemented using PyTorch and conducted on 8 NVIDIA A100 GPUs. More details can be found in the supplementary material and the code.

5.1 Quantitative results

Moving MNIST The Moving MNIST dataset is characterized by simple yet highly nonlinear dynamics involving continuous movements and interactions of digit shapes. Our VideoTitans effectively captures these nonlinear temporal dynamics, demonstrating state-of-the-art performance in long-term forecasting by adaptively preventing error accumulation. This indicates the robustness of our model in handling synthetic nonlinear trajectories. Detailed quantitative results are provided in the left column of Table 1.

TrafficBJ The TrafficBJ dataset contains real-world urban traffic sequences exhibiting complex spatio-temporal patterns and periodic fluctuations. VideoTitans successfully captures both short-term local variations and long-term global trends, achieving superior forecasting accuracy compared to transformer-based and recurrent models. This highlights the applicability of VideoTitans to dynamic urban traffic scenarios. Comprehensive performance comparisons are provided in the middle column of Table 1.

Human 3.6M The Human 3.6M dataset includes sequences of articulated human motions captured under controlled conditions, demanding precise modeling of intricate spatio-temporal interactions. VideoTitans demonstrates robust performance by effectively modeling subtle short-term movements while maintaining long-term coherence in human motion sequences. These results underscore its strength in detailed human motion prediction. Full comparisons are available in the right column of Table 1. We further include results on the Caltech-Pedestrian [14] in the Supplementary Material.

WeatherBench The WeatherBench dataset involves forecasting long-range meteorological variables such as temperature (t2m), wind fields (uv10), and total cloud cover (tcc), characterized by complex spatio-temporal dependencies and nonlinear interactions on a global scale. VideoTitans demonstrates robust performance on these variables, as detailed in Table 3, achieving competitive performance compared to previous methods with lower FLOPs. This performance suggests that VideoTitans effectively captures large-scale dependencies inherent in meteorological data, making the model potentially suitable for weather forecasting tasks. Nonetheless, since VideoTitans does not explicitly model

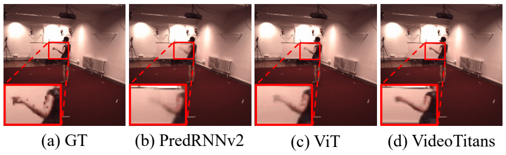

(a) GT (b) PredRNNv2 (c) ViT (d) VideoTitans

Figure 3: Qualitative comparison of predicted future frames on the Human3.6 dataset. Our model makes more accurate predictions, particularly noticeable in the person's arm.

	t2m				uv10				tcc			
Method	MSE (↓)	MAE (↓)	RMSE (↓)	FLOPs (G)	MSE (↓)	MAE (↓)	RMSE (↓)	FLOPs (G)	MSE (↓)	MAE (↓)	RMSE (↓)	FLOPs (G)
ConvLSTM	1.9866	0.8558	1.4095	136	2.4170	1.0180	1.5547	136	0.0722	0.1746	0.2688	136
E3D-LSTM	1.5921	0.8059	1.2618	169	2.4111	1.0341	1.5528	171	0.0573	0.1529	0.2394	169
MIM	2.3940	0.9837	1.5473	109	3.3708	1.2363	1.8360	109	0.0798	0.1836	0.2825	109
PhyDNet	290.9133	8.8492	17.0496	36.8	16.7869	2.9188	4.0972	36.8	0.0997	0.2261	0.3157	36.8
PredRNN	1.7250	0.7987	1.3134	278	2.6378	1.0804	1.6241	279	0.0789	0.1803	0.2810	278
PredRNN++	1.4575	0.7676	1.2073	413	2.5476	1.0548	1.5961	414	0.0797	0.1954	0.2824	413
PredRNNv2	1.7826	0.8074	1.3351	279	2.8591	1.1303	1.6909	280	0.0828	0.1874	0.2878	279
MAU	1.2413	0.6977	1.1141	39.6	2.1530	0.9594	1.4673	39.6	0.0707	0.1715	0.2660	39.6
TAU	1.3611	0.7056	1.1667	6.70	1.7051	0.8509	1.3058	6.70	0.0661	0.1653	0.2570	6.70
SimVP-IncepU	1.7897	0.8015	1.3378	8.03	1.9993	0.9510	1.4140	8.04	0.0754	0.1760	0.2747	8.03
SimVP-gSTA	**1.1523**	**0.6524**	**1.0735**	7.01	1.7272	0.8812	1.3142	7.02	**0.0469**	**0.1474**	**0.2166**	7.01
SimVP-Swin	1.2235	0.6665	1.1061	6.88	1.5709	0.8168	1.2533	6.89	0.0589	0.1567	0.2426	6.88
SimVP-Uniformer	1.1948	0.6697	1.0930	7.45	1.4781	0.8059	1.2158	7.46	0.0561	0.1553	0.2368	7.45
SimVP-ViT	1.2954	0.6873	1.1382	7.99	1.6893	0.8512	1.2997	8.0	0.0615	0.1596	0.2480	7.99
SimVP-Poolformer	1.2525	0.6711	1.1191	5.61	1.6678	0.8427	1.2914	5.62	0.0562	0.1530	0.2371	5.61
VideoTitans	1.1852	0.6636	1.1158	**4.92**	**1.4056**	**0.7984**	**1.1850**	**4.92**	0.0554	0.1522	0.2353	**4.92**

Table 3: Performance comparison of VideoTitans and state-of-the-art methods on the WeatherBench dataset for predicting temperature (`t2m`), wind velocity (`uv10`), and total cloud cover (`tcc`). VideoTitans demonstrates competitive predictive accuracy across all variables, effectively capturing complex global spatio-temporal patterns inherent in weather forecasting tasks.

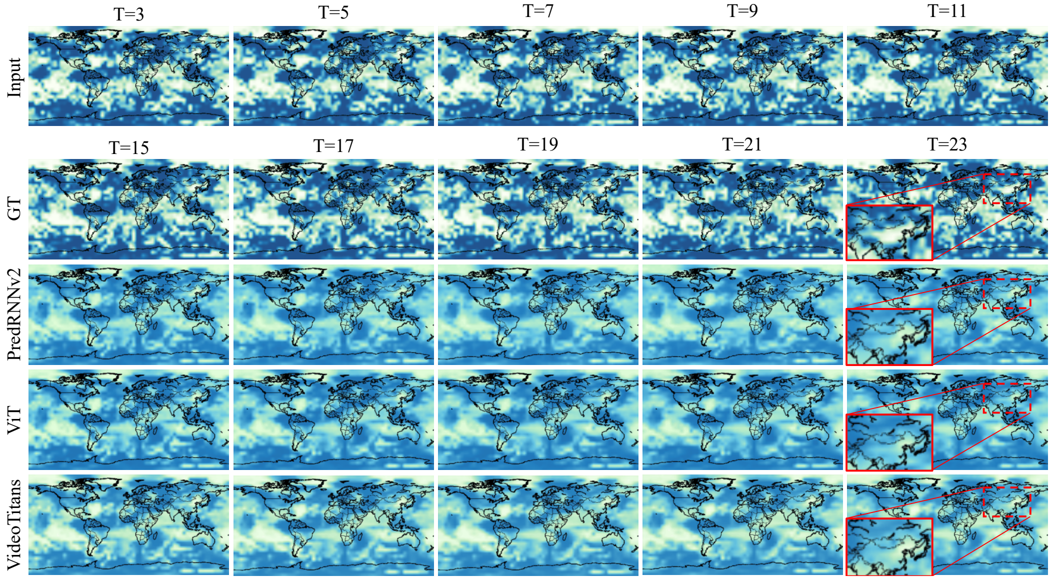


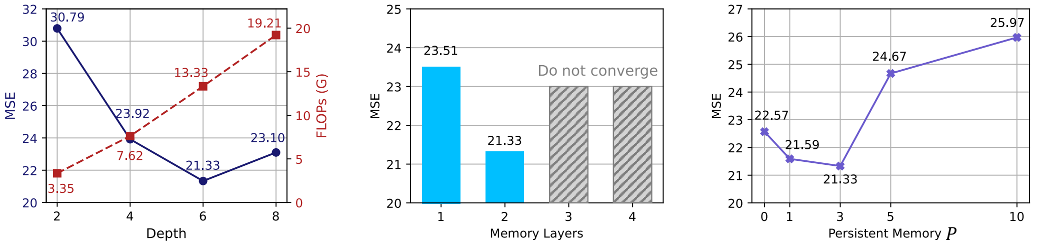
Figure 4: Qualitative comparison of predicted total cloud cover (`tcc`) frames on the WeatherBench dataset. Predictions from VideoTitans are compared against PredRNNv2 (recurrent-based) and ViT (transformer-based). Red boxes highlight regions where VideoTitans better preserves cloud patterns compared to other methods, indicating its capability to effectively model global spatio-temporal interactions in meteorological data.

the continuous fine-grained dynamics common in atmospheric processes, additional architectural improvements could further enhance its predictive capabilities for subtle climate interactions.

5.2 Qualitative results

Human3.6 Figure 3 presents qualitative comparisons between VideoTitans and baseline methods on the Human 3.6 dataset. The superiority of VideoTitans is clearly evident, as it generates sharper and more accurate predictions for subtle and articulated human movements compared to recurrent (PredRNNv2) and transformer-based (ViT) methods. This highlights VideoTitans' capability to effectively capture detailed motion dynamics and maintain prediction quality.

WeatherBench We evaluate VideoTitans on the WeatherBench dataset using the standard protocol from OpenSTL, predicting weather variables at 1-hour intervals up to 12 hours into the future. The qualitative results in Figure 4 visualize the predicted total cloud cover (`tcc`) at the reduced resolution following the OpenSTL standard experimental setup. Since our primary objective is general video



(a) Depth vs. MSE & FLOPs (b) Memory Layers vs. MSE (c) Persistent Memory vs. MSE

Figure 5: Ablation analysis of VideoTitans on the Moving MNIST dataset. (a) illustrates how model memory depth affects both FLOPs and MSE, showing that moderate depth offers the best trade-off between efficiency and accuracy. (b) presents the effect of the number of neural memory layers, where performance peaks at two layers, while deeper configurations encounter training instability. (c) shows the impact of persistent memory parameters $\mathbf{P}$, indicating that selecting the optimal value is key to achieving the best performance.

prediction, we opt for this protocol; however, future work will extend our evaluation to the higher-resolution WeatherBench2 [46] dataset at its full spatial resolution (0.25-degree). This will allow us to better examine VideoTitans' capability in capturing finer-grained meteorological dynamics. Additional qualitative results across datasets are provided in the Supplementary Material.

5.3 Ablation Study

Effect of Memory Depth We analyze the effect of varying the depth (number of attention blocks) of the VideoTitans on the Moving MNIST dataset in figure 5 (a). Increasing the depth initially improves prediction accuracy, reaching the lowest MSE at a depth of 6. However, beyond this depth, we observe diminishing returns, as the MSE slightly increases at depth 8, suggesting a trade-off between computational complexity and prediction accuracy. These results indicate that a depth of 6 achieves the optimal balance between model complexity (measured in FLOPs) and forecasting performance.

Hyperparameter	Value
Neural Memory Depth	2
Neural Demory Dim	512
Head	4
Momentum Order	1
Max Gradient Norm	1.0
Persistent Mem Tokens	4
Chunk Size	256
Segment Len	256
Long-term Mem Tokens	16

Table 4: Detailed hyperparameter configuration used for VideoTitans training.

Design of Neural memory We investigate the effect of neural memory depth on the prediction performance of VideoTitans by varying the number of layers within the memory module from 1 to 4 in figure 5 (b). The model achieves the best performance (MSE: 21.3265) with a memory depth of 2. Despite extensive experimentation—including careful tuning of hyperparameters, gradient norm clipping, learning rate adjustments, and various initialization strategies—deeper memory modules (3 and 4 layers) consistently faced severe training instabilities and failed to converge. This highlights a critical trade-off between memory depth and training stability, indicating that very deep neural memory structures may require extensive and precise hyperparameter tuning or architectural modifications to ensure convergence and maintain stability.

Influence of Persistent Memory Parameter ($\mathbf{P}$) Figure 5 (c) examines how varying the Persistent Memory Parameter ($\mathbf{P}$) influences the prediction performance. The optimal performance (MSE: 21.3265) is achieved at $\mathbf{P} = 3$. Smaller values ($\mathbf{P} = 0$ or 1) and larger values ($\mathbf{P} = 5$ or 10) degrade performance, suggesting that a moderate value of $\mathbf{P}$ balances model complexity and memory capacity for the best forecasting outcomes.

Effect of Persistent Memory Table 5 evaluates the impact of including Persistent Memory in VideoTitans on the Moving MNIST dataset. The presence of Persistent Memory significantly reduces MSE from 23.5125 to 21.3265, indicating that Persistent Memory effectively helps the model retain critical temporal context, leading to improved prediction accuracy.

Persistent Memory	MSE
With	**21.3**
Without	23.5

Table 5: Persistent Memory ablation on Moving MNIST.

Attention Type	MSE
Sliding Window	**21.3**
Global Attention	24.4
No Attention	30.7

Table 6: Attention mechanism ablation on Moving MNIST.

Method	MSE
Memory as a Context	66.5201
Memory as a Gate	**21.3265**
Memory as a Layer	24.4822

Table 7: Comparison of VideoTitans performance with different memory integration strategies.

Impact of Attention Mechanism Table 6 compares the effect of different attention strategies—Sliding Window Attention, Global Attention, and no attention. Sliding Window Attention achieves the lowest MSE (21.3265), demonstrating its superior capability to effectively focus on local temporal dynamics compared to Global Attention (MSE: 24.4822) or removing attention entirely (MSE: 30.79).

Memory Integration Strategies Table 7 compares the performance of VideoTitans on MMNIST using three different memory integration strategies of Titans [2]: Memory as a Context (MAC), Memory as a Gate (MAG), and Memory as a Layer (MAL). MAG achieves significantly better performance (lowest MSE), clearly outperforming both MAC and MAL. This demonstrates that incorporating memory using a gating mechanism is particularly effective at capturing and selectively integrating critical historical information, leading to superior video prediction results.

6 Limitation and Future Work

In this work, we extensively evaluate VideoTitans on the video prediction task. Although our proposed model demonstrates strong generalization across multiple diverse datasets, further studies should investigate the effectiveness of our method in broader vision tasks such as action recognition, video segmentation, and anomaly detection.

While diffusion-based models have recently shown strong performance in video generation, our focus is on video prediction—forecasting future frames based on observed inputs, rather than generating from noise or prompts. Diffusion models involve iterative denoising processes with significant computational cost, prioritizing high-quality synthesis. In contrast, video prediction demands temporal consistency and real-time efficiency. Therefore, we compare against models specifically designed for future frame prediction. Still, leveraging the high fidelity of diffusion models alongside the real-time efficiency of predictive models offers a promising direction for long-range video forecasting.

Finally, we observe that despite the compelling strengths of the Titans concept, the choice of hyperparameters significantly impacts performance stability. Particularly, the neural memory module depth and gradient constraints are crucial; exceeding two layers frequently causes numerical instability during training. One important contribution of this work is the identification and documentation of these critical hyperparameters (Table 4), enabling stable and reproducible implementations of VideoTitans in future research.

7 Conclusion

In this work, we propose VideoTitans, a neural architecture designed for spatio-temporal video prediction, effectively capturing both local motion dynamics and long-term dependencies. By integrating three key components—Short-Term Memory with attention-based processing for recent frames, Long-Term Memory for selectively encoding and retrieving historical contexts, and Persistent Memory for task-specific knowledge—VideoTitans efficiently models complex video sequences while maintaining computational scalability. Extensive experimental evaluations demonstrate that VideoTitans consistently outperforms CNN-based, Transformer-based, and recurrent models, achieving superior predictive accuracy while significantly improving efficiency for long-term forecasting. These results underscore the effectiveness of VideoTitans as a robust and scalable solution for video-based predictive modeling, paving the way for advancements in real-world applications such as autonomous systems, surveillance, and robotics.

Acknowledgement. This work was partially supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2024-00338439) and Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2025-25441838, Development of a human foundation model for human-centric universal artificial intelligence and training of personnel).

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[2] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

[3] Aydar Bulatov, Yuri Kuratov, Yermek Kapushev, and Mikhail Burtsev. Beyond attention: breaking the limits of transformer context length with recurrent memory. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[4] Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint arXiv:2006.11527*, 2020.

[5] Wonmin Byeon, Thomas M Breuel, Federico Raue, and Marcus Liwicki. Scene labeling with lstm recurrent neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[6] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[7] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[8] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and beyond. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2021.

[9] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[10] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS Workshop on Deep Learning*, 2014.

[11] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[12] Hanming Deng, Yang Hua, Tao Song, Zongpu Zhang, Zhengui Xue, Ruhui Ma, Neil Robertson, and Haibing Guan. Object guided external memory network for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[13] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017.

[14] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] Duarte Duque, Henrique Santos, and Paulo Cortez. Prediction of abnormal behaviors for intelligent video surveillance systems. In *2007 IEEE Symposium on Computational Intelligence and Data Mining*, pages 362–367. IEEE, 2007.

[16] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[17] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang Bernie Wang. Prediff: Precipitation nowcasting with latent diffusion models. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023.

[19] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] Xudong Guo, Xun Guo, and Yan Lu. Ssan: Separable self-attention network for video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(1):87–110, 2022.

[22] Simon Hecker, Dengxin Dai, and Luc Van Gool. Failure prediction for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1792–1799. IEEE, 2018.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[24] Anthony Hu, Fergal Cotter, Nikhil Mohan, Corina Gurau, and Alex Kendall. Probabilistic future prediction for video scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[25] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

[26] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1325–1339, 2013.

[27] Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. Grid long short-term memory. *arXiv preprint arXiv:1507.01526*, 2015.

[28] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[29] Jasmeen Kaur and Sukhendu Das. Future frame prediction of a video sequence. *arXiv preprint arXiv:2009.01689*, 2020.

[30] Dongkyu Lee, Chandana Satya Prakash, Jack FitzGerald, and Jens Lehmann. Matter: Memory-augmented transformer using heterogeneous knowledge sources. In *Findings of the Association for Computational Linguistics (ACL)*, 2024.

[31] Sunghoon Lim, Sun Jun Kim, YoungJae Park, and Nahyun Kwon. A deep learning-based time series model with missing value handling techniques to predict various types of liquid cargo traffic. *Expert Systems with Applications*, 184:115532, 2021.

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[33] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[34] Ioannis Mademlis, Anastasios Tefas, Nikos Nikolaidis, and Ioannis Pitas. Summarization of human activity videos via low-rank approximation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[35] George Mandler. The structure of value: Accounting for taste. In *Affect and cognition*, pages 3–36. Psychology Press, 2014.

[36] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *International Speech Communication Association (Interspeech)*, 2010.

[37] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[38] Xuran Pan, Tianzhu Ye, Zhuofan Xia, Shiji Song, and Gao Huang. Slide-transformer: Hierarchical vision transformer with local self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[39] Bo Pang, Kaiwen Zha, Hanwen Cao, Chen Shi, and Cewu Lu. Deep rnn framework for visual sequential applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] Jin-Hwi Park, Young-Jae Park, Junoh Lee, and Hae-Gon Jeon. Deviancenet: Learning to predict deviance from a large-scale geo-tagged dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[41] Jin-Hwi Park, Young-Jae Park, Ilyung Cheong, Junoh Lee, Young Eun Huh, and Hae-Gon Jeon. What makes deviant places? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(11):7405–7420, 2024.

[42] Young-Jae Park, Minseok Seo, Doyi Kim, Hyeri Kim, Sanghoon Choi, Beomkyu Choi, Jeongwon Ryu, Sohee Son, Hae-Gon Jeon, and Yeji Choi. Long-term typhoon trajectory prediction: A physics-conditioned approach without reanalysis data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[43] Young-Jae Park, Doyi Kim, Minseok Seo, Hae-Gon Jeon, and Yeji Choi. Data-driven precipitation nowcasting using satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[44] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

[45] Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

[46] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024.

[47] Mats L Richter, Julius Schöning, Anna Wiedenroth, and Ulf Krumnack. Should you go deeper? optimizing convolutional neural network architectures without training. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021.

[48] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[49] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*. IEEE, 2004.

[50] Minseok Seo, Hakjin Lee, Doyi Kim, and Junghoon Seo. Implicit stacked autoregressive model for video prediction. *arXiv preprint arXiv:2303.07849*, 2023.

[51] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2015.

[52] Gaurav Shrivastava and Abhinav Shrivastava. Video prediction by modeling videos as continuous multi-dimensional processes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[53] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

[54] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[55] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011.

[56] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2014.

[57] Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[58] Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z Li. Openstl: A comprehensive benchmark of spatio-temporal predictive learning. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023.

[59] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[60] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[61] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2017.

[63] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[64] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2017.

[65] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[66] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[67] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[68] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, S Yu Philip, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(2):2208–2225, 2022.

[69] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[70] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[71] Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. Motionrnn: A flexible model for video prediction with spacetime-varying motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[72] Y Wu, Y Zhao, B Hu, P Minervini, P Stenetorp, and S Riedel. An efficient memory-augmented transformer for knowledge-intensive nlp tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[73] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, et al. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[74] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[75] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[76] Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2021.

[77] Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. Long short-term memory over recursive structures. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

A Additional Dataset

A.1 Caltech-Pedestrian Dataset

The Caltech-Pedestrian [14] dataset presents challenging real-world urban scenarios involving diverse pedestrian movements, occlusions, and complex dynamics. It is evaluated using metrics such as Mean Square Error (MSE), Structural Similarity Index Measure (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS)[75], testing the robustness and accuracy of predictive models. As shown in Table8, VideoTitans demonstrates competitive performance across these metrics, effectively capturing intricate pedestrian trajectories and spatial relationships. This highlights its practical applicability in dynamic environments, combining predictive accuracy with computational efficiency.

Method	**MSE** (↓)	**SSIM** (↑)	**PSNR** (↑)	**LPIPS** (↓)	**FLOPs(G)** (↓)
ConvLSTM	139.6588	0.9345	27.4644	0.0857	595.0
E3D-LSTM	199.1374	0.9047	25.4612	0.1261	1004.0
MIM	**123.9034**	0.9410	28.1148	0.0642	1858.0
PhyDNet	310.6844	0.8615	23.2723	0.3218	40.4
PredRNN	129.3306	0.9375	27.8074	0.0745	1216.0
PredRNNv2	143.4366	0.9334	27.1864	0.0895	1223.0
MAU	177.4630	0.9174	26.1504	0.0969	172.0
TAU	128.9193	**0.9458**	27.8465	0.0551	80.0
SimVP-IncepU	160.2191	0.9338	26.8093	0.0675	60.6
SimVP-gSTA	127.7992	0.9456	27.9191	0.0577	96.3
SimVP-Swin	155.2470	0.9300	27.2542	0.0811	95.2
SimVP-Uniformer	135.9496	0.9393	27.6607	0.0687	104.0
SimVP-ViT	146.3816	0.9380	27.4267	0.0666	155.0
SimVP-Poolformer	153.3675	0.9334	27.3807	0.0700	79.8
VideoTitans	130.4290	0.9448	**28.8861**	**0.0512**	**9.9**

Table 8: Performance comparison on Caltech Pedestrian dataset.

A.2 KTH Dataset

The KTH [49] dataset is characterized by structured human actions and stable motion patterns which test a model's ability to capture temporal dynamics and spatial coherence in controlled settings. As shown in Table 9 VideoTitans achieves state-of-the-art performance across all evaluation metrics including MSE, Mean Absolute Error (MAE), PSNR, and SSIM. It combines high predictive accuracy with low computational complexity which confirms its practical effectiveness and shows its strength in modeling regular motion sequences with precision and efficiency.

Method	**MSE** (↓)	**MAE** (↓)	**PSNR** (↑)	**SSIM** (↑)	**FLOPs(G)** (↓)
ConvLSTM	47.65	445.5	26.99	0.8977	1368.0
E3D-LSTM	136.40	892.7	21.78	0.8153	217.0
MIM	40.73	380.8	27.78	0.9025	1099.0
PhyDNet	91.12	765.6	23.41	0.8322	93.6
PredRNN	41.07	380.6	27.95	0.9097	2800.0
PredRNNv2	39.57	368.8	28.01	0.9099	2815.0
MAU	51.02	471.2	26.73	0.8945	399.0
TAU	45.32	421.7	27.10	0.9086	73.8
SimVP-IncepU	41.11	397.1	27.46	0.9065	62.8
SimVP-gSTA	45.02	417.8	27.04	0.9049	76.8
SimVP-Swin	45.72	405.7	27.01	0.9039	75.9
SimVP-Uniformer	44.71	404.6	27.16	0.9058	78.3
SimVP-ViT	56.57	459.3	26.19	0.8947	112.0
SimVP-Poolformer	45.55	400.9	27.22	0.9065	63.6
VideoTitans	**34.27**	**320.8**	**29.31**	**0.9197**	**50.9**

Table 9: Performance comparison on KTH dataset.

B Qualitative Results

Figure 6 shows qualitative comparisons between VideoTitans, recurrent (PredRNNv2), and transformer-based (ViT) methods on the Moving MNIST dataset. Due to the dataset's relatively simple dynamics, all models perform similarly well, making it challenging to visually distinguish significant differences among predictions. Empirically, we observe that differences primarily lie in convergence speed rather than final performance, as extending training epochs tends to improve accuracy for all models. Nevertheless, VideoTitans consistently provides slightly more stable and accurate results. We also present qualitative results for t2m and uv10 variables from the WeatherBench dataset. Further qualitative results of VideoTitans are also available as GIF animations for better visualization.

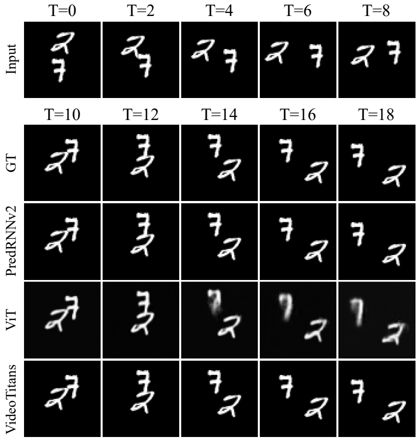


Figure 6: Qualitative comparison of predicted frames on the Moving MNIST dataset.

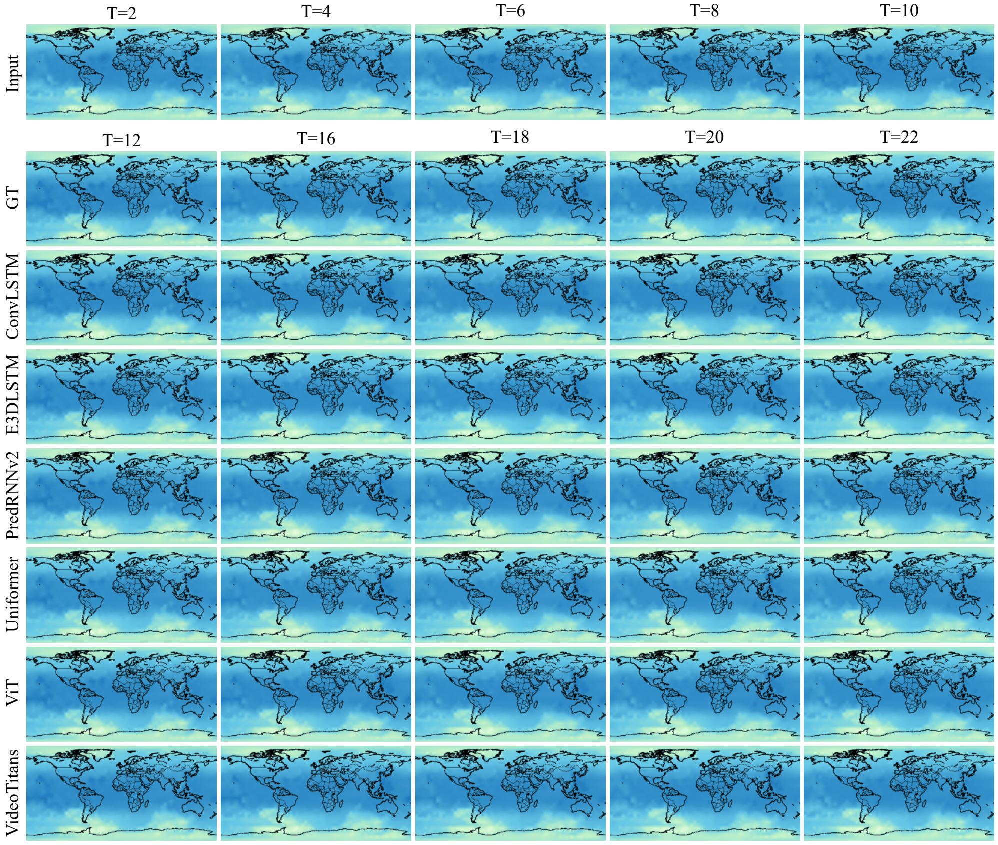


Figure 7: A qualitative comparison of predicted 2m temperature (`t2m`) frames on the WeatherBench dataset, comparing VideoTitans' predictions with those of other video prediction models.

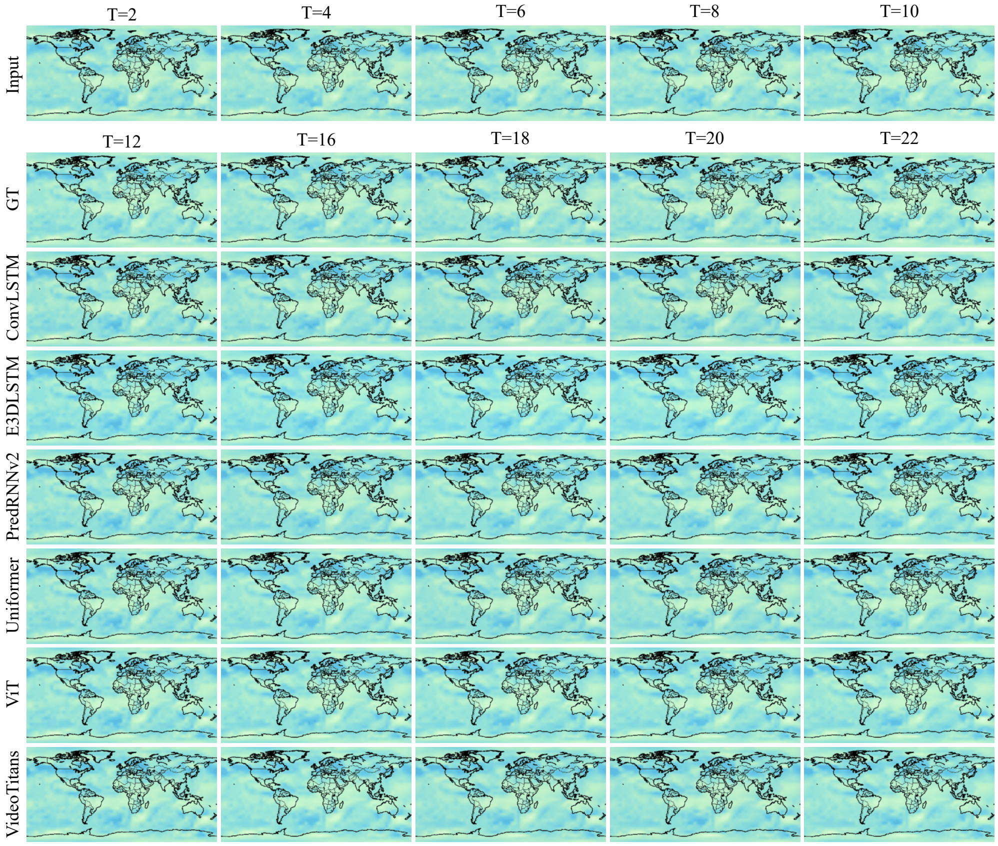


Figure 8: A qualitative comparison of predicted wind field (uv10) frames on the WeatherBench dataset, where VideoTitans' predictions are compared with those of other video prediction models.

C Implementation Details

C.1 Model Architecture

The architecture of VideoTitans consists of three main components: an encoder for spatial feature extraction, a Titans-based temporal modeling module, and a decoder for frame reconstruction.

Encoder. The encoder captures spatial and low-level visual features from input frames. Given an input tensor of shape (B, T, C, H, W), each frame is independently processed by convolution-based patch embedding. Specifically, we employ a convolutional layer with kernel size 16×16 and stride 16, converting the input as:

$$(B \times T, C, H, W) \rightarrow (B \times T, \text{embed\_dim}, H/16, W/16).$$

Afterward, spatial positional encodings are added to preserve positional information. The tensor is then reshaped for temporal processing:

$$(B \times T, \text{embed\_dim}, H/16, W/16) \rightarrow (B, H/16 \times W/16, T \times \text{embed\_dim}).$$

Titans-based Temporal Modeling. The temporal modeling is based on the Titans architecture, utilizing neural long-term memory that adaptively updates weights via a gradient-based surprise metric, efficiently capturing essential temporal patterns. Key hyperparameters, such as memory depth, memory dimension, persistent memory tokens, and maximum gradient norm, are critical for stable training. In particular, setting the maximum gradient norm to 1.0 prevents training instabilities such as gradient explosions.

The Titans module processes embeddings in segments, employing sliding window attention to model both local and global temporal dependencies. Persistent memory tokens encode context-independent knowledge to enhance generalization across datasets.

Decoder. The decoder reconstructs predicted frames from the temporal features. Mirroring the encoder structure, it utilizes transpose convolutional layers (kernel size 16×16, stride 16) to restore spatial dimensions:

$$(B, H/16 \times W/16, T \times \text{embed\_dim}) \rightarrow (B \times T, C, H, W).$$

The decoded frames are reshaped to the original dimensions (B, T, C, H, W) for comparison with ground-truth.

C.2 Training Procedure

We implement VideoTitans in PyTorch, using the Adam optimizer and Mean Squared Error (MSE) loss function. Key training parameters are summarized below:

- **Optimizer:** Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$).
- **Learning Rate Scheduler:** ReduceLROnPlateau (patience=10 epochs), initial learning rate chosen from $\{10^{-2}, 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$.
- **Batch Size:** 8 for all experiments.
- **Training Epochs:** MMNIST (200 epochs), Caltech Pedestrian (100 epochs), Human3.6, TrafficBJ, WeatherBench (50 epochs each).

Additionally, we apply the Exponential Moving Average (EMA) with a decay of 0.995 during training to enhance model stability and generalization.

C.3 Hyperparameter Sensitivity

A key contribution of our study includes identifying sensitive hyperparameters essential for VideoTitans' stable training. Notably, removing gradient norm constraints (e.g., setting max gradient norm) caused training instabilities, and overly deep neural memory layers (depth > 2) frequently result in numerical instability. Careful hyperparameter tuning is thus essential for robust training and optimal performance.

C.4 Memory Integration Strategies

There are three types of memory integration strategies in Titans: Memory as a Gate (MAG), Memory as a Context (MAC), and Memory as a Layer (MAL). MAG uses a gating mechanism to dynamically combine short-term attention and long-term memory, allowing the model to integrate previous knowledge adaptively. MAC retrieves past information from memory and appends it to the input sequence before processing it with attention, enabling selective use of historical data. MAL incorporates memory as an independent processing layer before the attention, similar to traditional hybrid recurrent models. Among these approaches, MAG achieves the best performance by effectively balancing short-term precision with long-term recall, leading to its selection as the baseline model for our work.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The main contributions and scope of our work are explicitly described in the Introduction section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

Justification: The limitations of our work are addressed in the "Limitation and Future Work" section of the paper.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: This paper does not present any theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide both detailed descriptions and the source code to ensure the reproducibility of our results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The implementation details and code are provided to ensure reproducibility.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experimental settings are detailed in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Error bars or statistical significance tests are not reported due to the high computational cost associated with repeated runs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the computational resources used in our experiments in the Implementation Details section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work conforms to the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The proposed model can benefit applications in autonomous systems and forecasting, but we also acknowledge potential risks in misuse for surveillance.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [No]

Justification: We did not implement specific safeguards as our model is not designed for high-risk applications such as language generation or facial synthesis. The model is intended for general-purpose video prediction tasks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We properly cite all datasets and baseline models used in our experiments, and ensure that their licenses and terms of use are respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: We provide code for our proposed model.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

Answer: [NA]

Justification: This paper does not involve research with human subjects or crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [NA]

Justification: This study does not involve human subjects or any form of crowdsourced data collection.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [No]

Justification: LLMs were used only for writing and editing purposes, not for methodological components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.