DynaTokens: Teaching Dynamics to Camera-Controlled Video Models at Test Time

Ziqi Ma **Hongqiao Chen** **Georgia Gkioxari**
California Institute of Technology

Abstract

Video generation must account for two sources of motion, one induced by the observer's camera path and the other caused by scene dynamics. An ideal camera-controlled video model should account for both motions: let users move the camera while evolving the scene dynamics. While current models handle camera-induced motion well in static settings, they struggle for dynamic scenes: objects are static, move incorrectly, or degrade in generation quality. We introduce DynaTokens, a lightweight set of learnable scene-specific tokens that teach dynamics to an existing camera-controlled world model. Our method is motivated by a simple asymmetry between the two sources of motion: whereas camera motion affects the generated view globally, object dynamics are spatially localized. Through cross-attention, DynaTokens trains the learnable tokens from a few example trajectories for a scene while keeping the base model frozen, and enables dynamics under new query camera paths. DynaTokens achieves a better simultaneous dynamics-camera tradeoff on VBench2 and WorldScore evaluations than LoRA, block finetuning, and specialized trainable-layer baselines. Analyses of token attention, ablations, and motion temporality suggest that matching the trainable interface to the structure of the learning target is important for effective adaptation. Project website: https://glab-caltech.github.io/dynatokens/

1 Introduction

Current camera-controlled video models [23, 25, 2, 22, 30, 10] can generate navigable "worlds" from a image or text prompts, opening new possibilities for gaming, immersive media, and robotics. However, despite their ability to follow camera trajectories, these models struggle with dynamics [31, 18]: the generated worlds often looks static, generate incorrect motion, or degrade in visual quality when the scene should evolve.

Recent work has shown that test-time training can adapt video foundation models to support new capabilities, including long-horizon generation [7], scientific discovery [32], and reasoning [1]. However, enabling scene dynamics in camera-controlled video models remains challenging because current architectures inject camera information directly into the generative flow process – via PRoPE [15], Plücker ray embeddings [33], or action conditioning [23, 10] – entangling camera control with dynamics generation. As a result, supervision on dynamics can inadvertently degrade camera control, and naive fine-tuning methods such as LoRA [11] perform poorly. This observation motivates a new approach to model adaptation. We introduce DynaTokens, a test-time training method that selectively teaches scene dynamics while preserving the camera-control behavior.

DynaTokens builds on a simple principle: in video generation, camera control is global, influencing every patch across every frame, whereas scene dynamics can be spatially local, often confined to the objects that move or change. To exploit this distinction, DynaTokens introduces learnable tokens that steer the video latents toward localized dynamics without altering the global scene generation. By decoupling local dynamic control from global camera control, this design enables dynamic scene generation while preserving camera controllability.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

DynaTokens is lightweight: a few scene-specific tokens are added and learned while keeping the base model frozen. It can be trained from only a few samples (≈ 15) and generalize to unseen camera trajectories, producing videos under novel camera controls. Importantly, these few training trajectories need not be high quality: they can be short, contain background artifacts or motion discontinuity. Through test-time training of the tokens, DynaTokens can overcome these limitations, generating temporally continuous videos with prompt-aligned dynamics for new trajectories.

We show quantitatively and qualitatively that DynaTokens effectively enables dynamics without compromising camera control, surpassing current capabilities of state-of-the-art camera-controlled world models on VBench2 [35] dynamic-related categories and WorldScore [8]. Baseline methods such as LoRA, block finetuning, or Test-Time Training layers struggle. Beyond dynamic motion categories, we show that DynaTokens can also enable physics and stylistic changes.

We provide analysis on the DynaTokens design to understand the disentangling of dynamics with camera control. We also probe the temporality aspects of dynamic tokens to better understand how they elicit dynamic behavior of base models. Our analyses on test-time training of dynamic tokens provide deeper insight into architecture and representation of world models, paving the way for future development of large-scale world models. We also illuminate how to effectively adapt these models at test time to enable new capabilities. Additionally, DynaTokens offers a path toward video data generation. Existing efforts in collecting dynamic, camera-annotated data rely on synthetic environments, such as games [6]. DynaTokens, via efficient test-time training, opens up generation of camera-controlled videos in diverse settings.

Our main contributions are as follows:

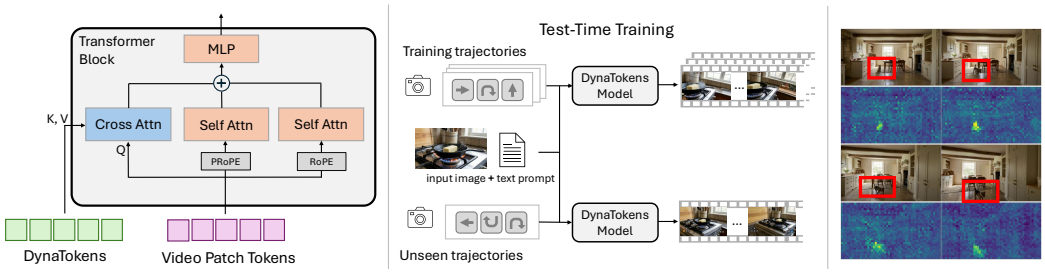
- We develop DynaTokens, which significantly improves dynamic scene generation for camera-controlled video models via test-time training, evaluated on VBench2 and WorldScore.
- DynaTokens is a new architecture design which decouples dynamics from camera control, enables much better performance than alternatives like LoRA and TTT layers.
- We identify a structural mismatch between existing adaptation methods and the underlying transformations they model: LoRA-type methods apply global parameter updates, whereas object dynamics is spatially localized. We provide analysis on how DynaTokens overcomes this limitation via a design that aligns with the learning target.

2 Related work

Camera conditioning in video models. Camera-controlled video models are flow-based video generation models that take in camera control, along with image and text, as conditioning signals. Camera conditioning can be achieved in 4 ways: noise warping [4, 22] which warps the noise based on camera movement, PRoPE [23, 31] which encodes camera frustum in positional embedding, action conditioning [23, 10] which encodes discrete camera actions, and plücker ray embedding [25, 22], usually channel-wise concatenated. We design DynaTokens to minimally affect camera conditioning and selectively "teach" dynamics.

Dynamics in camera-controlled video models. Camera-controlled video models [23, 25, 22, 10] struggle with dynamics due to a combination of 3D prior [2, 22] and the lack of dynamics in training data [17]. Methods that tackle dynamics can fall into two categories: (1) Improving existing general-purpose world models. These methods mostly focus on loosening 3D constraints in memory, such as [6, 9, 31], which addresses out-of-frame dynamics. (2) Alternative formulations of 4D-aware generation models, such as BulletTime [29] which allows both camera and time control, but is limited to short and small-range camera movements. The recent Atlas model [26] enables camera-controlled dynamics generation, but requires multiple input views.

Test-time training. Test-time training methods train foundation models at test time to surpass base model capabilities in challenging tasks, such as the ARC challenge [1], scientific discovery [32], and long video generation [7]. Architecturally, test-time training methods either use LoRA [1] or specialized architectures like Test-time Training (TTT) layers [34, 7]. We design DynaTokens, a new test-time training architecture for world models, to enable localized dynamic learning.



(a) DynaTokens design (b) TTT setting (c) Weight maps

Figure 1: DynaTokens enables camera-controlled video models to learn dynamics via test-time training. (a) DynaTokens introduces learnable tokens that are cross-attended by the video patch tokens in every transformer block. (b) For each scene, we train DynaTokens on a small set of camera trajectories (≈15), enabling inference on novel, unseen trajectories. (c) Visualization of the attention weights for a dynamic token, corroborating the localized dynamics hypothesis.

3 Method

DynaTokens introduces scene-specific tokens into a pretrained camera-controlled video generation model. During training, the base model is kept frozen, and only the tokens are updated, enabling correct dynamics when exploring different camera trajectories for the scene. We motivate the design of dynamic tokens, then explain our architecture, test-time training setup, and training objective.

3.1 Motivating example

Consider a simple setting where the world is a 2D coordinate frame, in which a filled unit circle, $P \subseteq \mathbb{R}^2$, is present, similar to the setting in [16]. The observer frame starts identical as the world frame. The goal of a video generative process is to recover the world at time t from the observer frame under both observer and scene motion.

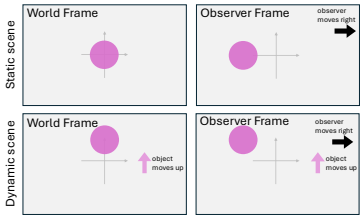


Figure 2: Motivating example.

Static scene, only observer moves: The observer moves with ψ_t which represents the "observer-to-world" mapping: $\psi_t(\mathbf{x}_{\text{obs}}) = \mathbf{x}_{\text{world}}$ for $\mathbf{x}_{\text{obs}} \in \mathbb{R}^2$, where $\mathbf{x}_{\text{world}}$ is a point's coordinate in the world frame and $\mathbf{x}_{\text{obs}}$ its coordinate in the observer frame. Assume $\psi_t : (x, y) \to (x + t, y)$ (observer moves right). If the scene is static, in the observer frame, this is equivalent to ψ_t^{-1} applied to object P: $P_t = \psi_t^{-1}(P) = \{(x + t, y) : (x, y) \in P\}$ – P moves left. In the observer frame, the scene is represented as $f(x, y, t) = \mathbb{1}_{\{(x+t)^2+y^2\leq 1\}}$ (top right of Fig 2).

Dynamic scene: Consider P to be a dynamic object, evolving with $\phi_t(P) \subseteq \mathbb{R}^2$. Unlike ψ_t which has global effect on the whole frame, ϕ_t is only applied to the local subset of object points, P. Assume ϕ_t represents uniform-speed "up" movement: $\phi_t : (x, y) \to (x, y+t)$ in the world frame. In the observer frame, this is equivalent to $P_t = \psi_t^{-1}(\phi_t(P)) = \{(x - t,\ y + t) : (x, y) \in P\}$ – P moves left and up. The world from the observation frame can be represented as $f_{\text{dyna}}(x, y, t) = \mathbb{1}_{\{(x+t)^2+(y-t)^2\leq 1\}}$, shown in Fig 2 bottom right.

Localized dynamics hypothesis. In a generative context, we care about the effect of object dynamics on the scene in the observer frame, f, which describes the "static world" and f_{dyna} which describes the "dynamic world". We define operator D to capture its effect, $f_{\text{dyna}} = Df$.

The motivating example above illustrates an important observation: at any given time, the effect of D is limited to P's movement. Thus, we can decompose D into a dynamics function and localized gating function: $(Df)(x, y, t) = m(x, y, t) \cdot (Gf)(x, y, t)$, where $m(x, y, t)$ is a binary gating function that localizes the dynamics and G applies the dynamics. In our example, $m(x, y, t) = \mathbb{1}_{\{(x+t)^2+(y-t)^2\leq 1 \vee (x+t)^2+y^2\leq 1\}}$, and G can be defined as $(Gf)(x, y, t) = f(x, y - t, t)$.

Similar to the motivating example of a circle in a plane, object dynamics in the real world are also localized: "a dog running" should affect only the dog region while preserving the rest of the scene. In other words, $D(f) = m \odot G(f)$. We build upon a base model that obeys camera control but largely keeps the scene static, *i.e.,* generates f, and turns it into f_{dyna} which captures both camera and scene

dynamics. We do this by recovering D, which we decompose into a localized gating function m which identifies where dynamics should occur, and a dynamics operator G that enables the motion.

3.2 Our approach: dynamic tokens

To achieve this, we design a lightweight add-on architecture for pretrained camera-controlled video models, enabling test-time learning of scene dynamics while preserving camera control. This presents two requirements: (1) the architecture needs to have enough representation capacity to "steer" the structure of the generated video latents while being lightweight; (2) the design should enable dynamics and camera control simultaneously.

We introduce DynaTokens, as illustrated in Fig. 1(a). Dynamic tokens are added to transformer blocks of a video generative model. The input video patch tokens interact with the learned dynamic tokens via cross attention. The output is added back to the model, along with the two paths of self-attention, with and without camera control (often instantiated via PRoPE [15]). We apply this to every double stream transformer block. The projection matrix that mixes multiple heads of cross-attention is initialized at zero to ensure minimal perturbation at initialization. We keep the base model frozen, and learn only the dynamic tokens at test time.

For each block, we introduce K dynamic tokens $\mathcal{D} = \{d_j \in \mathbb{R}^m\}_{j=1}^K$. If $\mathcal{H} = \{h_i \in \mathbb{R}^n\}_{i=1}^N$ are the video tokens input to the block, the output of cross-attention is represented as:

$$h_i' = h_i + \sum_j a_{ij} \cdot v(d_j), \qquad a_{ij} = \frac{\exp\big(q(h_i)^\top k(d_j)\big)}{\sum_l \exp\big(q(h_i)^\top k(d_l)\big)}. \tag{1}$$

The attention weights, a_{ij}, act as a gating function across the video frames, and enable application of dynamics only on relevant patches, essentially serving the role of m in our localized dynamics motivation above. Since the MLP after the attention block is applied channel-wise (*i.e.,* not mixing different patch tokens), this locality is preserved across transformer blocks. We train $K = 16$ dynamic tokens of dimension $m = 64$ in our experiments.

In Fig. 1(c), we visualize this gating mechanism in a real video by showing the attention weights applied on frames of a video generation with prompt: "The dog is on the left of the table, then it runs to the front of the table". The attention weights localize the dynamic regions, following the target object (dog) as it moves over time. We provide qualitative and quantitative analysis of attention map localization in §D in the Appendix.

3.3 Test-time training of dynamic tokens

Setup. Dynamic tokens are trained at test time to enable dynamic video generation. While large-scale data curation with rich scene and camera motion is difficult, DynaTokens' lightweight design enables learning only from a few videos (≈15), which are easier to curate. We design a curation process leveraging VLMs and video interpolation, detailed in §G of the Appendix, which yields videos with dynamics. These videos might contain artifacts, and we only curate them for a few camera trajectories. Trained on these few trajectories, DynaTokens enables correct dynamics on new camera trajectories. Our setup is illustrated in Fig. 1(b).

However, sample curation alone is insufficient: fine-tuning or LoRA-adapting a pretrained camera-controlled video model on the same data yields poor results, as we show in our experiments. In contrast, DynaTokens learns to generate the right dynamics under novel, unseen camera trajectories.

Training objective. We build on pretrained camera-controlled video models. As a representative architecture, we consider autoregressive video diffusion models, which generate videos chunk-by-chunk through diffusion. We instantiate our method on HY-WorldPlay [23], a state-of-the-art camera-controlled video model.

Training video models is challenging: even with discretized camera actions, a single scene can admit over 10^{19} possible trajectories over a 5-second clip. From only a handful of training trajectories (≈15) we aim to teach the add-on dynamic tokens to capture scene motion and generalize to unseen camera paths. To reduce exposure bias, where the model conditions on its own previous-chunk predictions that drift from the training setting, we leverage a diffusion-forcing-type [5] objective,

similar to [23, 22]:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{l,\,t}\left[\left\| v_\theta\big(x_l^{(t)},\, \tilde{x}_{1:l-1}^{(t)},\, t\big) - \big(x_l - x_l^{(0)}\big)\right\|^2\right] \tag{2}$$

where t is the flow timestep and l is the chunk index. We provide ablations on our design choices and hyperparameters in §5.

4 Experiments

We demonstrate the impact of DynaTokens: (1) whereas all current state-of-the-art camera-controlled video models struggle greatly with dynamics, DynaTokens can effectively enable dynamics within such models via test-time training; (2) alternative test-time training methods, such as LoRA [11] or TTT layer [7], fail, while DynaTokens enables learning dynamics while maintaining camera control. We evaluate on a broad range of videos with dynamic changes, including motion ordering, dynamic spatial relations, text-based motion accuracy, physics, and style transformations.

Benchmarks and Metrics. Since there are no existing dynamics-focused benchmarks for camera-controlled world models, we repurpose existing video benchmarks for dynamics-related evaluation: VBench2 [35] and WorldScore [8]. We evaluate two aspects: dynamics correctness and camera control correctness.

VBench2 is a text-to-video benchmark with VLM judges. While VBench2 holistically evaluates video models, we use its dynamics-related categories, "Dynamic Spatial Relations" (DSR, *e.g.,* "a dog is on the left of the table, then the dog runs to the front of the table") and "Motion Order Understanding" (MOU, *e.g.,* "a person is cooking, then they suddenly start organizing the pantry"). The original evaluation assumes a static camera, thus we make slight modifications to accommodate camera movement, such as not evaluating on frames where the main object is out-of-view due to camera movement. For camera, we calcaulte an aggregate score over rotation and translation error like WorldScore [8], leveraging state-of-the-art camera estimation method, ViPE [12]. We evaluate on a random subset of 20 prompts across the two categories, each for 6 different camera trajectories, on a total of 120 videos for VBench. Standard errors are reported in the Appendix.

WorldScore provides an input image and text prompt, and evaluates on optical-flow-based heuristics for dynamics. We focus on "Motion Accuracy" (MA) and "Camera Control" (Cam) scores. The original motion evaluation assumes a static camera, thus we make slight modifications to accommodate the moving camera, detailed in §F of the Appendix. We evaluate on 15 scenes selected randomly from the benchmark, each with 6 different camera trajectories, on a total of 90 videos for WorldScore. Standard errors are reported in the Appendix.

4.1 Comparing with state-of-the-art camera-controlled models

State-of-the-art camera-controlled models struggle with dynamics. While methods like Hydra [6] and LiveWorld [9] improve out-of-frame dynamics by modifying their memory, they still struggle with in-frame dynamics. We show that DynaTokens, via test-time training, effectively enables dynamics.

We compare with a suite of general camera-controlled video models – WorldPlay [23], Lingbot [25], and Lyra2 [22] – as well as dynamics-focused methods – Hydra [6] and LiveWorld [9]. For WorldPlay, we evaluate their camera-post-trained checkpoint. We also compare with Genie3 [10] qualitatively on a small set of samples due to the UI-only access and queue time. Refer to §F for more details.

Table 1(a) shows results on VBench2 "Dynamic Spatial Relation" (DSR), "Motion Order Understanding" (MOU), WorldScore "Motion Accuracy" (MA), and cameral control (Cam) on both benchmarks. Current models have clear failure modes, as shown in Fig. 3 (top):

- **Static scenes**: this is a clear failure mode for both WorldPlay and Lyra2 in almost all examples.
- **Incorrect motion**: Lingbot tends to generate wrong motions, such as the bird flying away or the cat not jumping. Genie 3 similarly generates wrong motions.
- **Duplicate subject**: LiveWorld splits one object into multiple, such as the 2 birds.
- **Poor visual quality**: Hydra often outputs blurry scenes (bird example).

More qualitative examples can be found in §A of the Appendix, and in the project webpage. These failures showcase that current camera-controlled models, when evaluated zero-shot, struggle with



Figure 3: Top: Comparison of DynaTokens with state-of-the-art camera-controlled video models. DynaTokens consistently enables dynamics, whereas existing models struggle, either keeping the object static, show incorrect motion, or produce low-quality videos. Bottom: Comparison across different test-time training strategies. Only DynaTokens is able to correctly learn dynamics while maintaining camera control. The camera control is overlayed on the frames using keyboard schema, "wasd" means translation and arrows mean rotation.

dynamics. As shown in Table 1(a), DynaTokens improves dynamics via test-time training, resulting in a 28% improvement in dynamic spatial relations, 35% in motion ordering on VBench2, and a 21% improvement in WorldScore motion accuracy. Additionally, DynaTokens maintains camera control on both benchmarks.

	VBench2			WorldScore	
	DSR	MOU	Cam	MA	Cam
Lyra2	0.54	0.17	0.87	1.11	**0.70**
Lingbot	0.68	0.25	0.82	4.37	0.64
WorldPlay	0.56	0.07	**0.88**	1.95	0.69
Hydra	0.51	0.18	0.61	2.51	0.56
LiveWorld	0.44	0.35	0.82	4.49	**0.70**
DynaTokens	**0.96**	**0.69**	0.87	**5.33**	0.69

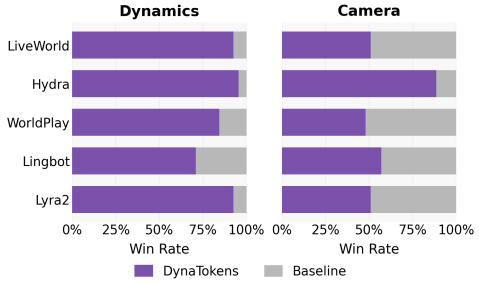
(a)

	VBench2			WorldScore	
	DSR	MOU	Cam	MA	Cam
DynaTokens	**1.00**	**0.72**	**0.88**	**4.46**	**0.67**
LoRA	0.61	0.22	0.83	2.79	0.56
LoRA-no PRoPE	0.41	0.06	0.84	3.62	**0.67**
Fine-tuning	0.59	0.67	0.83	2.66	0.54
TTT Layer	0.41	0.11	**0.88**	3.79	0.66

(b)

Table 1: (a) Comparing DynaTokens with state-of-the-art camera-controlled video models. We evaluate Dynamic Spatial Relation (DSR) and Motion Order Understanding (MOU) on VBench2 and Motion Accuracy (MA) on WorldScore, as well as camera. Current state-of-the-art models struggle greatly with dynamics. DynaTokens, via test-time training, effectively learns dynamics while maintaining camera. (b) DynaTokens outperforms alternative test-time training methods, LoRA, finetuning, and TTT layer for dynamics. Standard errors are in Table 4 and Table 5 in Appendix.

We additionally perform human evaluation, shown on the right. We perform pairwise ranking between DynaTokens and each baseline. For every baseline, we show DynaTokens' win rate against it across 100 rankings done by 5 independent annotators. DynaTokens shows clear advantage of 71%-95.5% win rate for dynamics. DynaTokens outperforms all other baselines except for WorldPlay on camera, with which it is a close match at 48%.



4.2 Comparison with other test-time methods

We have shown DynaTokens to effectively "teach" camera-controlled models dynamics while maintaining camera via test-time training in the previous section. Now, we further show that alternative test-time training methods are not as effective. We compare DynaTokens with LoRA, modified LoRA to preserve the PRoPE branch (LoRA-no PRoPE), fine-tuning last model blocks, and TTT layer for video models introduced by [7].

Table 1(b) shows results on VBench2 (DSR, MOU) and WorldScore. For each method, we use WorldPlay as the base model, and train on the same trajectories. We evaluate on 3 seen and 3 unseen camera trajectories for each scene to calculate the average score for dynamics and camera control. We test-time train each method on 12 distinct scenes and evaluate a total of 72 videos.

As shown in Table 1, DynaTokens shows clear advantage in dynamics learning across both benchmarks and all metrics, corroborating our hypothesis that disentangled design for localized dynamic learning enables effective learning while maintaining camera control. In the qualitative examples shown in Fig. 3(bottom), all other methods fail to fully learn dynamics, such as making the person "get up and stretch", or making the kangaroo "jump to the right of the box". More qualitative examples can be found in §A in the Appendix and on the project webpage.

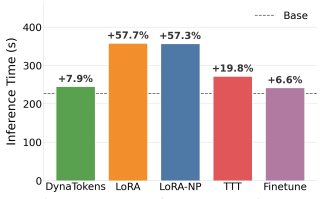


Figure 4: Inference Time

LoRA, although high in representation capacity, cannot effectively disentangle dynamics from camera, and holistically "steers" the generation output, thus unable to effectively learn dynamics and yields worse camera control. We show additional analysis on LoRA and LoRA-no PRoPE in §5. Fine-tuning the last transformer blocks is also not sufficient for learning dynamics. TTT Layer, although preserving camera, shows distorted objects, as shown in the human example of Fig. 3, and is unable to effectively learn dynamics. More qualitative examples can be found in Fig. 13 and Fig. 14 in the Appendix. We provide further analysis on the failure of LoRA and its effect on PRoPE in §5.

In addition to superior performance in both dynamics and camera, DynaTokens is also faster: as shown in Fig. 4, DynaTokens only adds 7.9% inference time compared to the base model (WorldPlay), while showing great improvement in dynamics. In comparison, LoRA has a much higher overhead of 57.7% inference time, despite having fewer parameters (30M = 0.3% of base model 8.6B parameters) than DynaTokens (240M, 2.7% base model parameters). Inference is evaluated on 4 A100 GPUs.

4.3 Generalization to Unseen Camera Paths

The space of all possible camera paths is large – given the duration of videos we generate (up to 93 frames), there are more than 10^{19} total possible camera trajectories. While trained only trains on $\approx$ 15 camera paths, DynaTokens achieves generalization to diverse camera poses. Table 2 compares DynaTokens' performance on seen and unseen camera paths on VBench2. Each evaluated unseen and seen path differ by 39.4 degrees geodesic distance on average, and differ completely in the first 28 frames' action, demonstrating sufficient spread in the exploration trajectories.

Trajectories	Dynamics	Camera
Seen	0.83	0.85
Unseen	0.81	0.86

Table 2: DynaTokens demonstrates strong generalization, evaluated on VBench2.

4.4 Enabling diverse dynamics: physics and stylistic change

We evaluate whether DynaTokens can handle diverse dynamics such as physics and style transformations. While DynaTokens is designed to learn localized dynamics, we stress-test our approach for global style transformations to probe its capability.

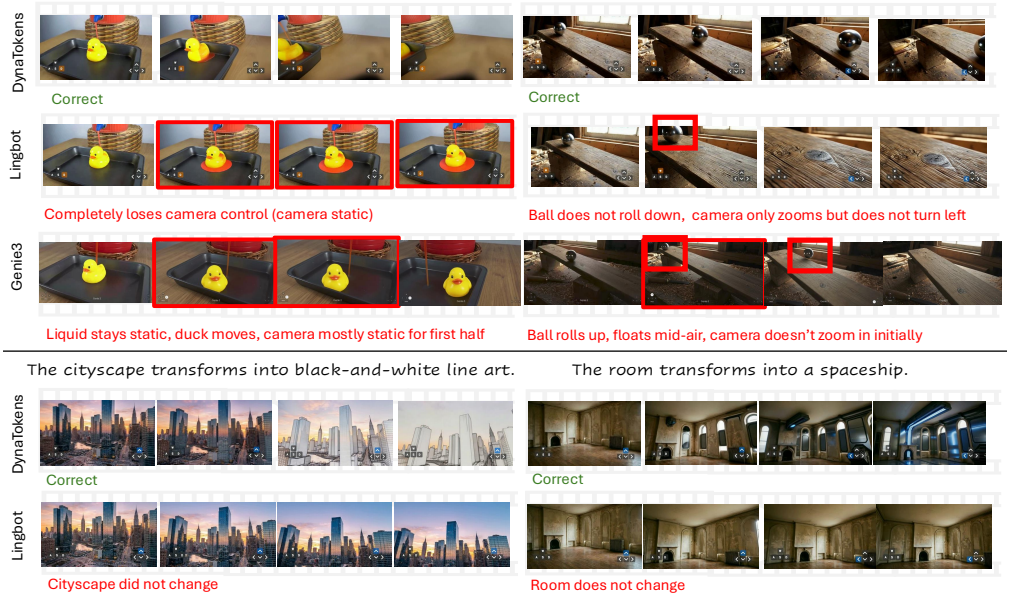


Figure 5: DynaTokens enables diverse dynamics, including physics and stylistic changes, whereas current state-of-the-art camera-controlled video models like Lingbot and Genie 3 struggle.

We show qualitative examples on Physics IQ [19] and StevoBench [18]. As shown on the top of Fig. 5, the best open-source state-of-the-art model, Lingbot, either fails to evolve physics (ball not rolling) or completely fails to turn the camera. Genie 3, a closed-source state-of-the-art camera-controlled model, also fails to evolve physics correctly. DynaTokens successfully learns to evolve physics in these settings. We evaluate qualitatively since Physics IQ's evaluation is based on pixel-wise alignment (IoU, MSE) under a static-camera ground truth, which cannot evaluate camera-controlled models. More examples can be found in Fig. 15 in the Appendix.

As shown on the bottom of Fig. 5, DynaTokens is capable of stylistic changes such as changing a cityscape into black-and-white line art, or transforming a room into a spaceship, while allowing the viewer to navigate the scene with camera control. Lingbot fails to transform the scene in both cases.

5 Ablations and Analyses

Why does LoRA fail to capture dynamics? Table 1 shows that LoRA fails to simultaneously enable dynamics and preserve camera. We provide further analysis.

(1) Non-localized effect. While DynaTokens enables localized gating, LoRA is unable to represent such localized functions since its update is "global" to all patches: $F(h_i) = (W + BA^\top)\, h_i$. The



Figure 6: Attention map visualization of DynaTokens for a complex scene with four moving objects (four ducks moving in different directions). The attention maps localize the moving ducks.

matrix multiplication formulation results in "patch mixing", and introduces global changes that interfere with camera. (2) Preserving PRoPE. For models that use PRoPE, the QKV matrix updates from LoRA result in a shift in the PRoPE product, as detailed in §H of the Appendix. To alleviate this, we also experiment with LoRA-no PRoPE, which removes LoRA on the PRoPE path. However, LoRA-no PRoPE drastically worsens dynamics while only slightly improving camera compared to LoRA, as shown in Table 1(b).

Ablating the design of DynaTokens. We ablate design of DynaTokens, and evaluate alternatives such as adding dynamic tokens directly to the patch tokens (additive), only using one token, changing the token dimensions, and eliminating context noising (using clean historical frames) during training.

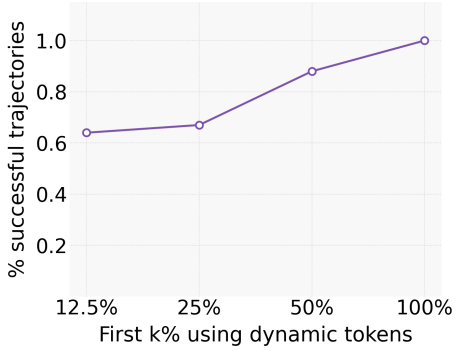
Table 3 shows that the current design achieves the best performance overall on VBench2 dynamics tasks, considering both dynamics and camera metrics. Training with a single dynamic token, while slightly better on camera, compromises dynamics metrics greatly. This is likely because the softmax attention weight, which acts as localized gating, is no longer meaningful. No context noising, although slightly better on the simpler Motion Order Understanding tasks, causes a significant drop in Dynamic Spatial Relations, which require later chunks of the video to produce spatially-correct motions despite the discrepancy between the model's self prediction at inference time with teacher forcing training.

	DSR	MOU	Cam
DynaTokens	**1.00**	0.72	0.88
Single token	0.57	0.56	**0.89**
Reduce token dim (32)	0.65	0.72	0.88
Increase token dim (128)	0.60	0.72	0.87
No context noising	0.67	**0.83**	0.84
Additive	0.91	0.67	0.85

Table 3: Ablations on dynamic token design evaluated on VBench2. The current design yields best holistic performance on dynamics and camera. DSR: Dynamic Spatial Relation. MOU: Motion Order Understanding.

Additional visualization of attention map localization. We additionally visualize the localized attention map in a complex scene with multiple moving objects in Fig. 6, where four ducks move in different directions. Fig. 7 compares the attention map from DynaTokens, the attention weight delta caused by LoRA, as well as the attention map from the text keyword of the moving object (kangaroo in this example). The comparison shows that DynaTokens' localization is significantly better.

Probing the temporal aspect of DynaTokens. We probe whether the base model is capable of continuing dynamics after the motion is kick-started by only applying DynaTokens to the first 1/8, 1/4 and 1/2 of the full video. As shown on the right, even only using dynamic tokens for the first 1/8 can yield over 64% successful dynamic trajectories, and using it for the first half yields successful dynamics 88% of the time. This suggests that once the motion is "kick-started", the base model has the capability of continuing it via autoregressive chunk-by-chunk generation. We also evaluate the application DynaTokens on longer videos, showcasing that the learned dynamics is not limited to the training duration, shown in Fig. 19 in the Appendix.



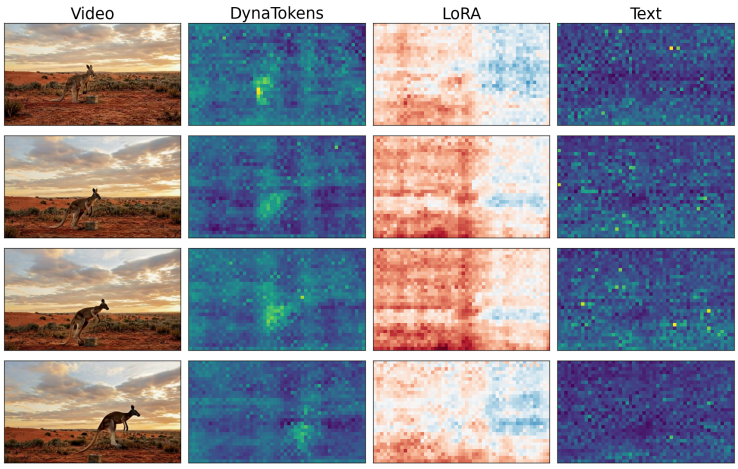


Figure 7: Attention map visualization of DynaTokens, LoRA, and the text keyword of the moving object (kangaroo). DynaTokens shows significantly stronger localization.

Robustness to Training Trajectories. We perform additional ablations on a subset of evaluation scenes to evaluate the robustness of DynaTokens to the quantity and quality of training trajectories. We find that even when training on as few as 3 trajectories, DynaTokens shows a considerable advantage on MOU (0.50 vs. 0.35 for the best baseline, LiveWorld), which is a positive result. Other metrics (DSR, MOU, MA) beat baselines even at 6 trajectories.

To ablate training trajectory quality, we swap good training trajectories with corrupted trajectories that contain background inconsistencies or sudden motion changes. DynaTokens' performance remains high, above baselines even with 33% of training trajectories corrupted. These ablations show that DynaTokens is robust to both the quantity and the quality of the test-time training trajectories, likely due to its capability to leverage the strong priors from the base model. More details can be found in §E of the Appendix.

6 Conclusions and Discussions

Dynamic scenes are a key challenge in camera-controlled world models. We present DynaTokens, a lightweight add-on module that can be test-time-trained to enable camera-controlled video models to learn localized dynamics for a given scene. This design is able to "selectively" teach dynamics while enabling good camera control, whereas other designs such as LoRA fail. We provide analysis on why the design of dynamic tokens can effectively disentangle local dynamics with global camera control, and show analyses with attention weights to demonstrate the localized gating effect of our design. DynaTokens, via test-time training, greatly surpasses current state-of-the-art camera-controlled video models on VBench2 dynamic categories and WorldScore, and can extend to physical dynamics and style transfer. Our analysis on the dynamic token design sheds light on the representation of dynamics and camera control for video models. DynaTokens also opens up the opportunity for data generation for dynamic-rich, camera-controlled videos in diverse settings, paving the way for generating dynamic, navigable worlds in the future. For more info, visit our project page https://glab-caltech.github.io/dynatokens/.

7 Acknowledgements and Disclosures

We thank Damiano Marsili, Aadarsh Sahoo, Jiacheng Liu, and Jhan Liufu for discussions. This project was funded by the Packard Fellowship, the Powell Foundation, an Amazon Research Award and a Google Faculty Award. We also thank Google for providing us with Gemini credits.

References

[1] Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 942–963. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/akyurek25a.html.

[2] Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B. Lindell, Zan Gojcic, Sanja Fidler, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. In *International Conference on Learning Representations (ICLR)*, 2026.

[3] W. G. C. Bandara and Vishal M. Patel. Attention prompt tuning: Parameter-efficient adaptation of pre-trained models for action recognition. *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–10, 2024. URL https://api.semanticscholar.org/CorpusID:271115318.

[4] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *CVPR*, 2025. Licensed under Modified Apache 2.0 with special crediting requirement.

[5] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2025.

[6] Kaijin Chen, Dingkang Liang, Xin Zhou, Yikang Ding, Xiaoqiang Liu, Pengfei Wan, and Xiang Bai. Out of sight but not out of mind: Hybrid memory for dynamic video world models. *arXiv preprint arXiv:2603.25716*, 2026.

[7] Karan Dalal, Daniel Koceja, Jiarui Xu, Yue Zhao, Shihao Han, Ka Chun Cheung, Jan Kautz, Yejin Choi, Yu Sun, and Xiaolong Wang. One-minute video generation with test-time training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17702–17711, 2025.

[8] Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27713–27724, October 2025.

[9] Zicheng Duan, Jiatong Xia, Zeyu Zhang, Wenbo Zhang, Gengze Zhou, Chenhui Gou, Yefei He, Feng Chen, Xinyu Zhang, and Lingqiao Liu. Liveworld: Simulating out-of-sight dynamics in generative video world models, 2026. URL https://arxiv.org/abs/2603.07145.

[10] Google DeepMind. Genie 3: A new frontier for world models. https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/, 2025. Accessed: 2026-03-03.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[12] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. In *NVIDIA Research Whitepapers arXiv:2508.10934*, 2025.

[13] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022.

[14] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

[15] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[16] Hansen Jin Lillemark, Benhao Huang, Fangneng Zhan, Yilun Du, and Thomas Anderson Keller. Flow equivariant world models: Memory for partially observed dynamic environments, 2026.

[17] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[18] Ziqi Ma, Mengzhan Liufu, and Georgia Gkioxari. Out of sight, out of mind? evaluating state evolution in video world models, 2026. URL https://arxiv.org/abs/2603.13215.

[19] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles?, 2025. URL https://arxiv.org/abs/2501.09038.

[20] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pages 28085–28128, 2025.

[21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2025.

[22] Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, Sanja Fidler, Jiahui Huang, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra 2.0: Explorable generative 3d worlds. *arXiv preprint arXiv:2604.13036*, 2026.

[23] Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world model. *arXiv preprint*, 2025.

[24] Kling Team, Jialu Chen, Yuanzheng Ci, Xiangyu Du, Zipeng Feng, Kun Gai, Sainan Guo, Feng Han, Jingbin He, Kang He, Xiao Hu, Xiaohua Hu, Boyuan Jiang, Fangyuan Kong, Hang Li, Jie Li, Qingyu Li, Shen Li, Xiaohan Li, Yan Li, Jiajun Liang, Borui Liao, Yiqiao Liao, Weihong Lin, Quande Liu, Xiaokun Liu, Yilun Liu, Yuliang Liu, Shun Lu, Hangyu Mao, Yunyao Mao, Haodong Ouyang, Wenyu Qin, Wanqi Shi, Xiaoyu Shi, Lianghao Su, Haozhi Sun, Peiqin Sun, Pengfei Wan, Chao Wang, Chenyu Wang, Meng Wang, Qiulin Wang, Runqi Wang, Xintao Wang, Xuebo Wang, Zekun Wang, Min Wei, Tiancheng Wen, Guohao Wu, Xiaoshi Wu, Zhenhua Wu, Da Xie, Yingtong Xiong, Yulong Xu, Sile Yang, Zikang Yang, Weicai Ye, Ziyang Yuan, Shenglong Zhang, Shuaiyu Zhang, Yuanxing Zhang, Yufan Zhang, Wenzheng Zhao, Ruiliang Zhou, Yan Zhou, Guosheng Zhu, and Yongjie Zhu. Kling-omni technical report, 2025. URL https://arxiv.org/abs/2512.16776.

[25] Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

[26] World Labs Team. Atlas: A world model for spatial intelligence. *World Labs Blog*, 2026. https://www.worldlabs.ai/blog/atlas.

[27] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras, 2021.

[28] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference of Computer Vision (ECCV)*, page 36–54, 2024. ISBN 978-3-031-72666-8.

[29] Yiming Wang, Qihang Zhang, Shengqu Cai, Tong Wu, Jan Ackermann, Zhengfei Kuang, Yang Zheng, Frano Rajič, Siyu Tang, and Gordon Wetzstein. Bullettime: Decoupled control of time and camera pose for video generation. *arXiv preprint arXiv:2512.05076*, 2025.

[30] Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, et al. Worldcompass: Reinforcement learning for long-horizon world models. *arXiv preprint*, 2026.

[31] Wei Yu, Runjia Qian, Yumeng Li, Liquan Wang, Songheng Yin, Sri Siddarth Chakaravarthy P, Dennis Anthony, Yang Ye, Yidi Li, Weiwei Wan, and Animesh Garg. Mosaicmem: Hybrid spatial memory for controllable video world models. *arXiv preprint arXiv:2603.17117*, 2026.

[32] Mert Yuksekgonul, Daniel Koceja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan Kautz, Yejin Choi, James Zou, Carlos Guestrin, et al. Learning to discover at test time. *arXiv preprint arXiv:2601.16175*, 2026.

[33] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

[34] Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T. Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations (ICLR)*, 2026.

[35] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

A Additional Qualitative Results

Fig. 9 & 10 & 11 & 12 show additional qualitative comparisons between DynaTokens and current state-of-the-art camera-controlled models. Fig. 15 shows additional examples of physical dynamics, and Fig. 16 shows that DynaTokens can enable dynamics of multiple objects. Fig. 13 & 14 show additional comparisons between DynaTokens and other test-time training methods. For interactive video visualization, please visit our project webpage https://glab-caltech.github.io/dynatokens/.

B Evaluation: Standard Errors

	VBench			WorldScore	
	DSR	MOU	Cam	MA	Cam
Lyra2	0.54 (0.071)	0.17 (0.042)	0.87 (0.017)	1.11 (0.226)	**0.70 (0.015)**
Lingbot	0.68 (0.071)	0.25 (0.058)	0.82 (0.017)	4.37 (0.605)	0.64 (0.023)
WorldPlay	0.56 (0.070)	0.07 (0.032)	**0.88 (0.009)**	1.95 (0.273)	0.69 (0.013)
Hydra	0.51 (0.066)	0.18 (0.050)	0.61 (0.019)	2.51 (0.520)	0.56 (0.029)
LiveWorld	0.44 (0.061)	0.35 (0.062)	0.82 (0.016)	4.49 (0.817)	**0.70 (0.013)**
DynaTokens	**0.96 (0.030)**	**0.69 (0.061)**	0.87 (0.011)	**5.33 (0.531)**	0.69 (0.015)

Table 4: Comparison across methods on VBench and WorldScore with standard errors across test samples. DynaTokens' advantage is significant.

	VBench			WorldScore	
	DSR	MOU	Cam	MA	Cam
Dynatoken	**1.00 (0.000)**	**0.72 (0.082)**	**0.88 (0.011)**	**4.46 (0.538)**	**0.67 (0.060)**
LoRA	0.61 (0.113)	0.22 (0.101)	0.83 (0.008)	2.79 (0.523)	0.56 (0.062)
LoRA-no PRoPE	0.41 (0.123)	0.06 (0.056)	0.84 (0.008)	3.62 (0.337)	**0.67 (0.056)**
Finetuning	0.59 (0.123)	0.67 (0.114)	0.83 (0.021)	2.66 (0.358)	0.54 (0.064)
TTT Layer	0.41(0.110)	0.11 (0.076)	**0.88 (0.007)**	3.79 (0.384)	0.66 (0.061)

Table 5: Comparison of DynaTokens with other test-time training methods, including LoRA, block finetuning, and TTT layer, with standard errors across evaluation examples. DynaTokens' advantage is significant.

Table 4 and Table 5 report standard errors for each method across the test set. DynaTokens' advantage is significant both compared to baseline world models and compared to alternative test-time training methods.

C Additional Comparison with VPT/APT

	VBench			WorldScore	
	DSR	MOU	Cam	MA	Cam
Dynatoken	**1.00**	**0.75**	**0.86**	**5.15**	**0.69**
VPT/APT	0.50	0.17	0.84	2.35	0.67

Table 6: Comparison of DynaTokens and VPT/APT. Naive prefix tuning methods cannot disentangle dynamics from camera control, and struggles to learn dynamics effectively.

We perform additional comparison with prefix-tuning methods, Visual Prompt Tuning (VPT) [13] (and its video version APT [3]) on a subset of evaluation scenes. The implementation of VPT/APT in the camera-controlled video generation setting is the same. When trained on the same data, VPT/APT struggles to learn dynamics effectively, stays close to the static settings for many (especially unseen) trajectories, and sometimes generates artifacts such as duplicating objects. Quantitatively, DynaTokens shows a significant advantage in Table 6.

	IoU	IoU-d1
Dynatokens	**0.342**	**0.511**
LoRA	0.006	0.017
Text	0.011	0.024
Chance	0.010	0.022

Table 7: IoU and IoU-d1 (relaxed to count all distnace $\leq$1 neighbors as correct) of Dynatokens attention map, attention map delta before and after LoRA, the attention map from text keywords corresponding to the moving object (such as the word "dog"), as well as a random attention mask pattern. DynaTokens has a significantly higher overlap with the SAM mask of the moving object, while LoRA and text perform similarly to random chance.

The comparison with VPT and APT illustrates an important design aspect of DynaTokens. While VPT and APT were originally designed for recognition (VPT for images, and APT a more efficient version for videos), applying them to camera-controlled video generation means inserting trainable "data tokens" alongside the original data tokens. While DynaTokens also inserts new tokens, the mechanism of token interaction with the base model is very different. Unlike VPT/APT, DynaTokens does not participate in the global self attention alongside the data tokens, where camera control takes place. As shown in Fig. 1(a) in the main paper, the camera conditioning happens via PRoPE in the self-attention. VPT/APT-style methods will interfere with camera conditioning (the added tokens are entangled in the camera and data path), thus failing to disentangle object movement from camera movement. This entanglement is exactly the source of issues with camera-controlled video models and the motivation behind DynaTokens.

D Additional Analysis of DynaTokens' Attention Map Localization

We quantify DynaTokens' attention localization by measuring the overlap between attention maps with SAM [20] masks via IoU and IoU-d1 (relaxing the IoU by counting all distance $\leq$1 neighbors as correct). We note that while the attention map illustrates DynaTokens' localization to the motion, its purpose is not to segment the exact object geometry in video frames. Additionally, DynaTokens work in latent patches which are coarse spatially and temporally (a latent patch is 16x16 pixels across 4 physical frames), and thus we expect coarse overlap rather than accurate, high-IoU boundary alignment. We compare with the attention map delta caused by LoRA, as well as the attention map from the text keyword of the moving object. We additionally show the expected IoU of a random attention mask pattern, denoted by "Chance". As shown in Table 7, DynaTokens has a significantly higher overlap with the SAM mask of the moving object, while LoRA and text perform similarly to random chance.

We further calculate Ripley's L statistic which measures whether points in 2D have a clustered distribution pattern (higher means more clustered). DynaTokens scores 5.95, LoRA 4.44, text keyword 2.17 (L(3)-3), which shows a clear lead by DynaTokens.

Fig. 17 & 7 & 18 show additional attention map visualizations with DynaTokens and compares them to LoRA and text-based attention maps. These visualizations further support our localized dynamics hypothesis that motivates our DynaTokens design.

E Robustness to Training Trajectories

We perform additional ablations on a subset of evaluation scenes to evaluate the robustness of DynaTokens to the quantity and quality of training trajectories. In Table 8, we perform an additional study by training on 3/6/12 trajectories on a subset of 30 evaluation videos. Notably, even when training on as few as 3 trajectories, DynaTokens shows a considerable advantage on MOU (0.50 vs. 0.35 for the best baseline, LiveWorld), which is a positive result. Other metrics

# trajectories	VBench			WorldScore	
	DSR	MOU	Cam	MA	Cam
3	0.67	0.50	0.80	4.02	0.67
6	0.75	0.67	0.80	4.68	0.68
12	1.00	0.75	0.86	4.81	0.68
15	1.00	0.75	0.87	5.15	0.68

Table 8: Studying the effect of the number of training trajectories. Even when training on as few as 3 trajectories, DynaTokens shows a considerable advantage on MOU (0.50 vs. 0.35 for the best baseline, LiveWorld).

(DSR, MOU, MA) beat baselines even at 6 trajectories. This shows that DynaTokens is effective even with a small number of training trajectories.

In Table 9, we perform an additional "swapping" analysis on a subset of 30 evaluation videos: we swap up to 33% of good training trajectories with corrupted trajectories (containing background inconsistencies or sudden motion changes). We observe DynaTokens' performance remains high, above baselines even with 33% of training trajectories corrupted by artifacts. This is likely due to the DynaTokens formulation retaining the base model prior of coherent video generation.

# artifact swap	VBench			WorldScore	
	DSR	MOU	Cam	MA	Cam
0 (original)	1.00	0.75	0.87	5.15	0.68
1 (7%)	1.00	0.75	0.86	4.99	0.66
5 (33%)	1.00	0.67	0.86	4.50	0.66

Table 9: Studying the effect of the quality of training trajectories by swapping k good trajectories with k trajectories with artifacts. Even with 33% trajectories swapped to corrupted, DynaTokens' performance remains high.

F Benchmark Evaluation Details

For VBench2 evaluation, we use the "Dynamic Spatial Relations" and "Motion Order Understanding" categories. We use VBench's original VLM evaluation framework with minor modifications to accommodate camera movement, since the original framework is designed for static camera. For example, VBench2 determines object spatial relation by the first and last frame. Since camera-controlled models might occasionally turn the camera away from the main object (*e.g.,* camera turns left while object moves right and out of frame), we add an "object presence" detection and only apply this check when the main object is present in the last frame. We also remove the questions solely based on the first frame, since all methods we test take the same first frame as input. We use Gemini 2.5 Flash instead of Llava for better spatial relation understanding. To avoid Gemini bias, we additionally use GPT-5 for verification, which yields very similar results: the mean difference is 0.025, which is less than 10% of the observed advantage of DynaTokens over the best baseline (Lingbot) on dynamic spatial relations. For methods that generate highly distorted or unrecognizable subjects, we add "recognizable and not distorted" to the evaluation prompt. We use a state-of-the-art camera estimation method, ViPE [12] to estimate camera, and calculate camera score based on a geometric mean of rotation and translation errors, where rotation error is denoted by geodesic distance, similar to WorldScore [8]. Since the translation is dependent on the unit for each model, we apply scaling to correct for the unit conversion.

The WorldScore benchmark's dynamic evaluation is largely based on optical flow, and assumes a static camera (thus low background flow). To adapt it to the moving camera setting, we adapt the Motion Accuracy and Camera Control metrics. Two other metrics, Motion Magnitude and Motion Smoothness, are both dominated by camera-induced flow rather than scene dynamics and are thus not suitable for this task. Motion Accuracy uses SEA-RAFT [28] dense flow and SAM2 [21] object masks to score whether the prompted subject moves more than its surroundings; we (i) set the object-region flow to zero when SAM2 fails to localize the subject, so lost or distorted subjects are penalized rather than collapsing into a whole-frame statistic, and (ii) replace the background maximum with the background *median*, since under camera motion the background flow distribution is heavy-tailed and its maximum is unstable. Camera Control recovers the trajectory via DROID-SLAM [27] and scores the clipped geometric mean of rotation geodesic error and scale-aligned translation ℓ_2 error, with $C^* = \arg\min_C \|t_{\text{gt}} - C\, t_{\text{pred}}\|_2$; we constrain $C \in [c_{\min}, c_{\max}]$ to avoid the loophole where under-translating prediction is rescaled to match the ground-truth magnitude.

For evaluation of the baseline camera-controlled models, we apply conversion from action string (which WorldPlay takes as input) to camera matrices for Lingbot, Lyra2, LiveWorld and Hydra. Since Hydra only takes video input, and our setting does not have history, we repeat the initial frame as the input video.

G Test-Time Training Details

Data Curation. We generate video samples with dynamics to train dynamic tokens. Given an initial frame I_0, text prompt τ, and a target camera trajectory π partitioned into N segments, $\pi_{1:1}, \ldots, \pi_{1:N}$, we construct an imperfect video supervision $(I_0, \tau, \pi) \mapsto V$:

(i) Dynamics synthesis. A VLM (Gemini 3.1 Pro) first enhances τ by making the dynamics more explicit. Then, an image-to-video model conditioned on I_0 and the enhanced caption is used to produce a static-camera clip V_{dyn}. The same VLM then scores V_{dyn} on physical plausibility and camera staticity, refining the caption until passes.

(ii) Camera projection. We sample $N{+}1$ uniformly spaced state frames $\{S_0, \ldots, S_N\}$ from V_{dyn}. For each i, we run HY-WorldPlay on $(S_i, \tau_\varnothing, \pi_{1:i})$ with the neutral prompt $\tau_\varnothing$="`A static scene.`", recovering a camera-aligned view $\hat{S}_i$ at the boundary of $\pi_{1:i}$. For instance, with π="`forward-6, right-7, forward-6`" and $N{=}3$, we run HY-WorldPlay three times: on S_1 with prefix $\pi_{1:1}$="`forward-6`", on S_2 with $\pi_{1:2}$="`forward-6, right-7`", and on S_3 with $\pi_{1:3}{=}\pi$. This yields $\hat{S}_1, \hat{S}_2, \hat{S}_3$ that each capture the scene state S_i as observed from the camera position reached after executing $\pi_{1:i}$.

(iii) Interpolation and stitching. Consecutive keyframes $(\hat{S}_{i-1}, \hat{S}_i)$ are bridged by a dual-endpoint image-to-video interpolator, Kling o3 [24], guided by a transition caption that the VLM writes from the two endpoints and the camera-action label. The per-segment clips are concatenated into the final video V, which is passed to a VLM for review.

It is important to note that, although our curated videos contain both scene and camera motion, stitching can introduce artifacts in background appearance and motion continuity. For example, the scene layout or object appearance may change across frames, and temporal discontinuities may arise. Qualitative examples can be found in Fig. 8. However, since the base model remains frozen and dynamic tokens contribute only through localized, additive cross-attention, this imperfect supervision can teach scene dynamics without corrupting the pretrained prior. Model inference calls take 16 minutes on 8 A100 GPUs (with shared prefix skipping), and API call for Gemini and Kling cost $1.52 for all training trajectories. Our DynaTokens design can train on few trajectories, enabling camera-controlled models to produce correct dynamics for new, unseen camera trajectories.

We show examples of curated data below. They contain background inconsistencies since each keyframe's projection might not "hallucinate" unseen space in the same way.



Figure 8: Examples of curated training examples. Curating one sample takes 20 GPU minutes (A100) and cost $1.52 in API calls (Kling and Gemini). The curated data contains correct dynamics but may include background inconsistencies. Our test-time training approach resolves such inconsistencies by leveraging model priors.

Training. We train $K = 16$ dynamic tokens of dimension $m = 64$ in our experiments, for videos ranging from 61 to 93 frames. We use the Muon [14] optimizer with learning rate 3e-3 with weight decay and L2 regularization. The training memory is around 22GB on 4 GPUs, for which we use 40GB or 80GB A100s. For context noising, we add high noise (step [500, 985)) to the context chunks. Training takes 80 minutes for 15 trajectories (total 900 frames). To put this in context, just doing inference using Lyra2, a baseline video model, on our 6 evaluation trajectories, takes 66 A100 minutes. This suggests the test-time adaptation, which unlocks inference on diverse unseen trajectories, is not a huge overhead compared to the per-trajectory inference cost.

Training limitations. DynaTokens is a lightweight module added to pretrained video models. Dynamic tokens enable dynamics by effectively disentangling it from camera control. However, if the base model is poor at certain dynamics, learning with dynamic tokens should not expect to fully fix these issues, such as complex physics.

H Additional Analysis on LoRA's Failure

We provide more detailed derivation on LoRA's effect on PRoPE:

$$q_i = (W_Q + \Delta W_Q)h_i, \qquad k_j = (W_K + \Delta W_K)h_j \tag{3}$$

$$\begin{aligned} \tilde{q}_i = D(c_i)\, q_i, \qquad \tilde{k}_j = D(c_j)\, k_j, \\ s_{ij} = \big\langle D(c_i)W_Q h_i,\ D(c_j)W_K h_j \big\rangle + \big\langle D(c_i)\Delta W_Q h_i,\ D(c_j)W_K h_j \big\rangle \\ + \big\langle D(c_i)W_Q h_i,\ D(c_j)\Delta W_K h_j \big\rangle + \big\langle D(c_i)\Delta W_Q h_i,\ D(c_j)\Delta W_K h_j \big\rangle \end{aligned} \tag{4}$$

Only the first term corresponds to the original PRoPE value.

We also provide more analysis on why LoRA-No PRoPE fails. While it does not significantly reduce the number of free parameters compared to normal LoRA, this might be causing great discrepancy between the PRoPE vs. non-PRoPE stream (which was designed to be symmetric in the original architecture), thus hindering learning. This shows that LoRA, with its global entanglement across spatial patches, struggles to only learn dynamics without affecting camera control. Dynamic tokens, on the other hand, minimizes the effect on PRoPE by being local - unaffected patches' keys and values stay mostly stable.

I Societal Impact

We improve generative video models. We acknowledge that generative models can be used in ways that raise ethical concerns, including the creation of misleading synthetic media, amplification of societal biases present in training data, and potential misuse for surveillance or harmful automation. Our work is intended solely for scientific research purposes and focuses on advancing the understanding and capability of generative modeling systems. We do not release systems or artifacts designed for deceptive or malicious use, and we encourage responsible deployment practices, including transparency about generated content, careful dataset curation, and evaluation for fairness and safety.

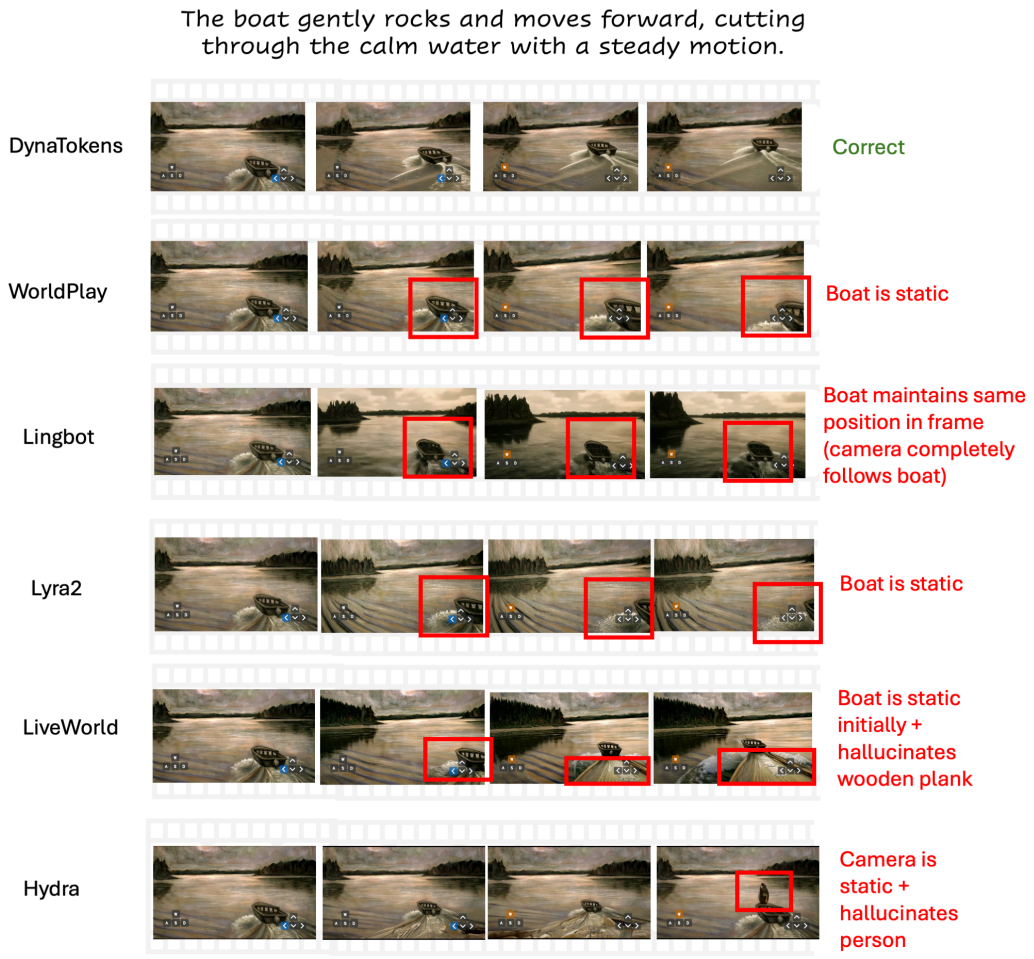


Figure 9: Additional qualitative examples of comparison with state-of-the-art camera-controlled video models.

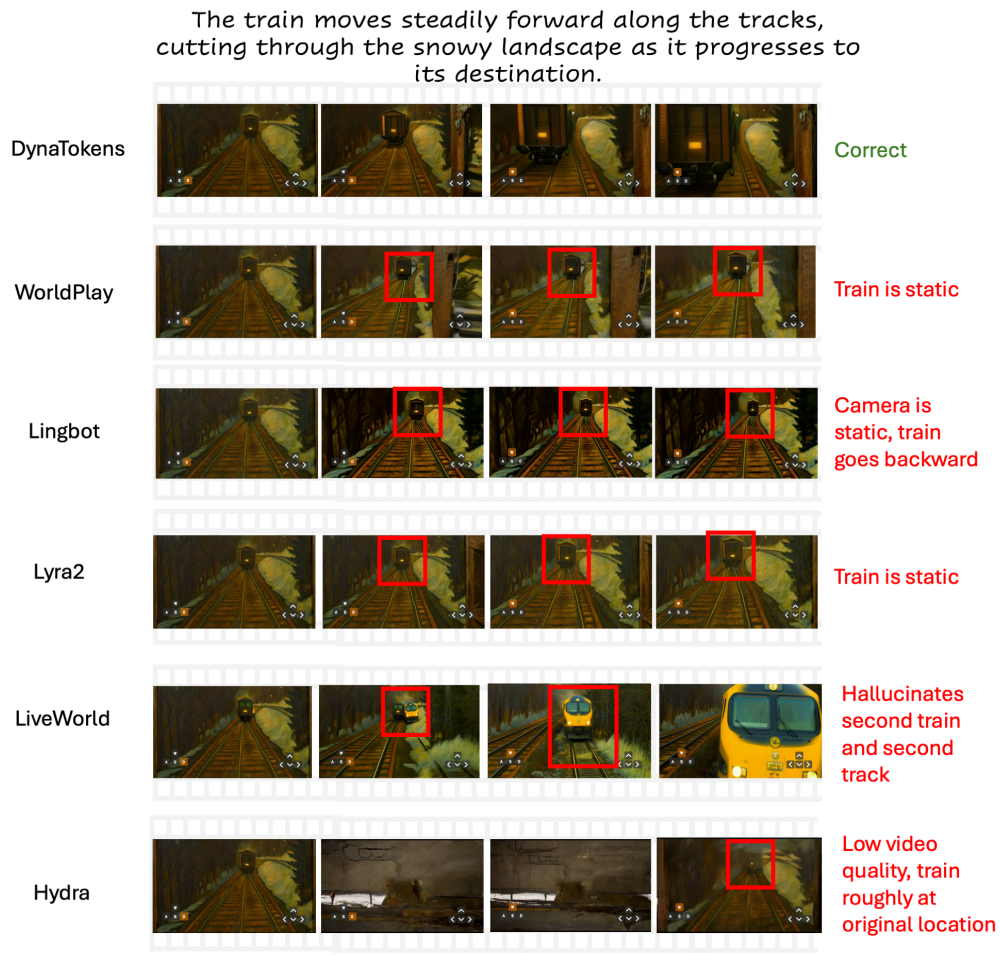


Figure 10: Additional qualitative examples of comparison with state-of-the-art camera-controlled video models.

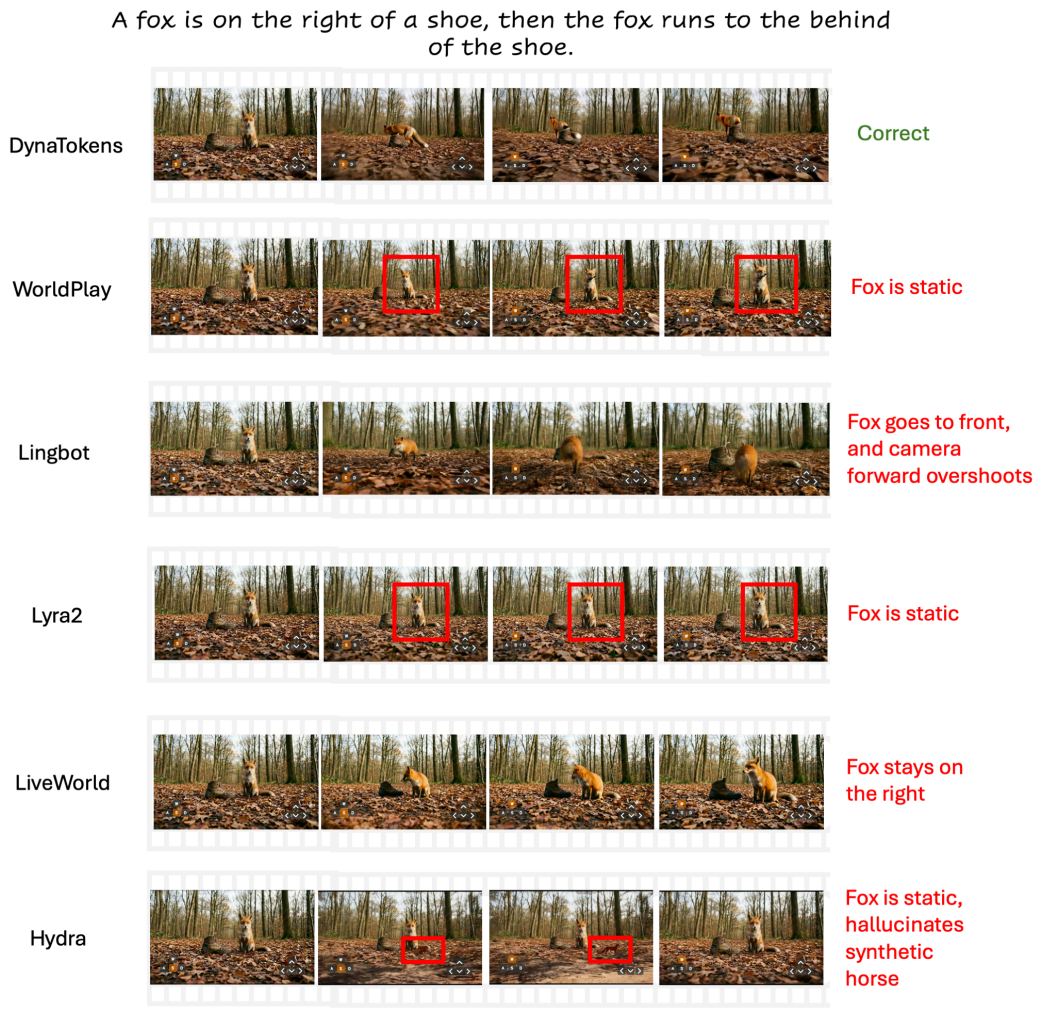


Figure 11: Additional qualitative examples of comparison with state-of-the-art camera-controlled video models.

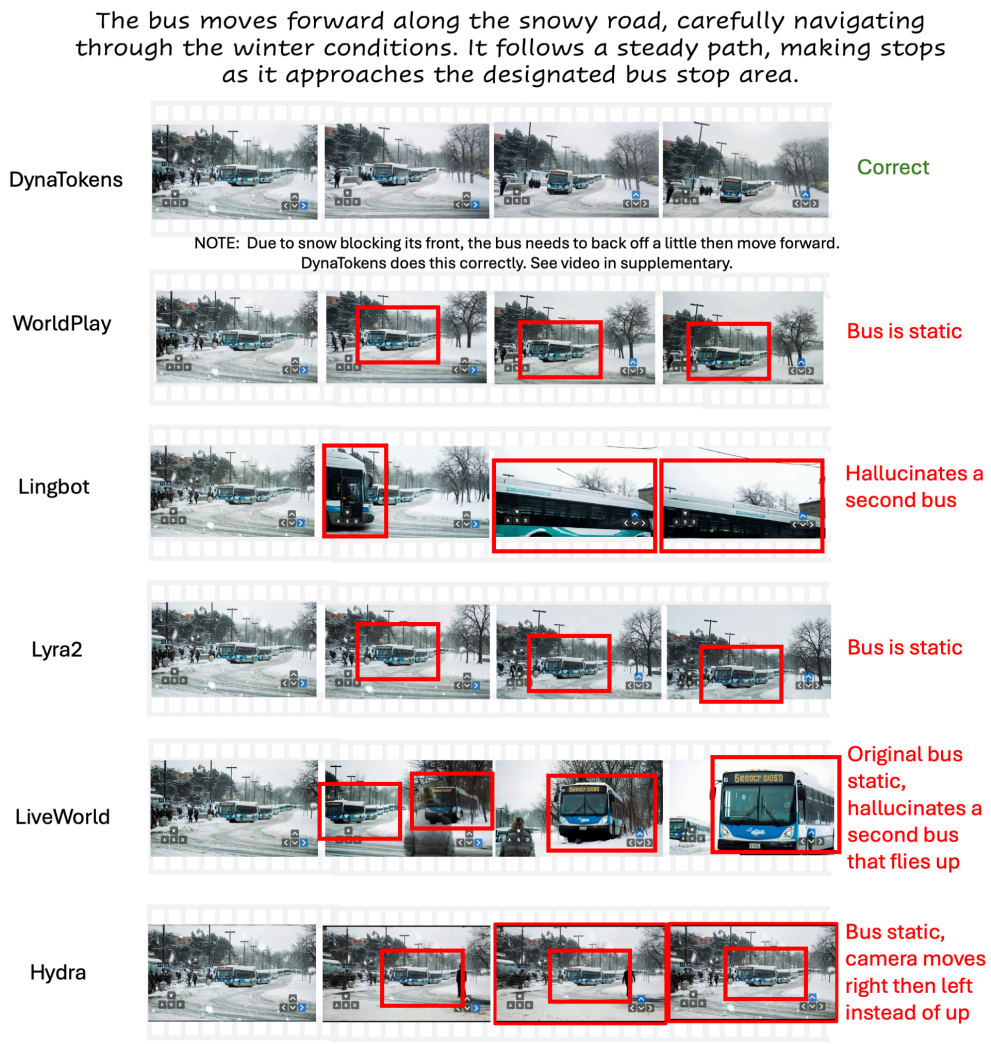


Figure 12: Additional qualitative examples of comparison with state-of-the-art camera-controlled video models.

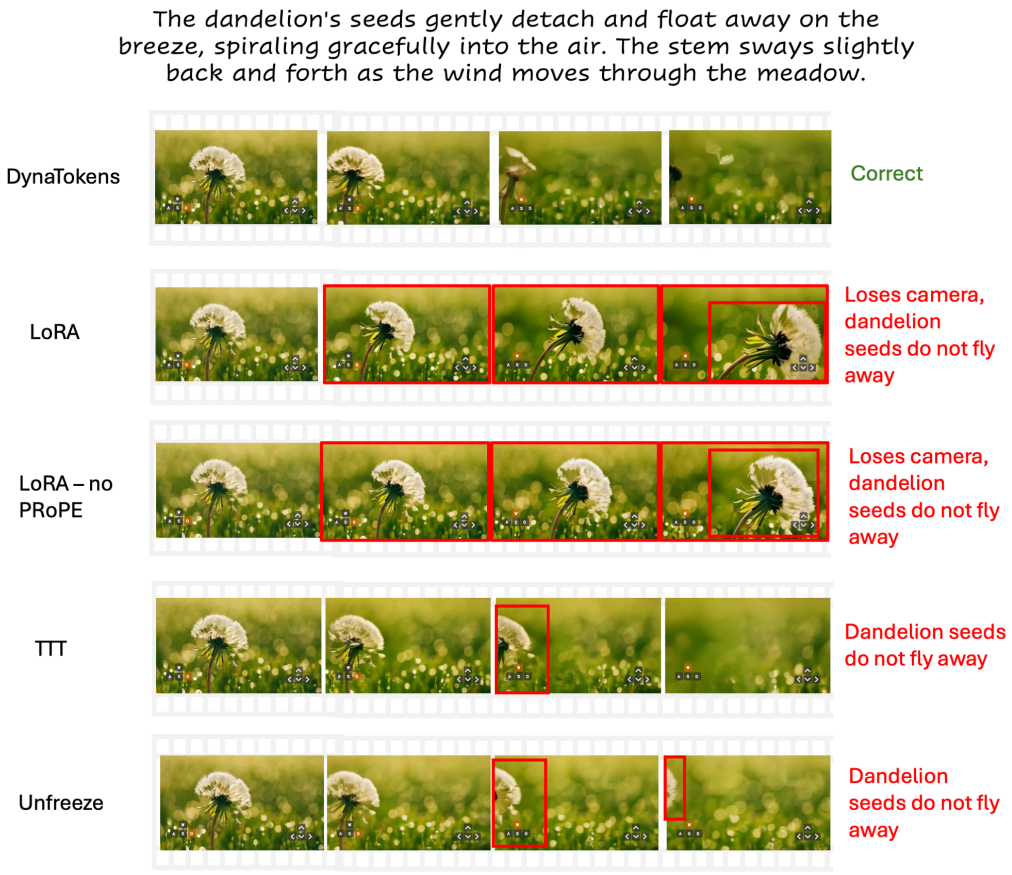


Figure 13: Additional qualitative examples of comparison across test-time methods.

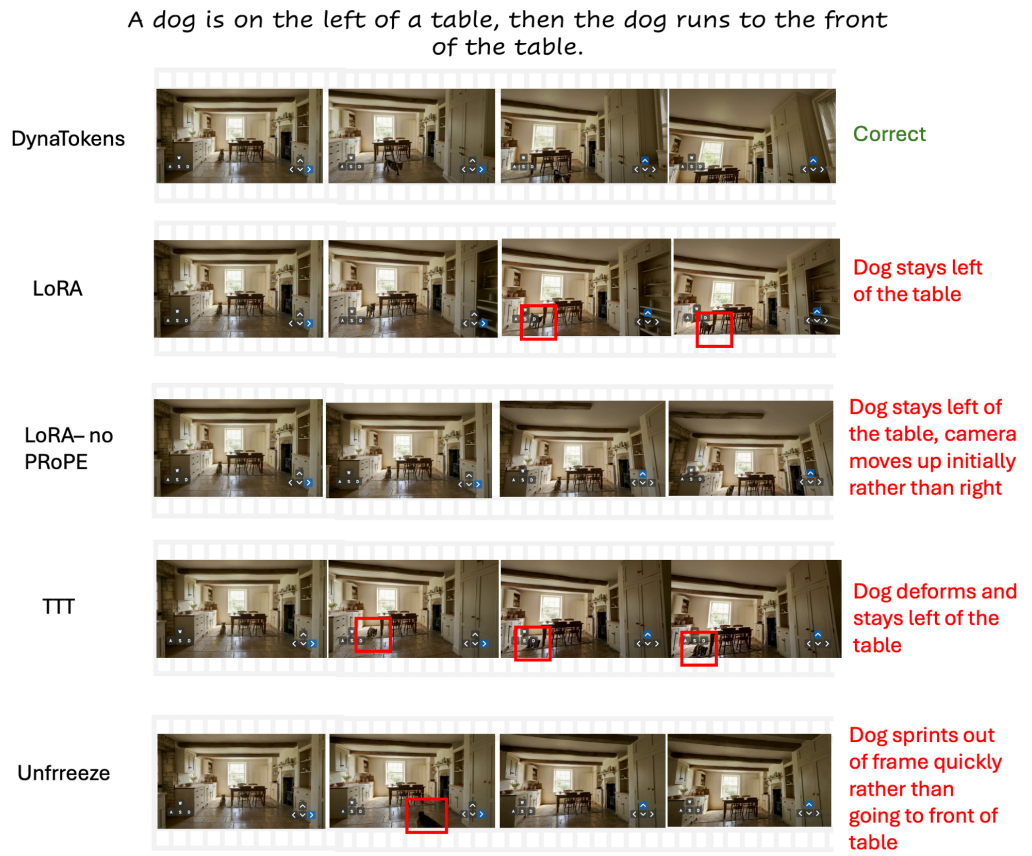


Figure 14: Additional qualitative examples of comparison across test-time methods.

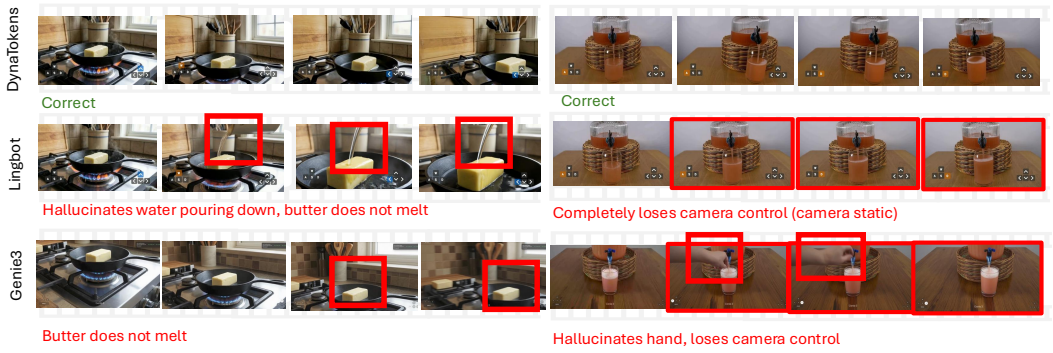


Figure 15: Additional qualitative examples for physical dynamics.

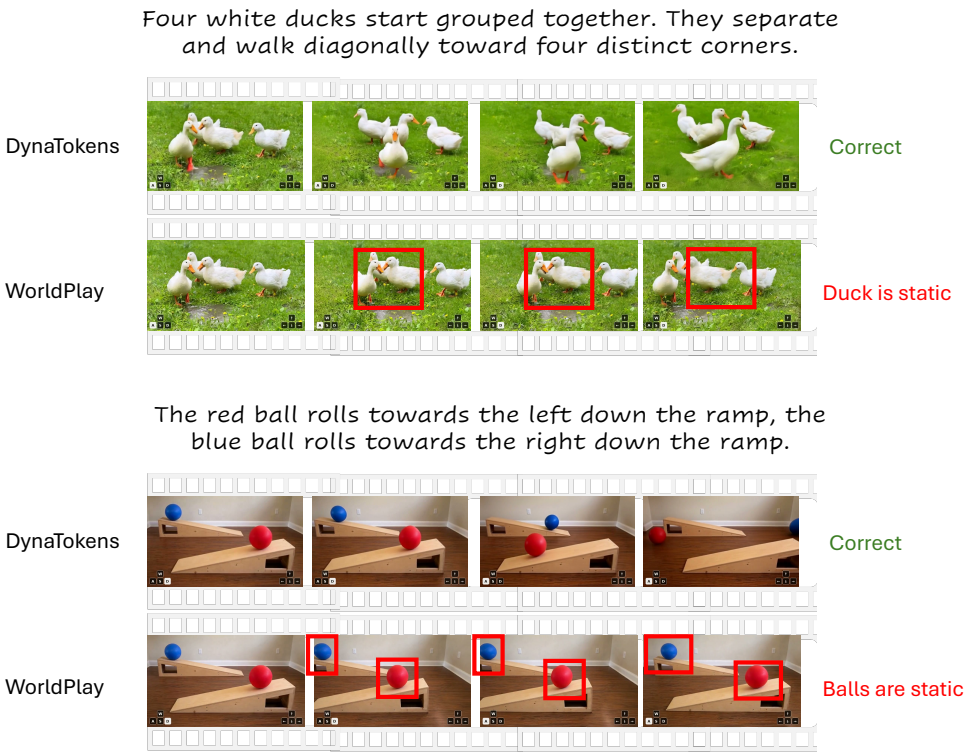


Figure 16: DynaTokens can enable dynamics of multiple objects.

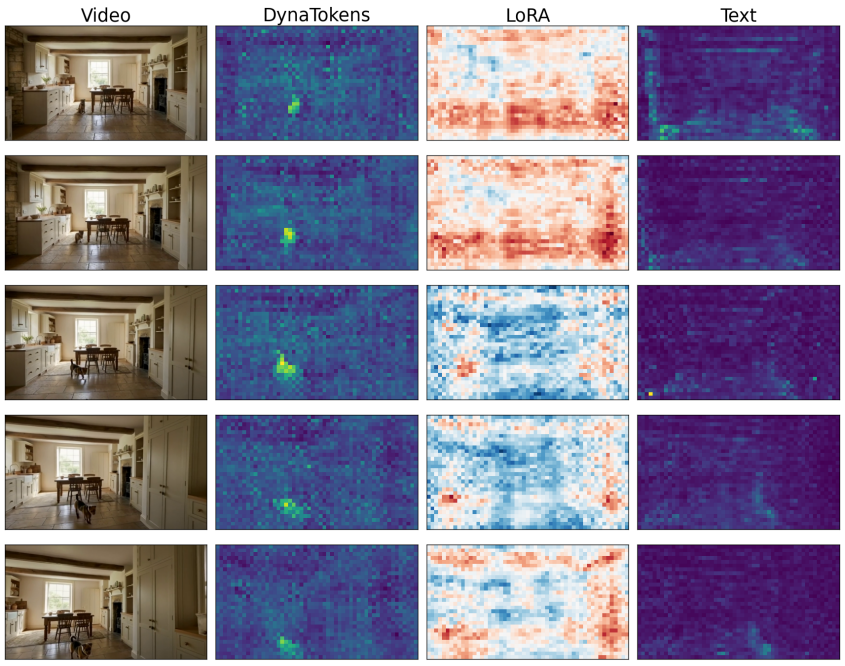


Figure 17: Attention map visualization of DynaTokens, LoRA, and the text keyword of the moving object (dog).

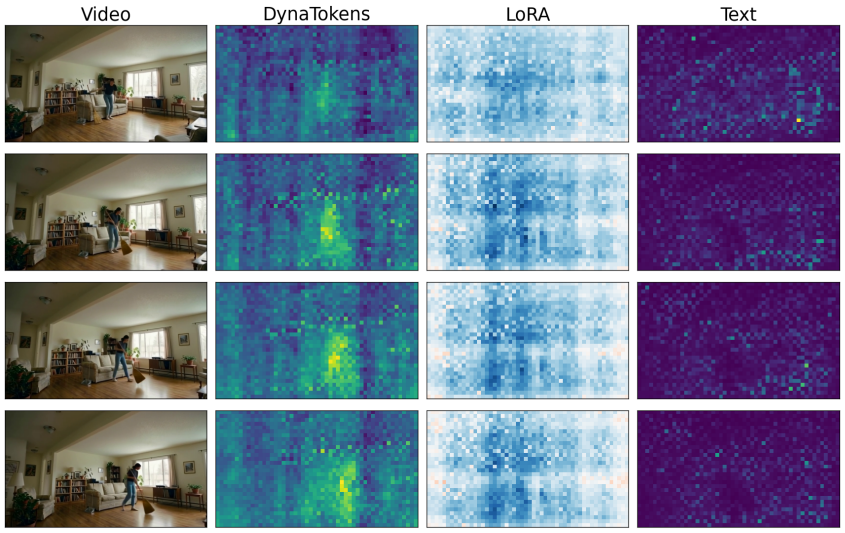


Figure 18: Attention map visualization of DynaTokens, LoRA, and the text keyword of the moving object (person).

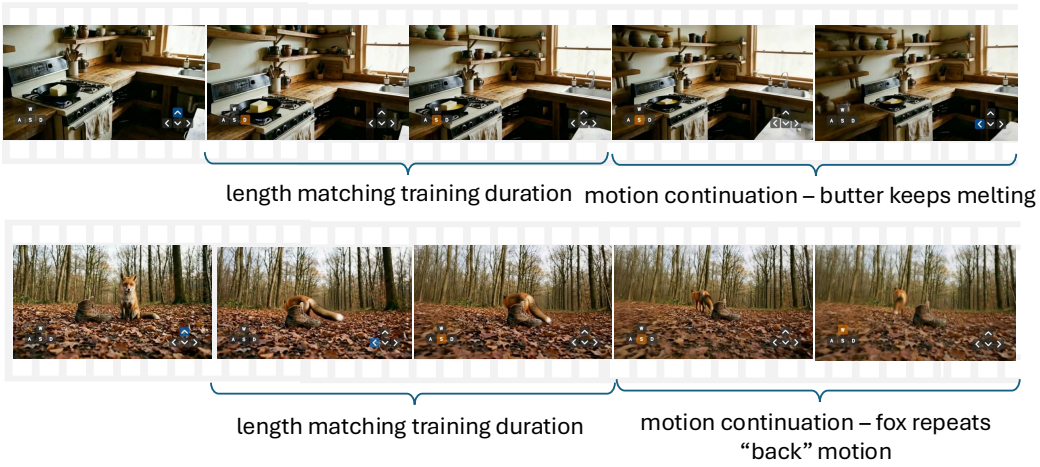


Figure 19: Evaluation on longer videos.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction introduce DynaTokens, a test-time method designed to let camera-controlled video models learn dynamics. While the paper mainly focuses on the application of camera-controlled video generation, it conveys that architectures designed to match the learning target (localized dynamics) enables effective learning. The paper provides analyses and experiments to support this.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: DynaTokens is a lightweight module trained at test time to surpass the base model capabilities. However, if the base model has significant limitations (e.g. for complex physics), then training with DynaTokens cannot completely solve the problem.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: We provide analysis on motivating example and LoRA's effect on training and PRoPE. We provide full assumptions and proof.

   Guidelines:

   - The answer [N/A] 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 fully disclose all details and submit code in supplementary.

   Guidelines:

   - The answer [N/A] 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: We open access to everything.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe all hyperparameters and training and evaluation details in the method and ablations section.

   Guidelines:

   - The answer [N/A] 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: We follow the convention in evaluating on public video benchmarks, where the evaluation examples are fixed, and usually people do not report error bars and report exact metrics on the fixed benchmark.

   Guidelines:

   - The answer [N/A] 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 provide compute resource information in the main text and appendix.

   Guidelines:

   - The answer [N/A] 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: We follow the ethics guidelines.

   Guidelines:

   - The answer [N/A] 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: We provide discussion on societal impact in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not directly pose risks.

    Guidelines:

    - The answer [N/A] 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 ackowledge the base models and benchmarks we use.

    Guidelines:

    - The answer [N/A] 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 documentation.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: We do not use human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: We do not involve human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not use LLM as important original part of the research.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.