StaDy: Factorizing the World into Static and Dynamic via Likelihood Matching

Frank Fundel* **Thomas Ressler-Antal***
Malek Ben Alaya **Stefan Andreas Baumann** **Björn Ommer**

CompVis @ LMU Munich, Germany
Munich Center for Machine Learning (MCML)



Figure 1: **StaDy mitigates capacity-induced appearance leakage in latent action models.** Motion transfer: a latent action z extracted from a driving pair (x_s, x_t) is applied to a different recipient source. **Top**: the standard latent action model leaks the driving identity as the latent dimensionality grows. **Bottom**: adding StaDy transfers the motion faithfully while preserving the recipient's appearance across the full sweep.

Abstract

Latent action models (LAMs) aim to learn compact representations of state transitions directly from visual observations, but they suffer from a fundamental ambiguity: the latent code can encode target-state information rather than true dynamics. Existing approaches address this issue through restrictive bottlenecks, which reduce leakage at the cost of limiting expressivity. We propose StaDy, a regularization framework that encourages latent codes to retain useful dynamics while reducing target information beyond that inherent in the dynamics. Concretely, we introduce a likelihood-matching objective that aligns the decoder's predictions conditioned solely on the latent variable with its unconditional predictions. This discourages target memorization without a restrictive latent bottleneck. A controlled synthetic environment with known motion-transfer targets demonstrates that StaDy separates dynamics from static content. We further show that these benefits translate to diverse open-world videos, where StaDy supports substantially larger latent bottlenecks that improve motion understanding and transfer while limiting leakage more effectively than an unregularized baseline. Project page: `https://compvis.github.io/StaDy`.

*Equal contribution.

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

1 Introduction

The visual world is inherently dynamic, defined by the motion and change that animate it. When we watch a video, we perceive not just the appearance of a scene but the dynamics that transform it from one moment to the next. A hand reaches for a cup; a crowd disperses; a door swings open. These changes are meaningful precisely because they are separable from the static content of the scene; they describe what happened, not just what is there. Learning compact representations of visual dynamics is therefore a central challenge in representation learning, video generation, and world models. Latent action world models are increasingly relevant to robotic planning, where they provide a learned interface for predicting the consequences of candidate actions [12].

A promising line of work approaches this by learning latent actions directly from raw video, without ground-truth labels of any kind [3, 9, 27, 33]. Given two consecutive frames, a model distills the transformation between them into a compact latent code, which it then uses to reconstruct the later frame from the earlier one. The learned code should capture dynamics, not content, representing how a scene evolves rather than what it looks like. When successful, such latent actions offer a structured, self-supervised representation of motion that can guide scene evolution in generative models [26] or serve as a proxy for actions in settings where explicit labels are unavailable [5, 11, 20]. This enables world-model pretraining on large collections of unlabeled video, with the learned action interface subsequently supporting optimization of action sequences for planning [12].

Despite their appeal, these models can suffer from a fundamental ambiguity. Because the encoder observes both frames simultaneously, it can take a shortcut: rather than representing the transformation from source to target, the latent code can absorb information about the target's appearance, identity, or layout. The model still minimizes its reconstruction loss, but the learned representation need not capture the intended dynamics. Instead of answering "how does this scene evolve?", it answers "what does the next frame look like?". This content leakage undermines the core premise of latent action learning, yet it is difficult to detect from reconstruction metrics alone; a model that exploits this shortcut can appear to perform well while learning very little about the underlying dynamics.

Existing methods address this primarily through capacity control: the latent code is made low-dimensional, discrete, or otherwise bottlenecked [3, 11, 14, 27, 33], restricting how much target information can pass through the encoder. This strategy is mostly effective, but creates a tension. If the bottleneck is too loose, target information leaks into the code; if it is too tight, the code cannot express the diversity and granularity of real visual dynamics. The same mechanism that suppresses leakage also limits representational power, and this ceiling becomes especially constraining in scenes with multiple objects, fine-grained geometry, or articulated motion.

We argue that leakage should not be addressed by restricting information flow alone. Instead, the goal is to retain useful dynamics while reducing excess target information, allowing for predictiveness inherent in the dynamics themselves. We introduce StaDy, a regularization framework that penalizes differences between the decoder's latent-only and unconditional predictions. Together with reconstruction, this discourages target memorization while preserving information needed to model the transition. This distinction matters for how the representation scales. A capacity-limited latent space can appear disentangled only because it is too small to memorize the target. The capacity needed to represent the dynamics depends on their complexity and is generally unknown in advance. StaDy, by contrast, aims to make target memorization suboptimal even when the latent space is enlarged. This enables a different scaling behavior: increasing the capacity of the dynamics representation can improve its ability to capture fine-grained motion without proportionally increasing content leakage. We test this by evaluating capacity gains jointly with leakage and retained motion information.

We instantiate this principle in a flow-matching framework, using the decoder as an implicit critic of latent-only target predictiveness. Controlled synthetic and facial-motion experiments demonstrate reduced leakage. Beyond these settings, we train on diverse open-world video: larger latents improve motion recognition on four benchmarks, while transfer and direct target-feature probes show substantially less leakage growth than the unregularized baseline.

Our contributions are threefold:

1. We identify target-state leakage as a capacity-dependent failure mode of latent action models and show why bottleneck-based mitigation imposes an inherent trade-off between disentanglement and expressivity.

2. We propose StaDy, a likelihood-matching regularizer that discourages target memorization while preserving useful dynamics without a restrictive latent bottleneck.
3. We demonstrate that larger latent spaces improve downstream motion representations on diverse open-world video while limiting the accompanying growth in appearance leakage.

2 Related Work

Latent action world models. A growing body of work learns latent action models from passive video by training an inverse dynamics encoder jointly with a forward predictor, so that the latent variable inferred from a pair of frames captures the transition between them in a self-supervised manner. This recipe has been explored in simulated and game environments [9, 27] and scaled to foundation-level world models trained on hundreds of thousands of hours of internet gameplay [3], to internet videos of humans and robots performing manipulation tasks [33], and to a range of world-modeling and visuomotor control settings [5, 11, 20]. Garrido et al. [12] extend latent action world models to diverse in-the-wild videos and reuse the learned action interface for robotic planning. A closely related thread targets motion transfer rather than control: DisMo [26] learns latent actions from raw video via a flow-matching reconstruction objective and conditions an off-the-shelf video generator on it, with the explicit goal of producing a representation invariant to appearance, structure, viewpoint, and object category. WhatMoves [10] extends this idea for spatial scene composition. In all of these methods the encoder observes both the source frame x_s and the target frame x_t, which is precisely the regime in which target leakage can arise: nothing in the reconstruction objective alone forces the latent code to describe the transition rather than the target itself. The dominant mitigation is capacity control, via VQ-VAE-style discrete bottlenecks [30] with small codebooks [3, 27, 33] or β-VAE KL penalties on continuous latents [1, 11, 14]: a smaller channel has less room to encode the target frame. CLAW [2] instead uses gradient reversal through a latent-only diffusion predictor to discourage future-information leakage. StaDy shares this goal but matches latent-only predictions to an unconditional reference rather than adversarially increasing prediction error, while reconstruction preserves useful dynamics.

Unconditional branches as references. StaDy builds on a now-standard ingredient in conditional generative modeling: training the same network to predict both with and without the conditioning signal. Classifier-free guidance [15] popularized this dual training and uses it at sampling time, interpolating between the conditional and unconditional predictions to trade off fidelity and diversity, and the same recipe extends naturally to flow-matching models [6, 34]. A related thread treats one prediction of the model as a reference for another: compositional diffusion combines multiple conditional predictions with a shared unconditional one at inference to compose concepts [22], and autoguidance steers the main branch using a weaker version of the model itself [17]. In all of these, the unconditional or auxiliary branch is used at inference, and the conditional prediction is pushed *away* from it. StaDy uses the same construction at training time and for the opposite purpose: we penalize differences between latent-only and unconditional predictions. The flow decoder thus acts as an implicit critic of marginal informativeness, and this pressure is balanced against reconstruction to retain useful dynamics, including their inherent target predictiveness.

3 Implicit Disentanglement in Latent Action Learning

Latent Action Learning. Latent action learning aims to recover a compact representation of the change between two observations $(\mathbf{x}_s, \mathbf{x}_t)$ – often consecutive frames of a video, though the formulation is not restricted to it – without access to ground-truth action labels. Typically, an inverse-dynamics encoder E_ϕ is paired with a forward decoder G_θ: the encoder infers a latent code $\mathbf{z} = E_\phi(\mathbf{x}_s, \mathbf{x}_t)$ from both observations, and the decoder reconstructs $\mathbf{x}_t$ from $\mathbf{x}_s$ and $\mathbf{z}$. The standard objective is a reconstruction loss

$$\mathcal{L}_{\text{rec}}(\phi, \theta) = \mathbb{E}_{(x_s, x_t)} \left[\mathcal{D}(x_t,\, G_\theta(x_s, z)) \right], \qquad z = E_\phi(x_s, x_t), \tag{1}$$

where $\mathcal{D}$ is a domain-appropriate distance. The encoder and decoder are jointly optimized, and when successful, z captures the transition between x_s and x_t rather than their static content. The decoder G_θ can be implemented using flow matching [21, 26]. In this setting, a time-dependent velocity field v_θ is learned to define a deterministic flow from a simple base distribution (typically $\mathcal{N}(\mathbf{0}, \mathbf{I})$) at

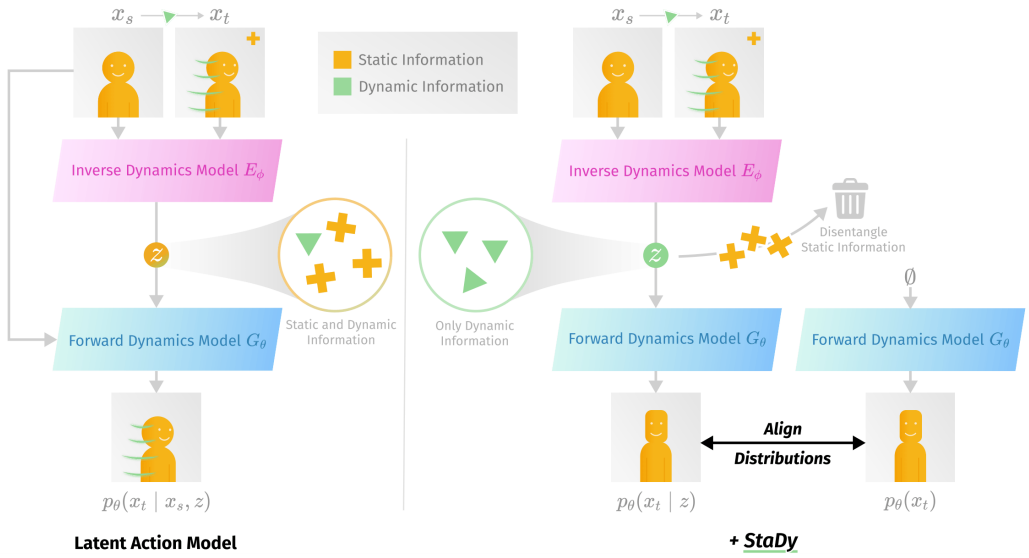

Figure 2: **Architecture. Left**: a standard latent action model encodes (x_s, x_t) into a latent z that mixes static and dynamic information. **Right**: StaDy adds a marginal-invariance objective that aligns $p_\theta(x_t \mid z)$ with the unconditional $p_\theta(x_t)$, discouraging z from carrying static information on its own while reconstruction encourages retention of useful dynamics.

$\tau = 0$ to the conditional data distribution at $\tau = 1$. The decoder is trained via the flow-matching reconstruction objective

$$\mathcal{L}_{\text{rec}}(\phi, \theta) = \mathbb{E}_{(\mathbf{x}_s, \mathbf{x}_t), \mathbf{x}_0, \tau} \left[\| v_\theta(\mathbf{x}_\tau, \tau; \mathbf{x}_s, \mathbf{z}) - u_\tau(\mathbf{x}_0, \mathbf{x}_t) \|_2^2 \right], \tag{2}$$

which regresses the target velocity u_τ at intermediate points $\mathbf{x}_\tau$. A common choice is linear interpolation, $\mathbf{x}_\tau = (1 - \tau)\mathbf{x}_0 + \tau \mathbf{x}_t$, yielding the target velocity $u_\tau = \mathbf{x}_t - \mathbf{x}_0$. After training, the decoder generates samples conditioned on $(\mathbf{x}_s, \mathbf{z})$ by integrating the learned ODE from $\tau = 0$ to $\tau = 1$.

Target Leakage. The intent is for $\mathbf{z}$ to describe the *transition* between $\mathbf{x}_s$ and $\mathbf{x}_t$ rather than their static content. However, nothing in the reconstruction loss enforces this: the encoder E_ϕ observes $\mathbf{x}_t$ directly, so Eq. (2) is minimized just as well by a code $\mathbf{z}$ that directly describes $\mathbf{x}_t$ in isolation rather than how $\mathbf{x}_s$ connects to it. We refer to this failure mode as *target leakage*: $\mathbf{z}$ becomes a direct predictor of $\mathbf{x}_t$ rather than a description of its relation to $\mathbf{x}_s$, the decoder learns to ignore $\mathbf{x}_s$ at least w.r.t. parts that $\mathbf{z}$ already specifies, and the resulting code need not retain the transferable structure desired of a latent action. However, target predictiveness is not itself evidence of leakage: dataset structure can make even true dynamics informative about the endpoint. We aim to remove additional target information while retaining this unavoidable dependence.

Implicit Disentanglement. Many latent action models restrict the information $\mathbf{z}$ can carry through low-dimensional latents [20], small discrete codebooks [3, 27, 33], or information bottleneck penalties [11]. The intended regime allows enough information to describe the transition, but too little to copy the target. Finding it requires tuning to the data's dynamics; latent dimensionality alone is not an information-theoretic capacity bound. If parts of the target are hard to predict from the source, describing how they changed may require just as much information as describing their final appearance. The bottleneck then provides little incentive to encode the change rather than the target itself.

Other methods encode an augmented frame pair while reconstructing the original target from the unaugmented source [4, 26, 29]. This discourages encoding appearance altered by the augmentation, provided the dynamics survive it. Target information unaffected by the augmentation can still leak. Both strategies therefore provide indirect controls, motivating an explicit regularizer.

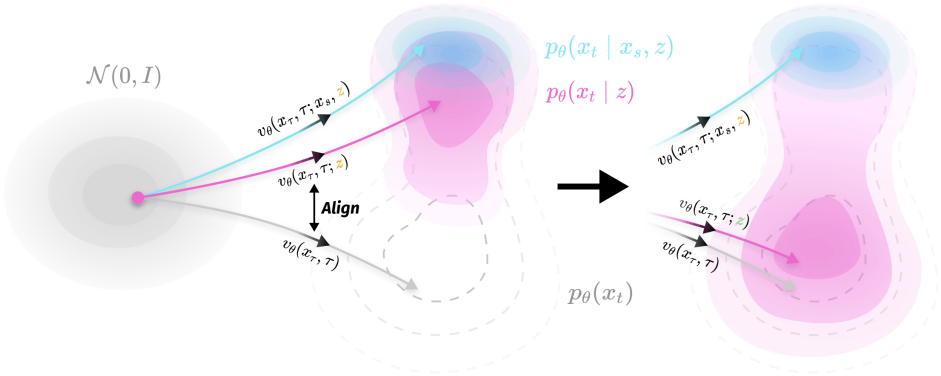

Figure 3: **Marginal-invariance regularization for latent action models.** A latent-action flow-matching decoder transports noise to target frames. The latent action z can shift the model distribution $p_\theta(x_t \mid z)$ away from the unconditional marginal $p_\theta(x_t)$. StaDy penalizes the squared difference between their velocity fields at matched (x_τ, τ), using the unconditional field as a reference and updating only the encoder through this penalty.

4 Marginal Invariance Regularization

Both capacity constraints and augmentations attempt to mitigate target leakage indirectly: the former by limiting channel capacity, and the latter by making target information inconsistent with the reconstruction objective. However, neither rules out leakage by construction. The reconstruction loss in Eq. (2) further exacerbates this issue: it only requires z to be predictive of x_t given x_s, without constraining *how* this prediction is achieved. In particular, a degenerate solution can encode x_t directly in z, making the prediction trivial while failing to represent the underlying transition. As illustrated in Fig. 2 (left), the latent then mixes static (appearance) and dynamic (motion) information. For latent action learning, we instead require z to capture *how* to move from x_s to x_t, rather than *where* we end up.

Conditional Informativeness and Marginal Independence. A useful latent action should be informative about x_t when combined with x_s, while limiting what it reveals about x_t on its own. Let a denote the task-relevant action. If a is recoverable from z, then, for finite mutual information,

$$\underbrace{I(z; x_t)}_{\text{latent target information}} = \underbrace{I(a; x_t)}_{\text{action target information}} + \underbrace{I(z; x_t \mid a)}_{\text{additional target information}} \geq I(a; x_t). \tag{3}$$

The floor need not be zero: an opening action can predict an open endpoint. Marginal independence is therefore an idealized reference, with the practical objective balancing reduced target predictiveness against retaining useful latent actions. Appendix C gives the proof and scope.

Marginal Invariance as a Soft Regularizer. We implement this balance by combining reconstruction with a penalty on latent-only target predictiveness. Model marginal invariance,

$$\underbrace{p_\theta(x_t \mid z)}_{\text{latent-only prediction}} = \underbrace{p_\theta(x_t)}_{\text{unconditional prediction}}, \tag{4}$$

provides the reference condition for the penalty. For a flow-matching generator, we compare latent-only and unconditional velocity fields:

$$v_\theta(x_\tau, \tau; z, \not{x}_s) \approx v_\theta(x_\tau, \tau; \not{z}, \not{x}_s), \tag{5}$$

where $\not{z}$ and $\not{x}_s$ denote the absence of z- and x_s-conditioning, respectively.
We therefore define the *marginal-invariance loss*

$$\mathcal{L}_{\text{inv}}(\phi) = \mathbb{E}_{x_s, x_t, x_0, \tau}\left[\left\| v_{\text{sg}[\theta]}(x_\tau, \tau; z, \not{x}_s) - v_{\text{sg}[\theta]}(x_\tau, \tau; \not{z}, \not{x}_s)\right\|_2^2\right], \qquad z = E_\phi(x_s, x_t), \tag{6}$$

where $\text{sg}[\cdot]$ is the stop-gradient operator. Crucially, both branches are detached from the decoder: we treat the unconditional field as a fixed reference defined by the data marginal, and the z-conditioned

field is held at the decoder's current behavior so that only the encoder E_ϕ is updated by this term to reduce latent-only target predictiveness. Without the latter detachment, G_θ could trivially minimize $\mathcal{L}_{\text{inv}}$ by learning to ignore z in the unconditional-on-x_s branch, satisfying the constraint while defeating its purpose. With θ gradient-stopped on both sides, $\mathcal{L}_{\text{inv}}$ depends on ϕ alone. The flow model thus serves as an implicit critic: differences between its latent-only and unconditional predictions provide a signal for reducing target information encoded in z, without explicitly computing a divergence between the generated image distributions.

Training G_θ. The generator G_θ is trained with the normal flow matching loss under full conditioning (x_s, z). Additionally, we apply classifier-free guidance [15]-style conditioning dropout, masking x_s and z with some probability, such that the unconditional and z-only velocity fields used in the regularizer are trained alongside. When x_s is dropped but z is retained, we stop the gradient through z in the flow matching loss. Otherwise, this branch would push z to be predictive of x_t on its own, directly opposing the regularizer. This avoids an additional training signal that explicitly rewards latent-only target prediction. Table 9 in Appendix G evaluates these stop-gradient choices and other architectural components.

Full Objective. The full training objective is the sum of the reconstruction and marginal-invariance losses,

$$\mathcal{L}(\phi, \theta) = \mathcal{L}_{\text{rec}}(\phi, \theta) + \lambda \, \mathcal{L}_{\text{inv}}(\phi), \tag{7}$$

where λ controls the strength of the regularizer. Both losses are minimized jointly: $\mathcal{L}_{\text{rec}}$ updates both ϕ and θ, while $\mathcal{L}_{\text{inv}}$ updates only ϕ by construction. Finite λ defines a trade-off: the learned decoder supplies a surrogate for target predictiveness, not a guarantee of independence or action sufficiency. Excessive regularization can suppress useful endpoint-correlated dynamics. We therefore evaluate leakage jointly with motion transfer and downstream performance (Section 5.3).

5 Experiments

We evaluate StaDy in three complementary settings. A controlled synthetic dataset provides exact motion-transfer targets and latent-only probes, allowing us to test whether latent actions separate motion from appearance. On CelebV-HQ videos of human heads, we evaluate motion transfer, identity leakage, and recipient identity preservation as latent capacity increases. Finally, diverse open-world videos test whether these benefits extend to more complex dynamics and enable larger latents to improve motion understanding and transfer.

Implementation Details. For the controlled synthetic experiments, we use a lightweight latent construction setup where target appearance, pose, and motion parameters are independently processed by MLPs and concatenated into a latent representation before being passed to the decoder. For experiments operating directly on image frames, we use a DINOv2-B [24] feature extractor. In the real-world setting, we implement the encoder as a 3D vision transformer [8] with an initial frame embedding stage consisting of a DINOv2-B model, and the decoder as a pixel-space diffusion transformer [25] with x_0-parameterization [18]. Additional implementation details and ablations of our main design choices and components are provided in the Appendix.

5.1 Controlled Tests on Toy Dataset

We first validate the method in a controlled synthetic setting where appearance and motion are disentangled by construction. This lets us probe two claims directly, at a finer level than is possible in real video: *(i)* the marginal-invariance loss suppresses appearance encoding, even when target appearance is explicitly available in the latent pathway; and *(ii)* the latent still captures motion, evaluated against pixel-level ground-truth transfers that are unavailable in real video.

Evaluation Setup. We construct a toy video dataset of single moving shapes. Each clip features one shape sampled from circle, square, triangle, ellipse, with appearance drawn from one of three texture modes (linear, angular, or stripe gradients) and size in $[16, 64]$ pixels. Trajectories follow a bouncing motion parameterized by linear and angular velocities up to 128 and 120 units per frame respectively. Because appearance and motion are independently sampled and trajectories are fully parametric, we can construct *ground-truth motion transfers*: given a source video's parameters and

a target appearance/shape, we render the exact trajectory applied to the new appearance, providing a pixel-level reference for evaluating motion transfer quality. For more detailed information, see Appendix Section D.1.

To isolate the effect of the proposed regularizer, we compare a non-regularized baseline against our regularized variant under matched architectures, data, and training settings.

(i) The marginal-invariance loss suppresses appearance encoding. We evaluate the proposed regularization in a controlled setting where the latent representation is explicitly constructed using target appearance information. Concretely, the latent is formed by concatenating MLP embeddings of the target appearance, target pose, and motion parameters before being passed to the decoder together with the source frame. This provides a stringent test of whether the regularizer can shape the learned representation to ignore appearance information despite it being directly available within the latent pathway. To probe this directly, we freeze the learned encoder components and train a lightweight probe decoder $\mathcal{P}$ to reconstruct the target frame from the latent representation alone. If the latent retains appearance information, the probe should reconstruct the target accurately; if the regularizer successfully suppresses appearance encoding, reconstruction should fail. Results are shown in Table 1. The regularized model achieves substantially worse probe reconstruction than the unregularized baseline, indicating that the latent is significantly less predictive of target appearance despite being explicitly constructed using target appearance features. Qualitative examples in Figure 4 show the same trend: the regularized model collapses to nearly uniform gray outputs, whereas the unregularized baseline produces comparatively accurate reconstructions.

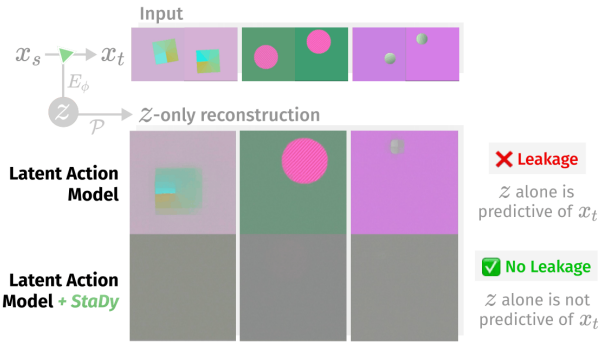

Figure 4: **Qualitative probing of target leakage.** Probe decoder $\mathcal{P}$ reconstructs target appearance from the unregularized latent, indicating leakage. With StaDy, reconstructions collapse to near-uniform outputs.

Table 1: **Probing target leakage via z-only reconstruction.** Higher metrics indicate less leakage. The frozen-encoder probe decoders show that StaDy effectively reduces target-appearance information being leaked in the latent z.

Method	L1 ↑	LPIPS ↑
Ref.: $\text{dist}(x_i, x_j)$	0.376	0.558
Ref.: $\text{dist}(x_i, \mathbb{E}[x])$	0.307	0.623
LAM	0.148 ✗	0.326 ✗
+ StaDy	**0.335** ✓	**0.656** ✓

(ii) The latent representation captures motion. We next evaluate whether the regularized latent still captures the underlying motion. Using the learned encoder, we construct the latent representation from the target appearance together with motion and pose information from a source video, and condition the decoder on this latent together with the first frame of a target video. Because the trajectories are fully parameterized, we can render the exact ground-truth transfer – the target shape executing the source trajectory – and compare it against the generated frame using the L1 distance. Results are shown in Figure 5. See Appendix Fig. 10 for transfer examples. The regularized model substantially improves transfer performance while only mildly increasing reconstruction error. Suppressing appearance information therefore preserves the motion needed for transfer. With little appearance leakage, transferring motion to a new appearance should incur a similar error to reconstruction, placing the model near the identity diagonal. Consistent with this expectation, StaDy moves closer to the diagonal and achieves approximately 0.035 L1 error on both tasks.

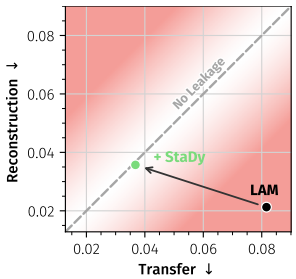

Figure 5: **Matched reconstruction/transfer error.** On the toy dataset, StaDy improves transfer with limited reconstruction cost, approaching the identity diagonal expected for an ideal leakage-free model.

5.2 Leakage Robustness on Human Heads

Avoiding the Capacity–Leakage Trade-off. We test robustness to appearance leakage as latent capacity grows on CelebV-HQ [35], a dataset of human-head videos. We compare StaDy against a reconstruction-only LAM, augmentation, β-VAE, and a CoMo-inspired temporal feature-difference control [32], with matched architectures and capacities. The latter uses our backbone rather than reproducing the full CoMo system. For each of 1,997 transfers, we extract $z = E_\phi(x_s, x_t)$ from a driving pair and apply it to a different recipient frame x'_s. ArcFace [7] similarity to x_t measures appearance leakage, while similarity to x'_s measures recipient identity preservation. MMPose landmark-displacement cosine similarity measures motion fidelity. Higher motion fidelity and preservation, together with lower leakage, indicate more faithful transfer.

Figure 6 extends the comparison across six latent capacities. At larger capacities, the LAM and alternative baselines lose motion fidelity and recipient identity, while StaDy maintains stronger transfer with less leakage. The plotted StaDy weights vary with capacity; Table 2 examines sensitivity to this choice. These results support our central motivation: in the reconstruction-only LAM, the bottleneck is the main constraint on appearance leakage, and loosening it lets the latent increasingly encode the target identity. Figure 1 illustrates this qualitatively: the baseline increasingly copies the driving identity, whereas StaDy better preserves the recipient's appearance.

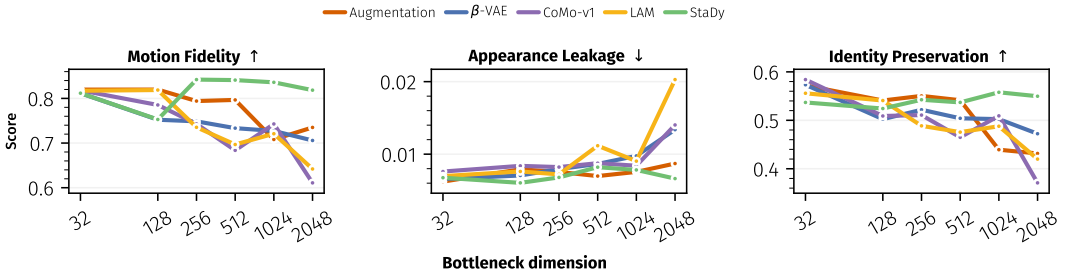

Figure 6: **StaDy mitigates capacity-induced leakage on human heads.** As latent capacity grows, the baselines lose motion fidelity and recipient identity while appearance leakage increases. StaDy maintains stronger motion transfer and identity preservation with less leakage.

Sensitivity to Regularization Strength. Table 2 shows that StaDy is stable across more than an order of magnitude in λ at high capacity. At 2048-D, increasing λ from 0.1 to 2 retains motion fidelity of 0.733–0.839, keeps leakage at 0.007–0.010, and preserves recipient identity at 0.468–0.554. Even $\lambda = 0.1$ halves leakage relative to the LAM (0.010 versus 0.020), while improving motion fidelity and identity preservation. Thus, the high-capacity benefit does not depend on a narrowly tuned weight. At $\lambda = 0.5$, increasing capacity from 128-D to 2048-D improves both motion fidelity (0.753 to 0.839) and identity preservation (0.524 to 0.554), with leakage remaining low (0.006 to 0.008). Strong regularization can overwhelm small bottlenecks: $\lambda \geq 1$ at 128-D gives near-zero motion fidelity, whereas $\lambda = 0.1$ retains motion at both capacities. Appendix F provides additional capacity–regularization analyses. We next test whether larger latents also benefit more complex open-world dynamics.

Table 2: **Regularization sensitivity on human heads.** At high capacity, StaDy retains motion and identity with low leakage across a broad range of weights. Strong regularization can suppress motion at small capacity. Bold marks the best result in each 2048-D column.

	Motion fidelity ↑		Leakage ↓		Preservation ↑	
λ	128-D	2048-D	128-D	2048-D	128-D	2048-D
0	0.819	0.642	0.008	0.020	0.541	0.420
0.1	0.785	0.733	0.007	0.010	0.518	0.468
0.5	0.753	**0.839**	0.006	0.008	0.524	**0.554**
1	0.018	0.818	0.007	**0.007**	0.488	0.550
2	−0.011	0.819	0.004	0.008	0.488	0.551

5.3 Scaling to Open-World Video

We next scale latent action learning beyond controlled shapes and human heads to diverse open-world videos. We train a LAM and StaDy on a larger-scale in-house dataset collected from internet videos depicting a wide variety of scenes, objects, and motions, with matched architecture, data, and compute. We evaluate motion transfer, downstream action recognition, and target-feature prediction from frozen latents, then test on actions whose labels are correlated with the target state. Unless stated otherwise, StaDy uses $\lambda = 0.2$ across capacities and benchmarks. Appendix E provides protocols and full sweeps.

Open-world motion transfer. Here we test whether larger latent actions improve motion transfer in diverse scenes without copying the driving appearance, assessing whether the controlled-setting benefits extend to open-world videos. We construct 2,000 motion-transfer trials from held-out videos in Koala [31]. Each trial extracts z from an adjacent-frame driving pair and applies it to a recipient frame from a different video. CoTracker [16] point-displacement cosine similarity measures how faithfully the output follows the driving motion. DINOv2 similarities to the driving target and recipient source measure appearance leakage and preservation, respectively. Table 3 shows comparable motion gains when increasing capacity from 96-D to 1536-D: 26% for the LAM and 24% for StaDy. However, the LAM's leakage increases by 21% and recipient preservation drops by 14%. With StaDy, both leakage and recipient preservation remain nearly constant. Thus, the transfer benefit of additional capacity extends to open-world videos without the accompanying increase in appearance copying observed for the LAM.

Larger latents improve downstream motion representations. We train attentive classification probes on frozen latent actions, aggregating transitions across each clip. SSv2 [13], Jester [23], FineGym99 [28], and Diving48 [19] cover human–object interactions, gestures, and fine-grained articulated motion. StaDy improves accuracy over the matched LAM at both capacities on every benchmark (Table 4). Increasing capacity raises its SSv2 accuracy from 19.72% to 34.95% and Diving48 accuracy from 10.69% to 18.08%, relative gains of 77% and 69%. On Diving48, the LAM changes little (9.21% to 9.73%), while StaDy makes substantially better use of the larger latent space. Together with the transfer results, these gains show that additional capacity supports richer motion representations while leakage remains controlled. Appendix E provides additional evaluation details.

Table 3: **Motion transfer on open-world videos.** Both methods gain motion fidelity with capacity, but StaDy limits leakage growth and preserves the recipient's appearance. Δ is the relative change from 96-D to 1536-D.

Metric	Method	96-D	1536-D	Δ
Fidelity ↑	LAM	**0.292**	**0.369**	+26% ✓
	+StaDy	0.290	0.360	+24% ✓
Leakage ↓	LAM	**0.070**	0.085	+21% ✗
	+StaDy	0.071	**0.072**	+1% ✓
Preservation ↑	LAM	**0.673**	0.579	−14% ✗
	+StaDy	0.672	**0.681**	+1% ✓

Table 4: **Larger latent actions improve motion understanding.** Top-1 accuracy (%) on frozen latents. StaDy improves on the LAM at both capacities across all benchmarks.

Benchmark	Method	96-D	1536-D
SSv2 [13]	LAM	17.10	27.05
	+StaDy	19.72	**34.95**
Jester [23]	LAM	59.07	80.89
	+StaDy	64.71	**84.18**
FineGym99 [28]	LAM	24.15	41.90
	+StaDy	28.54	**45.16**
Diving48 [19]	LAM	9.21	9.73
	+StaDy	10.69	**18.08**

Target leakage and endpoint-correlated dynamics. To test leakage directly in the representation, we freeze each encoder and train identical three-layer MLPs to predict the target frame's DINOv2 features from z alone. The probes train on the in-house corpus and are evaluated on 5,000 held-out Koala clips. Lower predicted-to-target feature similarity indicates less target information accessible to the probe. In Table 5, similarity grows substantially with capacity for the LAM, but less for StaDy at $\lambda = 0.2$ and only slightly at $\lambda = 0.4$. This provides representation-level evidence consistent with the reduced appearance leakage observed in motion transfer.

We also examine SSv2 directional and opening/closing subsets, where action labels can predict aspects of the endpoint. Table 6 uses the same attentive-probe protocol as the full benchmark. At 1536-D, StaDy improves both subsets, including opening/closing accuracy from 24.90% to 34.42%. Thus, reducing target predictiveness need not discard useful dynamics even when the action and endpoint are correlated: the learned latents retain discriminative action information alongside reduced target-feature accessibility. Appendix E.1 discusses the scope of this test.

Table 5: **StaDy reduces target information accessible from latent actions.** Lower DINOv2 cosine similarity indicates less accessible target information. Δ is the percentage increase from 96-D to 1536-D, computed before rounding.

Method	96-D ↓	1536-D ↓	Δ ↓
LAM	0.24	0.29	22%
StaDy ($\lambda = 0.2$)	0.24	0.27	11%
StaDy ($\lambda = 0.4$)	**0.23**	**0.24**	**3%**

Table 6: **StaDy retains useful endpoint-correlated dynamics.** SSv2 top-1 accuracy (%). At high capacity, StaDy improves recognition of both action subsets.

Subset	Method	96-D	1536-D
Directional	LAM	**36.72**	46.02
	+StaDy	36.28	**53.66**
Open/close	LAM	12.99	24.90
	+StaDy	**18.40**	**34.42**

More data and explicit regularization. We hypothesize that StaDy learns useful motion more efficiently from limited training data because less latent capacity is spent encoding appearance. Table 7 compares StaDy with LAM at equal data budgets. With only 1% of the training data, even 96-D latents benefit: identity preservation improves by 5.8% and motion error falls by 15.5%. At 1536-D, these gains grow to 58.2% and 42.2%, respectively, showing that explicit regularization is particularly valuable when a large bottleneck must learn from limited data. The advantage persists with 10% data: motion error decreases by 8.4% at 96-D and 45.0% at 1536-D, alongside identity gains of 1.2% and 17.9%. Thus, more training data does not eliminate the benefit of regularization. Better preservation and more faithful motion at the same data budget support improved training-data efficiency at both capacities; Appendix E.2 discusses the scope of this interpretation.

Table 7: **StaDy improves motion transfer at limited data budgets.** Relative changes compare StaDy with LAM at equal training-data fractions; gains are particularly strong at high latent capacity. Bold marks improvements over LAM.

Latent dim.	Data	Identity ↑			Motion error ↓		
		LAM	+StaDy	Δ	LAM	+StaDy	Δ
96-D	1%	0.2288	**0.2421**	+5.8%	0.0888	**0.0750**	−15.5%
	10%	0.5105	**0.5168**	+1.2%	0.0391	**0.0358**	−8.4%
1536-D	1%	0.1394	**0.2206**	+58.2%	0.1494	**0.0864**	−42.2%
	10%	0.4387	**0.5171**	+17.9%	0.0682	**0.0375**	−45.0%

6 Conclusion

We studied a fundamental failure mode of latent action models: because the encoder sees both frames, the latent can encode target appearance instead of the transition. Existing methods mitigate this with capacity bottlenecks or augmentations, but both act indirectly and trade expressivity for disentanglement. We introduced StaDy, a likelihood-matching regularizer that reduces latent-only target predictiveness while reconstruction encourages retention of useful dynamics. Controlled synthetic experiments verify motion transfer against exact targets, while human-head capacity sweeps demonstrate reduced appearance leakage. On diverse open-world videos, increasing latent dimensionality improves motion recognition across multiple benchmarks and motion-transfer fidelity while keeping appearance leakage nearly constant. Direct target-feature probes support reduced leakage in the representation, and data-fraction sweeps support improved training-data efficiency at both low and high latent capacity. StaDy thus offers a practical path toward richer, more transferable latent actions learned from diverse unlabeled videos.

Acknowledgements

This project has been supported by the Horizon Europe project ELLIOT (GA No. 101214398), the project "GeniusRobot" (01IS24083) funded by the Federal Ministry of Research, Technology and Space (BMFTR), the BMWE ZIM project "conIDitional LoRA" (No. KK5785001LO4), and the German Federal Ministry for Economic Affairs and Energy within the project "NXT GEN AI METHODS – Generative Methoden für Perzeption, Prädiktion und Planung". This project was also supported by Bayer AG and the German Academic Exchange Service (DAAD) under the Konrad Zuse School of Excellence for Reliable AI (RelAI). The authors gratefully acknowledge the Gauss Center for Supercomputing for providing compute through the NIC on JUWELS/JUPITER at JSC and the HPC resources supplied by the NHR@FAU Erlangen.

References

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2017.

[2] Tewodros Ayalew, Matthew Jeung, Samuel Wheeler, Xiao Zhang, Andre de la Cruz Arce, Kaylene Stocking, Michael Maire, and Matthew R. Walter. CLAW: Learning continuous latent action world models via adversarial latent regularization. *arXiv preprint arXiv:2606.04130*, 2026.

[3] Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

[4] Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai, 2024.

[5] Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[6] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

[7] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, October 2022.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[9] Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell. Imitating latent policies from observation. In *International Conference on Machine Learning (ICML)*, pages 1755–1763, 2019.

[10] Frank Fundel, Malek Ben Alaya, Thomas Ressler-Antal, Stefan Andreas Baumann, and Björn Ommer. What moves? localized motion representations for compositional scene control. In *European Conference on Computer Vision*, pages 297–315. Springer, 2026.

[11] Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. In *International Conference on Machine Learning (ICML)*, 2025.

[12] Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. *arXiv preprint arXiv:2601.05230*, 2026.

[13] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Freund, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, pages 5842–5850, 2017.

[14] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017.

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[16] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pages 18–35. Springer, 2024.

[17] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[18] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2026.

[19] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 513–528, 2018.

[20] Anthony Liang, Pavel Czempin, Matthew M Hong, Yutai Zhou, Jingzhen Wang, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

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

[22] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision (ECCV)*, 2022.

[23] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, April 2023. arXiv:2304.07193 [cs].

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

[26] Thomas Ressler-Antal, Frank Fundel, Malek Ben Alaya, Stefan Andreas Baumann, Felix Krause, Ming Gui, and Björn Ommer. DisMo: Disentangled motion representations for open-world motion transfer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[27] Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations (ICLR)*, 2024.

[28] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. FineGym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2020.

[29] Yihong Sun, Hao Zhou, Liangzhe Yuan, Jennifer J. Sun, Yandong Li, Xuhui Jia, Hartwig Adam, Bharath Hariharan, Long Zhao, and Ting Liu. Video creation by demonstration, 2024.

[30] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[31] Qiuheng Wang, Yukai Shi, Jiarong Ou, Rui Chen, Ke Lin, Jiahao Wang, Boyuan Jiang, Haotian Yang, Mingwu Zheng, Xin Tao, Fei Yang, Pengfei Wan, and Di Zhang. Koala-36m: A large-scale video dataset improving consistency between fine-grained conditions and video content, 2025.

[32] Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. CoMo: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025.

[33] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025.

[34] Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky T. Q. Chen. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*, 2023.

[35] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. Celebv-hq: A large-scale video facial attributes dataset, 2022.

A Broader Societal Impact

This work advances latent action learning, with potential applications in video generation, analysis, and action modeling for embodied systems. At the same time, it may enable downstream uses that could be misused, such as unintended motion transfer. Although our experiments are limited in scope, the method is general and could be integrated into larger systems. We emphasize transparency and encourage responsible use.

B Limitations

Our evaluation uses a flow-matching decoder; transfer to other generative frameworks remains untested. Open-world video broadens the empirical scope, but transfer remains pairwise and recognition probes aggregate pairwise latents rather than testing long-horizon generation. The information floor in (3) assumes action recovery from z, which reconstruction alone does not guarantee. Goal-reaching actions, newly revealed or occluded content, and exogenous changes can require useful target-predictive information that excessive regularization may suppress. These are structural risks, distinct from the observed low-capacity optimization failures in Appendix F. Closed-loop control and robotic planning remain unevaluated.

C Scope of the Action-Information Floor

Assume $a = g(z)$ almost surely and finite mutual information. Then $I(a; x_t \mid z) = 0$, so the chain rule gives

$$I(z; x_t) = I(z, a; x_t) = I(a; x_t) + I(z; x_t \mid a). \tag{8}$$

Nonnegativity of the last term yields (3). Equality requires no further target information in z given a; $z = a$ is an idealized example if the action is identifiable from the frame pair and sufficient for the task. Ambiguous actions or reconstruction of newly revealed content can prevent attaining this floor. Action recovery only from (x_s, z) does not imply the unconditional bound. For example, if $a = x_t$ and x_s are independent fair bits, $z = a \oplus x_s$ satisfies $I(z; x_t) = 0$ although a is recoverable from (x_s, z) and $I(a; x_t) = 1$ bit.

Exact independence is compatible with action preservation when $I(a; x_t) = 0$. One sufficient setting is $x_t = T_a(x_s)$ with a independent of x_s and every T_a preserving the source-state distribution. Real transitions need not satisfy this: an opening action can predict an open endpoint. Our velocity-field penalty is a practical surrogate balanced against reconstruction, not a direct estimator of excess mutual information or a guarantee of reaching the floor.

D Implementation Details

Table 8 summarizes the main architectural and optimization hyperparameters used in the original synthetic and CelebV-HQ experiments; the open-world evaluation protocol is described in Appendix E. Unless stated otherwise, the frame generator is a JiT-S transformer with 12 transformer blocks and hidden width 384. The model is trained as a pixel-space flow-matching model. We sample noise $\mathbf{x}_0 \sim \mathcal{N}(0, I)$, data $\mathbf{x}_1$, and interpolation time $\tau \in [0, 1]$, construct

$$\mathbf{x}_\tau = (1 - \tau)\mathbf{x}_0 + \tau \mathbf{x}_1, \tag{9}$$

and train the network with the corresponding velocity target

$$\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_0. \tag{10}$$

In the implementation, the generator is parameterized in pixel space and reparameterized to the velocity prediction used by the flow-matching objective.

When an image-based motion encoder is used, it has the same ViT-S backbone size as the frame generator: 12 transformer blocks and hidden width 384. Source and target frames are first embedded with a frozen DINOv2-B frame embedder. The source-frame patch tokens, target-frame patch tokens, and learned motion-query tokens are then concatenated into one sequence and processed with full self-attention. After the transformer stack, only the query-token outputs are kept; all

Table 8: Synthetic/CelebV-HQ architecture and optimization parameters. The motion bottleneck is controlled by the number of motion-query tokens N_q; each query token is projected to 32 dimensions.

Component	Parameter	Value
Frame embedder	Architecture	DINOv2-B
	Status	Frozen
	Role	Embeds source/target frames for the motion encoder
Frame generator	Architecture	JiT-S transformer
	Blocks	12
	Hidden width	384
	Objective	Pixel-space flow matching
	Prediction	$\mathbf{x}_0$ prediction reparameterized to $\mathbf{v}$
	Conditioning	Source frame and motion tokens
Motion encoder	Architecture	ViT-S transformer, when used
	Blocks	12
	Hidden width	384
	Input tokens	DINOv2-B source patches, DINOv2-B target patches, and query tokens
	Attention	Full self-attention over concatenated tokens
	Output tokens	Query tokens only
	Query width	32
	Queries N_q	$\{1, 4, 8, 16, 32, 64\}$
	Bottleneck size	$\{32, 128, 256, 512, 1024, 2048\}$
Optimization	Batch size	128
	Learning rate	10^{-4}
	Warmup	5000 steps

residual frame tokens are discarded. Each query token is projected to 32 dimensions. Thus, the total motion bottleneck is $32N_q$, where N_q is the number of query tokens. For example, $N_q = 1$ gives a 32-dimensional bottleneck, while $N_q = 64$ gives a 2048-dimensional bottleneck. The principal CelebV-HQ regularization weight is $\lambda = 0.5$, with additional weights evaluated explicitly in the sensitivity analysis. The open-world comparisons use the fixed setting reported as $\lambda = 0.2$.

D.1 Toy Dataset

We additionally evaluate on a fully parametric toy dataset in Section 5.1. This dataset is generated on the fly during training and evaluation. Each dataloader sample is produced by first sampling a set of appearance, pose, and motion parameters, and then deterministically rendering the corresponding video frames. As a result, every source–target pair is fully specified by known latent factors, rather than by an external video file.

The appearance parameters define the object and the scene appearance. We sample the shape type from circles, squares, triangles, and ellipses; the object size; the ellipse aspect ratio; two object colors; a background color that is sufficiently separated from the object colors; and an appearance mode. The appearance mode controls the fill pattern of the object and is sampled from solid color, linear gradient, angular gradient, and stripes. These parameters are static across the source and target frames, except in the optional framewise-appearance setting, where the fill mode can change after the first frame.

The pose and motion parameters determine where the object appears and how it moves. The initial center position is sampled inside a safe region so that the object remains visible. The initial rotation angle and angular velocity are also sampled. The dataset supports several motion families: linear motion, bouncing motion, accelerated motion, and circular motion. Linear motion uses a fixed velocity; bouncing motion reflects the velocity at the image boundary; accelerated motion uses a sampled acceleration; and circular motion samples an orbit center, radius, angular speed, and phase. Given these parameters, the dataset computes the framewise center positions and rotations, and renders each frame deterministically with OpenCV.

The dataloader returns the rendered video $\mathbf{x} \in [-1, 1]^{T \times H \times W \times C}$ together with the generating parameters. The returned appearance information contains one-hot shape type and appearance mode,

normalized size and aspect ratio, object colors, and background color. The returned pose information contains normalized framewise positions and sine/cosine rotation features. The returned motion information contains framewise position deltas and angular-motion features. In the compact setting, these motion features are normalized aggregate horizontal velocity, vertical velocity, angular velocity, and speed.

D.2 Latent Construction for the Toy Dataset

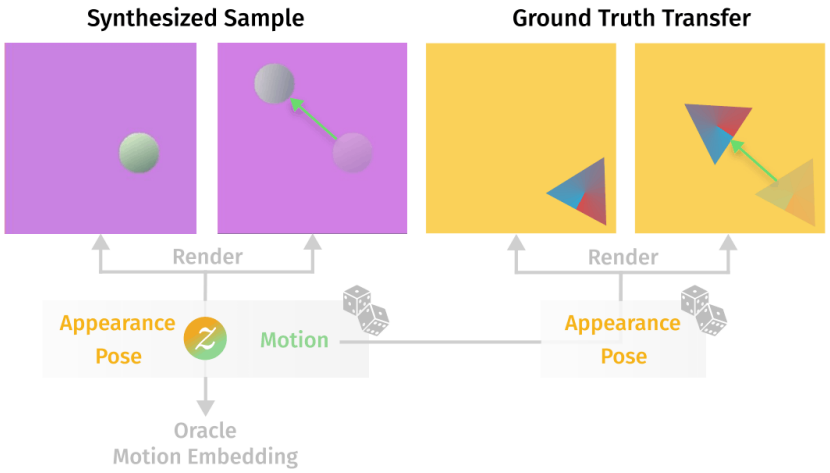

Figure 7: **Synthetic data generation and ground-truth motion transfer.** During data generation, we sample appearance, pose, and motion parameters to deterministically synthesize source and target frames. Ground-truth motion-transfer targets are then constructed by applying the same motion parameters to newly sampled appearance and pose parameters.

For the case where the latent representation is constructed rather than fully learned, we do not require an image-based motion encoder. Since the dataset is generated from known parameters, the parameters themselves form perfectly predictive and interpretable embeddings. We use this property to construct an adversarial test suite for disentanglement: the decoder receives a source frame and an artificial "motion" embedding whose contents can be controlled exactly.

In practice, we construct three distinct embeddings: an appearance embedding, a pose embedding, and a motion embedding. The appearance embedding contains all static visual information shared across the source and target frames, including the shape type, appearance mode, normalized size, aspect ratio, foreground colors, and background color. The pose embedding contains target-frame pose information, namely the target position and target rotation of the object. The motion embedding contains the transformation from source to target, namely the change in position and change in rotation.

Each feature group is first passed through its own MLP:

$$\mathbf{z}_{\text{app}} = f_{\text{app}}(\mathbf{a}), \tag{11}$$
$$\mathbf{z}_{\text{pose}} = f_{\text{pose}}(\mathbf{p}), \tag{12}$$
$$\mathbf{z}_{\text{mot}} = f_{\text{mot}}(\mathbf{m}), \tag{13}$$

where $\mathbf{a}$ denotes appearance parameters, $\mathbf{p}$ denotes pose parameters, and $\mathbf{m}$ denotes motion parameters. The selected embeddings are then concatenated token-wise to form the artificial conditioning sequence supplied to the decoder:

$$\mathbf{z} = \text{concat}\left(\mathbf{z}_{\text{mot}}, \mathbf{z}_{\text{app}}, \mathbf{z}_{\text{pose}}\right). \tag{14}$$

The decoder is conditioned on this sequence and on the source frame exactly as in the image-based motion-transfer setting.

This construction deliberately creates a strong adversarial setting. The pose and appearance embeddings contain enough information to reconstruct the target frame almost perfectly: appearance

specifies what should be rendered, and target pose specifies where and how it should be rendered. However, a properly disentangled motion-transfer model should rely only on the motion component when transferring dynamics to a new source frame. Therefore, leakage from the appearance or pose embeddings is directly measurable and interpretable in this setting.

E Additional Open-World Evaluations

Protocol and regularization convention. The models trained on the in-house corpus are evaluated on held-out Koala clips. Each model is evaluated on 2,000 motion-transfer trials using adjacent-frame driving pairs from held-out Koala videos and recipient frames from different videos, without random-frame sampling. Motion fidelity is CoTracker displacement cosine; DINOv2 similarities to the driving target and recipient source measure leakage and preservation. The principal comparisons use one fixed regularization setting across capacities and downstream tasks. Figure 8 extends the comparison to the full capacity–weight sweep.

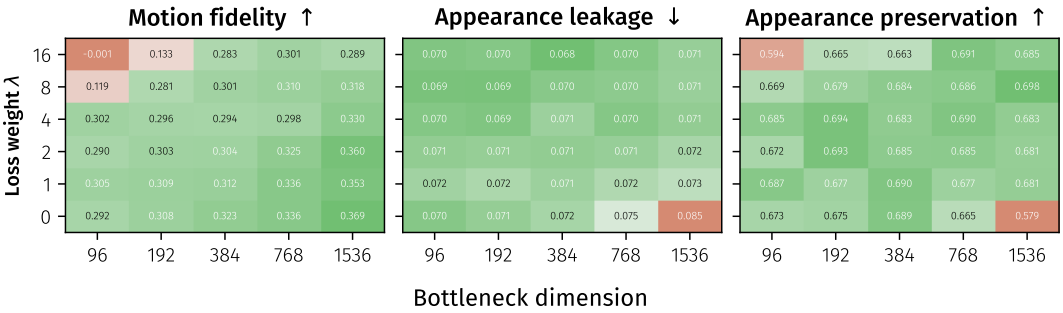

Figure 8: **Full Koala capacity–regularization sweep.** Models trained on the in-house corpus, 2,000 adjacent-frame transfers per model. The figure labels the regularization axis λ using the stored integer `regweight` values. The row labeled 2 reproduces the transfer results called $\lambda = 0.2$ in Table 3; this is a numerical correspondence, and the underlying scale conversion is not documented in the local sources. Each panel has its own color scale.

E.1 Recognition and endpoint-correlated actions

Frozen motion tokens are evaluated with the same learned class-token attentive-probe architecture on SSv2, Jester, FineGym99 and Diving48. Probes aggregate transitions across a clip.

Endpoint-correlated subsets. SSv2 directional pairs (left-to-right versus right-to-left, including analogous pushing/pulling pairs) and opening/closing pairs test actions whose labels can predict the endpoint. Table 6 uses the same attentive-probe protocol as the full-dataset evaluation. At 1536-D, StaDy improves both subsets while reducing target-feature decodability. This supports retention of useful dynamics under finite regularization; endpoint cues can also help classification, so it does not prove that all action information is retained.

E.2 Training-data fraction and explicit regularization

Garrido et al. [12] hypothesize that training-data complexity discourages endpoint copying. To examine the interaction with explicit regularization, Table 7 compares LAM and StaDy at training-data fractions of 1% and 10%, for 96-D and 1536-D latents. Identity scores are higher-is-better and motion errors are lower-is-better. The relative changes compare StaDy with LAM at each fixed data fraction and latent dimensionality, computed as $(\mathrm{StaDy} - \mathrm{LAM})/\mathrm{LAM}$.

The results and their interpretation are reported in Section 5.3. The sweep varies dataset size and potentially diversity together, so it does not isolate a causal effect of diversity. Better performance at equal data budgets is consistent with improved data efficiency; quantifying the data savings at matched performance requires a denser sweep with controlled training compute.

F Regularization Sensitivity and Baseline Sweeps

Table 2 evaluates regularization on 1,997 CelebV-HQ transfers with MMPose motion fidelity and ArcFace appearance metrics. At 2048-D, $\lambda = 0.1$–2 retains motion fidelity while suppressing leakage. At 128-D, $\lambda \geq 1$ produces near-zero motion fidelity. These observed optimization failures are separate from the structural trade-off for endpoint-correlated dynamics.

Baseline selection and metric differences. Figure 6 compares six capacities, with StaDy weights selected separately at each. All models are evaluated at step 400k. The plotted StaDy weights are $\lambda = 0.1$ at 32-D, 0.5 at 128–1024-D, and 1 at 2048-D, whereas Table 2 reports the sensitivity to different weights at 128-D and 2048-D. Figure 9 instead uses DINOv2 appearance metrics and unequal training checkpoints; it illustrates the observed capacity–regularization interaction but is not a matched-checkpoint replacement for the baseline comparison.

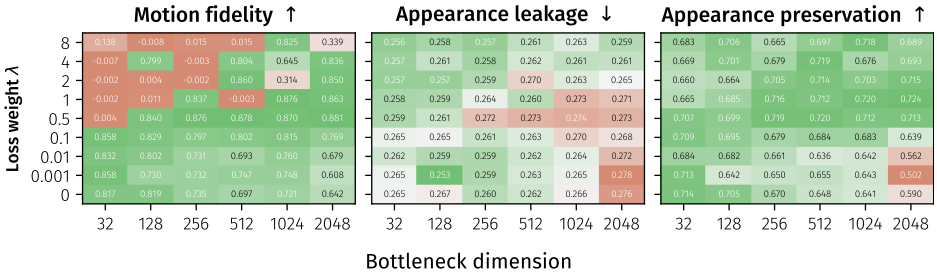

Figure 9: **Face-specific CelebV-HQ capacity–regularization sweep.** Heatmap showing MMPose motion fidelity and DINOv2 appearance leakage/preservation. Table 2 instead uses ArcFace appearance metrics. The heatmap evaluates the unregularized LAM at step 400k and regularized models at step 950k, so its values are not a directly matched-checkpoint reproduction of Table 2.

G Design Ablations

Ablation of gradient flow and architectural choices. Table 9 evaluates the gradient-flow and architectural choices using the latent encoder from Section 5.1. The full model gives the best transfer CLIP similarity and L1 error. Removing either stop-gradient or the motion branch degrades transfer. Removing the encoder stop-gradient or both stop-gradients improves reconstruction while worsening transfer, illustrating why reconstruction quality alone does not identify a useful latent action representation.

Table 9: **Design ablations.** Removing stop-gradients or the motion branch degrades transfer. Removing the encoder stop-gradient or both stop-gradients improves reconstruction at the cost of transfer.

Setting	Transfer		Reconstruction	
	CLIP ↑	L1 ↓	CLIP ↑	L1 ↓
StaDy	**0.705**	**0.066**	0.735	0.032
w/o enc. stop-grad	0.680	<u>0.075</u>	<u>0.740</u>	<u>0.023</u>
w/o dec. stop-grad	0.651	0.080	0.735	0.078
w/o motion branch	<u>0.697</u>	0.080	0.731	0.031
w/o stop-grads	0.645	0.076	**0.747**	**0.020**

H Further Analysis

H.1 Qualitative Transfer Examples on Toy Data

We next present qualitative examples of our regularized model on the oracle toy dataset; see Fig. 10.

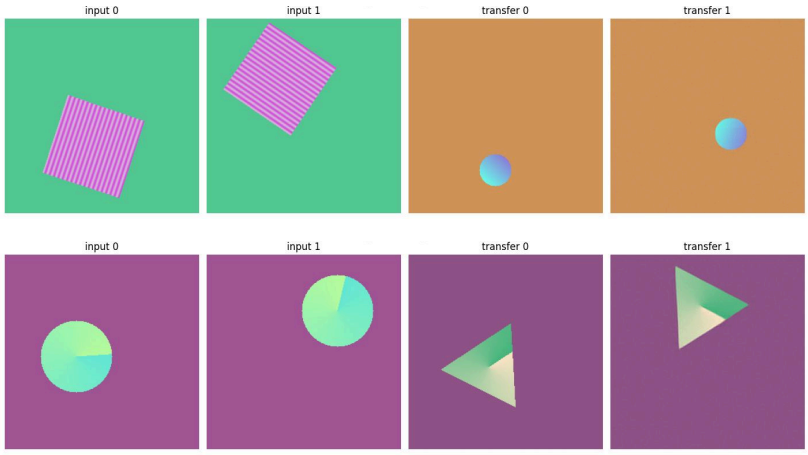

Figure 10: **Qualitative transfer examples.** The first two columns show the driving source and target frames. The third column shows the transfer source, and the fourth shows the generated frame after applying the extracted motion.

H.2 Decoder Reliance on Motion and Appearance

To analyze how the decoder changes its behavior under our regularization term, we again consider the minimal toy dataset setting in which source and target frames are generated from randomly sampled ground-truth parameters. Each rendered frame contains a single shape, and the underlying generative factors fully specify the scene. These factors include appearance, target pose, and source-to-target motion parameters. Since this setting provides direct access to the true generative parameters, it allows us to precisely interpret which information sources the decoder relies on.

We group the parameters into two categories: appearance parameters, such as shape type, color, and target position; and motion parameters, such as displacement and angular velocity. During training, we extract the decoder attention assigned to each parameter group and visualize the resulting attention maps. The results are shown in Figure 11.

The baseline model, trained only with the standard reconstruction objective, relies predominantly on the appearance stream, while the motion stream remains less utilized. In contrast, the model trained with our regularization term shifts its attention toward motion-related information, with substantially reduced reliance on appearance-based features. This suggests that our regularization biases the decoder toward modeling state transitions between frames rather than directly encoding the target state.

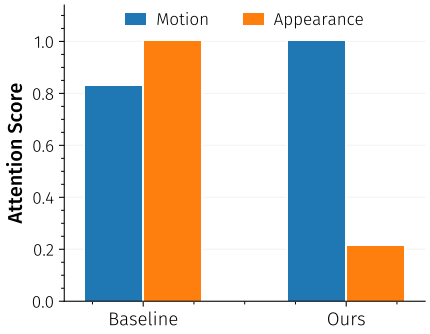

Figure 11: Decoder attention assigned to different parameter groups in the toy setting. The baseline model primarily attends to appearance information, whereas our regularized model shifts attention toward motion-related parameters.

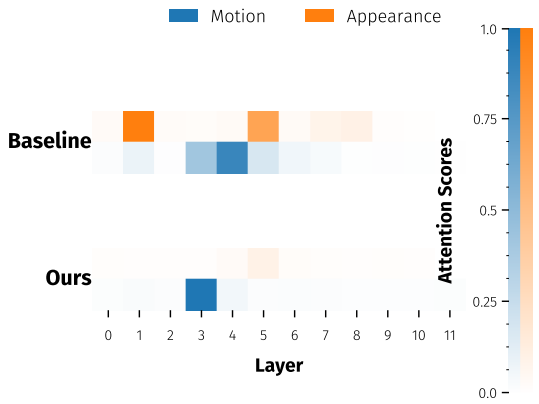

Figure 12: Relative contribution of different parameter groups to the decoder output. Our regularization increases the use of motion information and suppresses reliance on appearance features.

I Computational Resources for Experiments

We use NVIDIA H200 141GB accelerators. The experiments require approximately 25,500 H200 GPU-hours in total, distributed as follows:

- Open-world experiments: approximately 20,000 H200 GPU-hours across all 30 open-world ablations.
- CelebV-HQ experiments: approximately 5,000 H200 GPU-hours.
- Toy experiments: approximately 500 H200 GPU-hours.

GPU-hours count accelerator time summed across devices.

J Licenses of Used Assets

CelebV-HQ. The dataset's official agreement permits non-commercial research and prohibits commercial exploitation and redistribution. Internal copies are permitted at a single site within the same organization. The underlying internet videos remain the property of their respective rights holders.

Koala. The Koala-36M license permits non-commercial research, testing and validation. Commercial use requires a separate agreement with Kuaishou Technology, and redistribution must retain the copyright notice, conditions and disclaimer.

Something-Something v2 and Jester. Both datasets are governed by Qualcomm Technologies' Data License Agreement for Research Use. Access requires accepting that agreement. It restricts use to research and prohibits commercial use and redistribution of the licensed data.

FineGym. The official project page releases the annotations under Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0). This annotation license does not confer rights to the underlying YouTube videos, which remain subject to their respective rights and access conditions.

Diving48. The dataset originates from the official Diving48 project. We could not verify an explicit dataset license from the accessible primary sources and therefore do not assign it an open-data license.

In-house internet-video corpus. This corpus is collected from internet videos. No corpus-wide license is specified here; the source videos remain subject to their respective rights holders' terms. Public availability alone does not establish reuse or redistribution rights.

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 claims made in the abstract and introduction are accurate and aligned with the contributions and scope of the paper. They clearly state the novelty of our proposed approach and the empirical findings that support our conclusions. Limitations are also appropriately acknowledged to ensure the claims do not overstate the generality or applicability of our results.

   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: the paper acknowledges key limitations, including generalization challenges.

   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: [N/A]

Justification: The paper does not contain any theoretical results; it is entirely empirical. Therefore, this question is not applicable.

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 introduce a method that addresses key limitations in an existing modeling paradigm and provide sufficient detail to reproduce the main experimental results. We describe the model architecture, training procedure, datasets, and evaluation metrics in a clear and complete manner, and specify the key hyperparameters and implementation choices underlying our experiments. This level of detail is intended to enable independent verification of our results and conclusions.

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 include anonymized code and detailed reproduction instructions in the supplementary material to ensure faithful replication of our main experimental results.

   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: The main paper clearly describes the experimental setup required to interpret the results, including dataset splits, model architectures, and key hyperparameters. Complete training and evaluation details—such as exact hyperparameter values, optimizer configurations, and selection criteria—are provided in the supplementary material, ensuring reproducibility while keeping the main text concise.

   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: Due to the computational cost of training and evaluating the proposed models, running multiple independent trials to report error bars or statistical significance was not feasible. While such measures are important for assessing robustness, the scale of the experiments made repeated runs impractical given the available resources.

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 report compute details including GPU type, memory, and training time in the supplementary material.

   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: This work complies with the NeurIPS Code of Ethics. It does not involve human subjects, crowdsourcing, or sensitive data. The study emphasizes transparency and reproducibility, and all experimental procedures follow standard ethical research practices.

   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: The paper discusses both the potential benefits of the method and possible avenues for misuse in the supplementary material.

    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: [Yes]

    Justification: We aim to release our models under a license that requires responsible use, and we mention that in the supplementary material.

    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: The licenses, terms of use, and proper credits for all external assets (e.g., datasets, code) used in this work are explicitly provided in the supplementary material.

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: Anonymized code and documentation are provided in the supplementary material, and the models will be released publicly upon acceptance.

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: This work does not involve crowdsourcing or human subjects. All experiments are conducted on publicly available or synthetic datasets, with no human participation in data collection or evaluation. Therefore, this question is not applicable.

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: The paper does not involve any research with human subjects or the use of crowdsourced data. Therefore, Institutional Review Board (IRB) approval or equivalent review is not applicable.

    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: Large language models (LLMs) are not used as a core or non-standard component of the proposed methodology. The work focuses on a motion modeling approach developed independently of LLM-based techniques.

    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.