FLOMO: 3D Scene Flow as a World-Action Model Intermediate for Learning from Human Video

Anonymous Authors

Abstract: Pretrained video generation models distill internet-scale motion priors into robot policies, but full video is a redundant signal for action prediction. We argue that dense 3D motion is the medium through which video pretraining best transfers to robot actions, and that predicting motion rather than video affords better sample-efficiency for action prediction. FLOMO is a 5B video-generation model fine-tuned to predict 3D scene-flow and action tokens. Rendering scene flow as RGB and encoding it into the same latent space as video enables a pretrained video generation backbone to efficiently adapt to predict 3D scene flow. Co-trained on action-free human video and action-labeled robot teleoperation data, our model benefits from human-video pretraining and generalizes to unseen objects, motion primitives, and tasks. Removing the scene flow prediction objective degrades out-of-distribution generalization, while adding a video prediction target alongside motion lowers policy success rather than improving it. When motion, video, and action are predicted jointly, action tokens attend far more to motion than to video, suggesting that predicting video wastes model capacity. Together, these results indicate that motion mediates the transfer from video pretraining to action. More information can be found at flomo-wam.github.io.

Keywords: World-Action Models, Scene Flow, Video Pretraining

1 Introduction

Pretrained video generation models can be used to distill internet-scale motion priors into robot policies. Transfer from video to action is typically attributed to information overlap between RGB videos and the actions that produce them: a model that predicts the world's next pixels has implicitly learned the dynamics that link action to outcome [1, 2, 3]. Video, however, is a far higher-entropy signal than the action it implies. Videos that differ in texture, lighting, material, and background depict the same underlying motion and imply the same action, so a world-action model (WAM) trained to predict pixels spends capacity on appearance that action prediction never uses [4, 5, 6].

WAMs vary on how tightly they couple the video backbone to the action head, but they share a prediction target: future pixels. We question this assumption and ask: *which prediction target lets video pretraining transfer to action most directly?* We argue that the answer is 3D motion. Given the current observation, the action that produces a future state depends on how the scene transformed rather than on appearance and texture. Predicted frames that differ only in lighting or background share the same motion and imply the same action. Predicting motion rather than pixels hence targets a lower-entropy signal and incorporates this invariance instead of forcing the model to learn it, leaving the action head a tighter information bottleneck it can decode directly.

Our insight is that a pretrained video VAE already provides a latent space for motion. We represent motion as the 3D displacement of each tracked scene point over time in a canonical frame independent of camera motion, otherwise known as *3D scene flow*, and render this vector field as an RGB image. Because the VAE encodes this rendered scene flow into the same latent space as video, scene-flow tokens require no new tokenizer; we keep the VAE fixed and fine-tune only the backbone to predict motion.

We instantiate this in FLOMO, a Wan 2.2 5B DiT that, given an image and language instruction, flow-matches 3D scene-flow and action tokens, with scene flow encoded into the same video latent space the model was pretrained on. We co-train on action-free human video and action-labeled robot teleoperation data, and find that predicting motion alone improves both in-distribution policy success and out-of-distribution generalization, relative to predicting video or predicting video and motion jointly.

Our contributions are:

1. **Motion as a WAM prediction target (§3).** We argue that WAMs should predict motion rather than video: motion is a low-dimensional, texture-invariant target an action head can decode directly, and the priors of video pretraining are carried by the pretrained backbone, not by the prediction target. We realize this by rendering 3D scene flow as an RGB image and encoding it with the pretrained video VAE, so scene-flow tokens share the video latent space and a single backbone flow-matches motion and action without a new tokenizer.
2. **Motion is the most effective prediction target for policy learning (§4.2, §4.3).** Across real-robot manipulation, predicting motion yields higher in-distribution success and out-of-distribution generalization than predicting video or jointly predicting video and motion; adding a video target interferes with action learning rather than complementing it.
3. **Generalization from human video (§4.3).** Our model generalizes to objects, motion primitives, and tasks seen only in human data.

2 Related Work

2.1 Learning Robot Policies from Action-Free Videos

Action-labeled robot demonstrations are expensive, while large-scale action-free videos of humans, robots, and objects are abundant. One line of work uses passive video to pretrain visual representations, rewards, or value functions ([7], [8], [9], [10]). These methods improve data efficiency but do not learn explicit predictive models of future dynamics.

Another line of work learns latent actions or transition variables from unlabeled videos through self-supervised objectives such as future prediction, frame consistency, or reconstruction ([11], [12], [13]). While these methods reduce reliance on action labels, their latents may encode visual information useful for reconstruction but unnecessary for control.

Finally, work on learning from human videos studies cross-embodiment alignment or explicit retargeting to transfer human demonstrations to robots ([14], [15], [16], [17], [18]). These works focus on bridging the human-robot embodiment gap; our work instead focuses on the prediction target inside of a WAM.

2.2 Motion and Geometry-Aware Video Prediction

Raw pixel prediction is an expensive and indirect objective for learning physical dynamics, motivating methods that introduce motion or geometric intermediates. Some use 2D motion as an interface between action-free video and robot actions, predicting future image-point trajectories and conditioning policies on them ([4], [5]). Others use flow or motion fields as an interface linking video prediction and manipulation ([19], [6], [20], [21]), or separate motion and temporal reasoning from appearance and camera movement in video generation ([22]).

Closest to our representation choice are methods that model dynamic scenes with compact trace spaces, 3D scene flow, or point-track representations ([23], [24], [25]). These works show that 3D motion fields are a natural substrate for future dynamics prediction. Rather than building a separate motion space, we render scene flow into a pretrained video-generation latent space.

2.3 World Action Models

Pretrained video-generation models provide motion priors for robot learning because they are trained on large-scale videos of physical interactions and object motion. Recent work uses these models

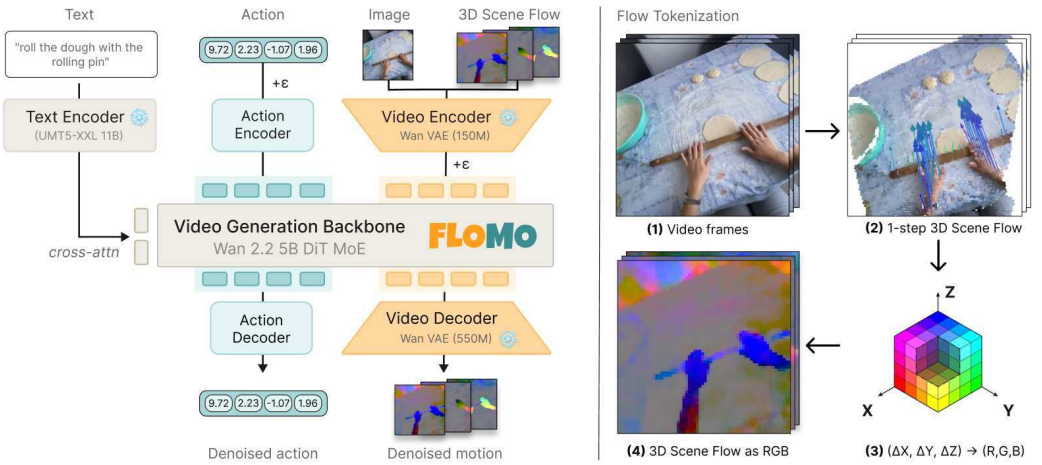

Figure 1: Given an image and text prompt, FLOMO jointly predicts dense 3D motion (scene flow) and robot actions using a pretrained video generation model using flow-matching. To tokenize scene flow, we transform instantaneous 3D velocities into RGB videos that are encoded by the pretrained video backbone's video encoder.

for control by coupling future video prediction with action generation. One family generates task-conditioned robot videos and then recovers or co-generates actions through inverse dynamics models or latent actions ([26], [27], [1], [2]). These approaches preserve the pretrained video model, but deployment is expensive and performance depends heavily on video fidelity. A second family avoids explicit video rollout at test time by extracting intermediate latents from a pretrained video backbone and training an action head on top ([28], [29], [30]). This reduces inference cost, but action prediction remains limited by the quality of the extracted video features.

More tightly coupled world-action models jointly model visual features and actions with shared or entangled architectures ([31], [32], [33], [3], [34]). Our work studies a complementary axis: what future state should the world model predict? Instead of pixels, video latents, or action tokens, we predict 3D scene flow, discarding texture, lighting, and material details that are often irrelevant for inverse dynamics.

3 Method

Existing WAMs predict future video jointly with actions, forcing the backbone to model details that low-level control doesn't require. FLOMO predicts 3D scene flow in place of video. We predict within the latent space of a pretrained video generator (Wan 2.2), so the model inherits large-scale video priors through its backbone. Our model is co-trained on egocentric human video and robot teleoperation, predicting both scene flow and actions from an initial observation and a language instruction.

3.1 Problem Formulation

We assume access to two types of datasets: an *egocentric video dataset* $\mathcal{V} = \{(\mathbf{o}_{0:T-1}, \ell)\}$ of RGB observation sequences $\mathbf{o}_{0:T-1} \in \mathbb{R}^{T\times H\times W\times 3}$ ($T = 16$) paired with language instructions ℓ, and a *robot interaction dataset* $\mathcal{R} = \{(\mathbf{o}_{0:T-1}, \ell, \mathbf{a})\}$ that additionally provides an action chunk $\mathbf{a} = (\mathbf{a}_0, \ldots, \mathbf{a}_{T-1}) \in \mathbb{R}^{T\times d_a}$ over a d_a-dimensional action space. For all clips in both datasets, we extract a 3D scene-flow sequence $\mathbf{F}_{0:T-1}$ in an offline preprocessing step (Sec. 3.2). Given an initial observation $\mathbf{o}_0$ and instruction ℓ, our model predicts future scene flow and actions; the full joint distribution our architecture represents is:

$$p(\mathbf{F}_{0:T-1},\ \mathbf{a} \mid \mathbf{o}_0,\ \ell)\,, \tag{1}$$

3.2 3D Scene Flow as the Video-Action Intermediate

The action producing a future state depends on how the scene moves, not on how it looks, motivating motion rather than pixels as the prediction target. 2D optical flow captures motion but conflates

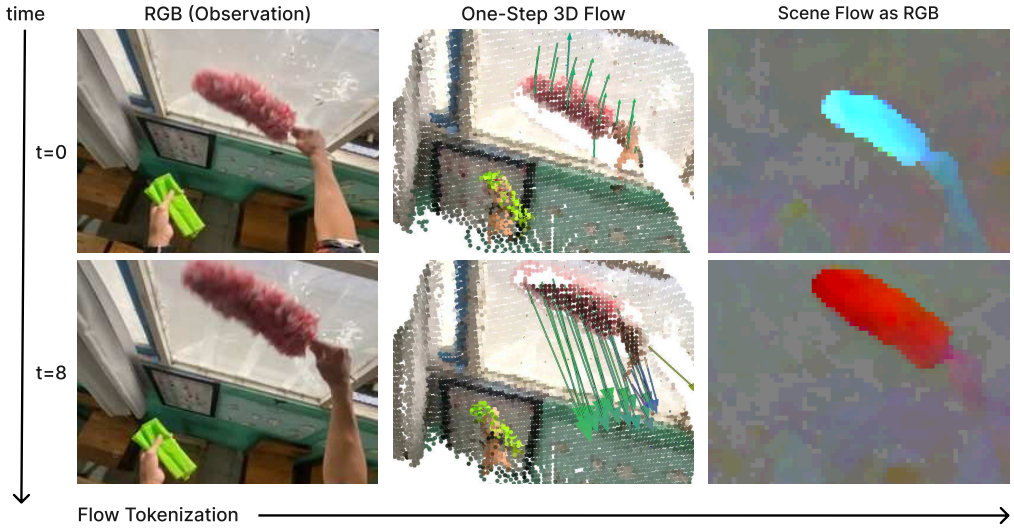

Figure 2: Scene-flow tokenization. Each row shows a timestep from our video dataset. The model recovers the dominant task-relevant motion field while remaining invariant to texture changes that confound pixel-space predictors.

camera ego-motion with task-relevant motion. We instead predict 3D scene flow in a canonical frame, which isolates the geometric transition between states while discarding appearance.

Scene-flow extraction and canonical frame. We extract 3D point trajectories in an offline preprocessing step using an off-the-shelf 3D point tracker [35]. For each clip in $\mathcal{V}$ and $\mathcal{R}$ we slide a window of $T = 16$ frames and place $Q = 4096$ query points on a 64×64 grid covering the first frame. The tracker back-projects these queries to 3D and propagates them across the window, yielding trajectories $\{\mathbf{p}_i(u,v)\}_{i=0}^{T-1}$. We express every trajectory in the camera frame of the window's first (conditioning) timestep, so the coordinate origin is the current camera's focal point. This makes the representation invariant to camera ego-motion and yields a motion field that primarily reflects object and hand movement.

Cumulative scene flow. Rather than supervising on instantaneous per-frame velocities, we predict *cumulative* scene flow, defined as the total 3D displacement of each point from the first frame to frame i:

$$\mathbf{F}_i(u,v) \;=\; \mathbf{p}_i(u,v) \;-\; \mathbf{p}_0(u,v), \tag{2}$$

so $\mathbf{F}_0 = \mathbf{0}$ by construction. We find cumulative scene flow to be a smoother target than instantaneous scene flow: it reduces the per-frame tracking noise that dominates the supervision signal, letting fine-grained motion like fingertip and object movement be predicted accurately.

3.3 Tokenized Scene Flow

We render the cumulative scene flow $\mathbf{F}_{0:T-1}$ as an RGB video $\mathbf{F}^{\mathrm{rgb}}$ by clipping each displacement channel to its global per-dataset 1st/99th percentile range and rescaling to $[0,1]$, then treat it exactly like an RGB video. We encode it with the frozen Wan 2.2-TI2V VAE encoder $\mathcal{E}$:

$$\mathbf{z}^f \;=\; \mathcal{E}\big(\mathbf{F}^{\mathrm{rgb}}\big) \;\in\; \mathbb{R}^{T_z \times c \times H_z \times W_z}, \tag{3}$$

where T_z is the number of latent frames (temporal stride $s_T = 4$, with the first frame encoded on its own to match the TI2V backbone), $c = 16$ is the VAE latent channel count, and $H_z \times W_z$ are the latent spatial dimensions (spatial stride 16). Representing scene flow as an RGB video lets us reuse the pretrained VAE without modification, so scene flow and video latents share the same distribution and the backbone does not significantly diverge from its video pretraining. Figure 6 illustrates this tokenization scheme.

3.4 Architecture

FLOMO is instantiated from the pretrained Wan 2.2 5B Text-Image-to-Video DiT [36], extended with lightweight modality-specific encoder and decoder heads. The current observation is encoded by

the frozen VAE, $\mathbf{z}^o = \mathcal{E}(\mathbf{o}_0)$, forming a clean conditioning prefix, while the frozen umT5-XXL text encoding $\boldsymbol{\tau} = \mathrm{umT5}(\ell)$ [37] conditions the backbone through cross-attention. The cumulative scene-flow latents $\mathbf{z}^f$ and action tokens are the prediction targets; future RGB latents $\mathbf{z}^v = \mathcal{E}(\mathbf{o}_{1:T-1})$ are an additional target used only in video-prediction ablations. Each action $\mathbf{a}_i \in \mathbb{R}^{d_a}$ is mapped to a DiT token by a learned linear projection $\mathbb{R}^{d_a} \to \mathbb{R}^d$. All target tokens are concatenated along the sequence dimension and jointly denoised by the shared DiT backbone. We fine-tune the backbone using LoRA [38] with rank $r = 64$ and scaling factor $\alpha = 128$, keeping the Wan VAE and text encoder frozen. We adopt an attention mask similar to Wan, allowing all tokens to attend to each other. Each prediction head can be independently activated or dropped per sample via loss masking, so the same backbone supports any subset of output modalities. By default, our model predicts scene flow and actions; our ablations reuse the same flexible architecture to additionally predict future video, or to predict video and actions without scene flow.

3.5 Training Objective

We adopt a flow-matching objective [39] over all prediction targets. For any clean target $\mathbf{x}_0 \in \{\mathbf{z}^f, \mathbf{a}, \mathbf{z}^v\}$, we form the noisy interpolant at a single noise level $t \sim \mathcal{U}[0,1]$ shared across all modalities, $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with velocity target $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{x}_0$. The training loss is

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{\mathbf{x}_0,\,\boldsymbol{\epsilon},\,t}\Big[\lambda_f \|\hat{\mathbf{v}}^f - \mathbf{v}^f\|_2^2 + \lambda_a \|\hat{\mathbf{v}}^a - \mathbf{v}^a\|_2^2 + \lambda_v \|\hat{\mathbf{v}}^v - \mathbf{v}^v\|_2^2\Big], \tag{4}$$

where $\lambda_a = 0$ for samples without action labels. By default, our model predicts motion and action only ($\lambda_f = 1$, $\lambda_a = 0.1$, $\lambda_v = 0$); the video term ($\lambda_v = 1$) is enabled only for ablations that predict future video. Actions use a 20-dimensional bimanual end-effector space (per arm: 3D position, 6D rotation, 1D gripper), normalized via 2nd/98th percentile clipping following [40].

3.6 Training Procedure

We train our model in two stages using a mixture of egocentric human video and robot interaction data.

Stage 1: Mid-training. We train on the full data mixture combining human video and robot interaction datasets. Human video samples are supervised on the scene-flow loss only ($\lambda_a = 0$, $\lambda_v = 0$), while robot samples are supervised on scene flow and action. Dataset-specific batch weights control the relative sampling frequency. This stage exposes the backbone to diverse egocentric motion at scale, building a strong motion prior before any action-specific finetuning.

Stage 2: Fine-tuning. In a second stage, we exclusively train on robot interaction data, dropping the human video stream entirely. This sharpens action prediction without diluting the gradient signal with action-free samples.

3.7 Inference

At test time, given an initial observation $\mathbf{o}_0$ and language instruction ℓ, our model jointly denoises its output heads for $N = 10$ steps using the UniPC ODE solver [41]. The robot executes the full predicted chunk of $T = 16$ actions before re-planning from the next observation.

4 Experiments

We evaluate FLOMO along three axes. **(i) Policy evaluation** (Sec. 4.2, Sec. 4.3): we measure in-distribution task success and out-of-distribution generalization to novel objects, motion primitives, and tasks, comparing prediction targets (motion, video, and joint video-motion) against a no-pretraining baseline and prior video-to-action methods. **(ii) Sample efficiency** (Sec. 4.4): a controlled inverse-dynamics probe isolates how efficiently each representation recovers actions, separating the choice of representation from forward-dynamics prediction quality. **(iii) Mechanism** (Sec. 4.5): we analyze how the action stream allocates attention across modalities, testing whether motion is the channel through which video pretraining reaches action prediction.

Figure 3: In-distribution success rates. Predicting motion (ours) is the best target on every task; predicting only video or video + motion lowers success.

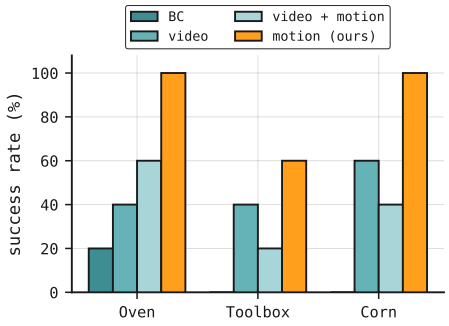

Table 1: Out-of-distribution results. FLOMO outperforms baselines across all tasks, highlighting its ability to generalize to unseen objects, motion primitives, and tasks. SR = successful rollouts / total; TP = avg. fraction of subtasks completed (%).

Method	Close Cabinet		Pull String		Highlighter	
	SR	TP	SR	TP	SR	TP
BC	3/5	80	0/5	0	1/5	50
Joint Video-Action	0/5	40	1/5	20	2/5	40
FLOMO w/ Video	0/5	0	0/5	0	0/5	0
AMPLIFY [5]	2/5	60	0/5	0	0/5	0
DreamZero [3]	2/5	80	1/5	30	1/5	30
FLOMO (ours)	**5/5**	**100**	**3/5**	**60**	**4/5**	**80**

4.1 Setup

Datasets. We train on a mixture of egocentric human video and bimanual YAM robot teleoperation, with per-stage sampling weights listed in Appendix Table 4. **EgoVerse** [42] contributes egocentric human video filtered for diversity from a 1,362-hour corpus (video and scene-flow supervision only). The remaining sources carry full video, scene-flow, and action supervision: **MolmoAct2-BimanualYAM** [43], filtered for diversity from ∼700 hours of bimanual teleoperation; 6h of in-house task-directed YAM demonstrations; 1h of in-house human demonstrations on overlapping tasks; and 30min of in-house undirected YAM play. Mid-training uses the full mixture; fine-tuning uses only the in-house YAM demonstration and play data.

Baselines. We compare FLOMO against three baselines: (1) **BC**, our backbone architecture (Wan 2.2, reduced to 8 layers) trained from scratch on robot data only, which ablates video pretraining; (2) **AMPLIFY** [5], which predicts tokenized 2D point tracks as an intermediate between video and action; and (3) **DreamZero** [3], a state-of-the-art world action model. We train all models on the same data mixture described above until convergence. To isolate the effect of the prediction target, we additionally evaluate ablations of our model that predict video and action without motion (**Joint Video-Action**), or that jointly predict video, motion, and action (**FLOMO w/ Video**).

Evaluation protocol. We report real-robot success rate (SR) and average task progress (TP) over multiple rollouts for policy evaluation. TP is computed per rollout as the fraction of ordered subtasks completed before failure, then averaged across rollouts; a rollout that fails at subtask k of S receives a score of $(k-1)/S$. A controlled inverse-dynamics probe (Sec. 4.4) separately measures the flow-matching loss for the action prediction head on a held-out dataset to isolate the choice of representation from forward-dynamics quality.

4.2 In-Distribution Real-Robot Evaluation

Setup. We evaluate on three contact-rich, precision tasks on the bimanual YAM platform. *Oven* requires opening a toaster-oven door, retrieving a croissant from inside, and placing it on a plate. *Toolbox* requires opening a hinged lid via a small crevice, grasping a screwdriver, and placing it in the container. *Corn* requires grasping a corn cob and placing it in a bowl. All rollouts begin from randomized object and robot configurations; we run 5 rollouts per task and report success rate (SR) and task progress (TP). We compare our model against a 20M behavior-cloning (BC) baseline as well as two variants that differ only in prediction target, specifically Joint Video-Action and FLOMO w/ Video.

Results. Figure 3 reports in-distribution results. FLOMO achieves the highest success rate on every task, averaging 87% versus 47% for Joint Video-Action, 40% for FLOMO w/ Video, and 7% for BC. These in-distribution results suggest that adding a video-prediction target on top of motion-prediction does not improve over predicting video and action alone, and predicting motion alone outperforms

both. The advantage is largest on *Oven*, where success depends on precisely aligning the gripper with the oven handle and executing multi-step motions required to swing the door open. 3D motion supervision targets the geometry of this contact directly, capturing the fine-grained dynamics of the interaction, whereas a video target must additionally reconstruct task-irrelevant appearance such as reflections and textures.

4.3 Out-of-Distribution Generalization

Setup. We probe generalization along three axes, each isolating a distinct distribution shift. *Close Cabinet* involves pushing a cabinet door shut, an unseen motion primitive in robot data during training. *Pull String* requires pulling a hanging string to a target mark, an unseen task. *Highlighter* introduces an unseen object (highlighter) which must be placed in a bowl. Every condition additionally includes unseen distractor objects and perturbed initial states. We run 5 rollouts per condition and report SR and TP against BC, AMPLIFY [5], DreamZero [3], and the our two prediction-target variants.

Results. Table 1 reports out-of-distribution results. Our model retains strong performance across all three axes (80% average success), while every method that predicts video as a target degrades severely: FLOMO w/ Video, which predicts video, motion, and action jointly, collapses to 0% across all three tasks despite reaching 43% in distribution, and DreamZero, a state-of-the-art dedicated video-prediction policy, averages only 27%. Joint Video-Action also performs poorly at 20%.

All methods that predict pixels, through a video objective (FLOMO w/ Video, Joint Video-Action, DreamZero), or through image-space 2D tracks (AMPLIFY), degrade sharply in performance. We hypothesize that our 3D scene flow representation retains performance due to its invariance to appearance and camera movement in pretraining and midtraining data. Removing the image-space prediction target recovers generalization, and adding it, even alongside motion, destroys generalization capabilities.

We attribute this gap to what each target encodes. A video objective binds the model to scene-specific appearance, texture, and object identity, none of which are preserved under the distribution shifts we test. The model overfits to the visual statistics of its training environments. By construction, 3D scene flow discards these factors and retains only the geometric transition between states, a representation that is invariant to appearance and shared across embodiments. This is what allows our model to transfer a motion primitive such as string-pulling, observed only in human video, to the robot: on *Pull String*, FLOMO succeeds on $3/5$ rollouts while no baseline exceeds $1/5$.

4.4 Inverse-Dynamics Scaling and Sample Efficiency

The policy results above establish motion as the more effective prediction target for WAMs, but the source of this gain remains ambiguous: it could stem from more accurate forward-dynamics predictions or a more decodable conditioning representation for inverse dynamics. We isolate the latter with a controlled inverse-dynamics probe. We repurpose FLOMO to predict future action chunks from *ground-truth* future signals, eliminating forward-dynamics error so that any remaining difference in action error is attributable to the conditioning representation alone. We compare conditioning on ground-truth future scene flow against ground-truth future RGB, training each on $\{1\%, 10\%, 100\%\}$ of a filtered subset of **MolmoAct2-BimanualYAM** and reporting the minimum held-out action error.

Motion is the more sample-efficient signal (Fig. 4): at 1% of the action data it yields roughly 22% lower action error than video (0.058 vs. 0.074); the gap collapses to roughly $1\% - 2\%$ at scales above 10% of the dataset. Because the future is supplied identically in both conditions, this gap is a property of the representation: motion exposes the action-relevant structure directly (what moved, and where), whereas recovering it from raw appearance takes more examples. Motion is thus both an easier target to predict and is the more efficient representation of the information needed to predict actions.

4.5 Where and when does action prediction use motion?

To compare the importance of each modality for action prediction, we investigate the attention maps of FLOMO w/Video. We hypothesize that if motion is a more effective channel through which video

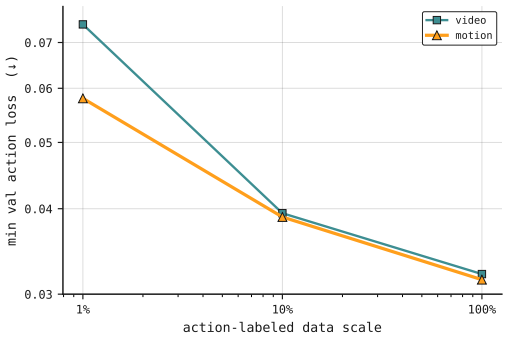

Figure 4: Inverse-dynamics scaling. Held-out action error versus action-data scale for two *ground-truth* conditioning signals (log-log). Recovering actions from motion is more sample-efficient than from video: the gap is largest at 1% of the data and narrows as data grows. Because the future is supplied in both cases, the gap reflects how directly each representation exposes action-relevant structure, not how hard it is to predict.

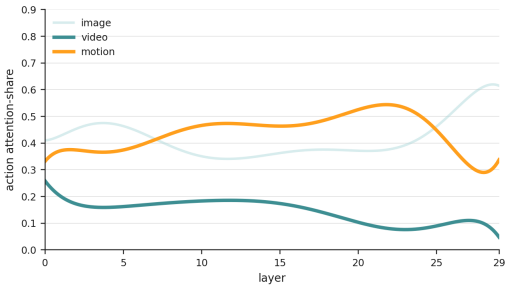

Figure 5: Action attention concentrates on motion tokens. Share of action-query attention to each input modality, averaged over attention heads and robot validation rollouts, as a function of DiT layer. Attention to motion rises through the trunk and dominates at the late-middle layers, while attention to video stays lowest throughout; averaged over layers, action queries attend to motion roughly $3\times$ more than to video.

pretraining transfers to action, then action tokens should attend predominantly to motion tokens rather than video tokens, and this pattern should persist across embodiments.

Figure 5 reports the share of action-query attention going to each input modality (image, video, motion) as a function of DiT layer, aggregated across attention heads, averaged over $1,000$ held-out samples. Beginning from an approximately uniform distribution at the input layer, action queries progressively reallocate attention toward motion tokens through the network trunk, reaching $\sim 55\%$ at the late-middle layers. Attention to video exhibits the opposite trend, remaining the lowest across all layers and declining to $\sim 10\%$ at the deepest layers. Averaged over attention heads and layers, action queries attend to motion approximately $3\times$ more than to video, and this ordering is consistent across both robot teleoperation datasets we evaluate. Taken together, these results indicate that the action stream draws predominantly on motion rather than video, supporting our hypothesis that 3D scene flow is the more direct channel from video pretraining to action.

5 Discussion and Limitations

We argue that the priors of video pretraining are carried by a model's weights, not by what it is trained to predict. A world-action model should target motion rather than pixels: a low-dimensional, texture-invariant signal that an action head can decode directly. Rendering 3D scene flow into the latent space of a pretrained video generator lets a single backbone predict motion and action without a new tokenizer, inheriting internet-scale video priors through its weights. Across real-robot manipulation, predicting motion outperforms predicting video and jointly predicting video and motion, both in distribution and on unseen objects, motion primitives, and tasks; action queries attend far more to motion than to video; and a controlled inverse-dynamics probe shows motion is the more sample-efficient conditioning signal. Together these results point to motion as the more effective medium through which video pretraining reaches action.

Two limitations temper these results. First, the backbone is a 5B video-generation model, so both fine-tuning and inference are compute intensive, and extracting 3D scene flow adds an offline preprocessing step for every clip; lighter motion representations or a distilled backbone would make the approach more practical. Second, FLOMO predicts motion and action jointly, and although the two usually agree, they sometimes diverge. A plausible predicted motion does not guarantee a correct action, because the action head is not constrained to be exactly consistent with the predicted flow and distinct actions can produce the same scene motion. In these cases, treating predicted motion as a proxy for the executed action breaks down, and tightening the coupling between the two predictions through mechanisms such as per-modality variance schedules or objectives to maximize mutual information between modalities is a natural direction for future work.

Acknowledgments

If a paper is accepted, the final camera-ready version will include acknowledgments here.

References

[1] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin, L. Magne, A. Mandlekar, A. Narayan, Y. L. Tan, G. Wang, J. Wang, Q. Wang, Y. Xu, X. Zeng, K. Zheng, R. Zheng, M.-Y. Liu, L. Zettlemoyer, D. Fox, J. Kautz, S. Reed, Y. Zhu, and L. Fan. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. doi:10.48550/arXiv.2505.12705. URL https://arxiv.org/abs/2505.12705.

[2] J. Liang, P. Tokmakov, R. Liu, S. Sudhakar, P. Shah, R. Ambrus, and C. Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. doi:10.48550/arXiv.2508.00795. URL https://arxiv.org/abs/2508.00795.

[3] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, A. Malik, K. Lee, W. Liang, N. Ranawaka, J. Gu, Y. Xu, G. Wang, F. Hu, A. Narayan, J. Bjorck, J. Wang, G. Kim, D. Niu, R. Zheng, Y. Xie, J. Wu, Q. Wang, R. Julian, D. Xu, Y. Du, Y. Chebotar, S. Reed, J. Kautz, Y. Zhu, L. J. Fan, and J. Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. doi:10.48550/arXiv.2602.15922. URL https://arxiv.org/abs/2602.15922.

[4] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems (RSS)*, 2024. URL https://arxiv.org/abs/2401.00025.

[5] J. A. Collins, L. Cheng, K. Aneja, A. Wilcox, B. Joffe, and A. Garg. AMPLIFY: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025. doi:10.48550/arXiv.2506.14198. URL https://arxiv.org/abs/2506.14198.

[6] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2407.15208.

[7] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022. URL https://arxiv.org/abs/2203.12601.

[8] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.00030. Notable-Top-25% (Spotlight).

[9] C. Bhateja, D. Guo, D. Ghosh, A. Singh, M. Tomar, Q. Vuong, Y. Chebotar, S. Levine, and A. Kumar. Robotic offline RL from internet videos via value-function pre-training. *arXiv preprint arXiv:2309.13041*, 2023. doi:10.48550/arXiv.2309.13041. URL https://arxiv.org/abs/2309.13041.

[10] N. Dashora, D. Ghosh, and S. Levine. ViVa: Video-trained value functions for guiding online RL from diverse data. *arXiv preprint arXiv:2503.18210*, 2025. doi:10.48550/arXiv.2503.18210. URL https://arxiv.org/abs/2503.18210.

[11] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. Behbahani, S. Chan, N. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2402.15391.

[12] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo. Latent action pretraining from videos. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.11758.

[13] Y. Chen, Y. Ge, W. Tang, Y. Li, Y. Ge, M. Ding, Y. Shan, and X. Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2412.04445.

[14] R. Punamiya, D. Patel, P. Aphiwetsa, P. Kuppili, L. Y. Zhu, S. Kareer, J. Hoffman, and D. Xu. EgoBridge: Domain adaptation for generalizable imitation from egocentric human data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2509.19626.

[15] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. EgoMimic: Scaling imitation learning via egocentric video. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://arxiv.org/abs/2410.24221.

[16] H. Kim, J. Kang, H. Kang, M. Cho, S. J. Kim, and Y. Lee. UniSkill: Imitating human videos via cross-embodiment skill representations. In *Conference on Robot Learning (CoRL)*, 2025. URL https://arxiv.org/abs/2505.08787.

[17] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu. OKAMI: Teaching humanoid robots manipulation skills through single video imitation. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2410.11792. Oral.

[18] P. Dan, K. Kedia, A. Chao, E. W. Duan, M. A. Pace, W.-C. Ma, and S. Choudhury. X-Sim: Cross-embodiment learning via real-to-sim-to-real. In *Conference on Robot Learning (CoRL)*, 2025. URL https://arxiv.org/abs/2505.07096. Oral.

[19] C. Yuan, C. Wen, T. Zhang, and Y. Gao. General flow as foundation affordance for scalable robot learning. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2401.11439.

[20] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh. FlowRetrieval: Flow-guided data retrieval for few-shot imitation learning. In *Conference on Robot Learning (CoRL)*, 2024. URL https://arxiv.org/abs/2408.16944.

[21] C. Gao, H. Zhang, Z. Xu, Z. Cai, and L. Shao. FLIP: Flow-centric generative planning as general-purpose manipulation world model. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2412.08261.

[22] S. Liu, X. Ren, T. Shen, H. Ling, S. Gupta, S. Wang, S. Fidler, and J. Gao. MoRight: Motion control done right. *arXiv preprint arXiv:2604.07348*, 2026. doi:10.48550/arXiv.2604.07348. URL https://arxiv.org/abs/2604.07348.

[23] S. Lee, Y. Jung, I. Chun, Y.-C. Lee, Z. Cai, H. Huang, A. Talreja, T. D. Dao, Y. Liang, J.-B. Huang, and F. Huang. TraceGen: World modeling in 3D trace space enables learning from cross-embodiment videos. *arXiv preprint arXiv:2511.21690*, 2025. doi:10.48550/arXiv.2511.21690. URL https://arxiv.org/abs/2511.21690.

[24] J. Guo, X. Ma, Y. Wang, M. Yang, H. Liu, and Q. Li. FlowDreamer: A RGB-D world model with flow-based motion representations for robot manipulation. *IEEE Robotics and Automation Letters (RA-L)*, 2026. URL https://arxiv.org/abs/2505.10075.

[25] A. Hung, B. P. Duisterhof, and J. Ichnowski. 3PoinTr: 3D point tracks for robot manipulation pretraining from casual videos. *arXiv preprint arXiv:2603.08485*, 2026. doi:10.48550/arXiv.2603.08485. URL https://arxiv.org/abs/2603.08485.

[26] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2302.00111.

[27] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations (ICLR)*, 2024.

[28] Y. Wen, J. Lin, Y. Zhu, J. Han, H. Xu, S. Zhao, and X. Liang. VidMan: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2411.09153.

[29] Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2412.14803. Spotlight.

[30] J. Pai, L. Achenbach, V. Montesinos, B. Forrai, O. Mees, and E. Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025. doi:10.48550/arXiv.2512.15692. URL https://arxiv.org/abs/2512.15692.

[31] C. Zhu, R. Yu, S. Feng, B. Burchfiel, P. Shah, and A. Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Robotics: Science and Systems (RSS)*, 2025. URL https://arxiv.org/abs/2504.02792.

[32] S. Li, Y. Gao, D. Sadigh, and S. Song. Unified video action model. In *Robotics: Science and Systems (RSS)*, 2025. URL https://arxiv.org/abs/2503.00200.

[33] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn, and J. Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026. doi:10.48550/arXiv.2601.16163. URL https://arxiv.org/abs/2601.16163.

[34] T. Yuan, Z. Dong, Y. Liu, and H. Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. doi:10.48550/arXiv.2603.16666. URL https://arxiv.org/abs/2603.16666.

[35] Y. Xiao, J. Wang, N. Xue, N. Karaev, Y. Makarov, B. Kang, X. Zhu, H. Bao, Y. Shen, and X. Zhou. SpatialTrackerV2: 3D point tracking made easy. *arXiv preprint arXiv:2507.12462*, 2025. doi:10.48550/arXiv.2507.12462. URL https://arxiv.org/abs/2507.12462.

[36] Team Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. doi:10.48550/arXiv.2503.20314. URL https://arxiv.org/abs/2503.20314.

[37] H. W. Chung, N. Constant, X. Garcia, A. Roberts, Y. Tay, S. Narang, and O. Firat. UniMax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*, 2023. doi:10.48550/arXiv.2304.09151. URL https://arxiv.org/abs/2304.09151.

[38] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[39] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. doi:10.48550/arXiv.2210.02747. URL https://arxiv.org/abs/2210.02747.

[40] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *Science Robotics*, 11(113):eaea6201, 2026.

[41] W. Zhao, L. Bai, Y. Rao, J. Zhou, and J. Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36: 49842–49869, 2023.

[42] R. Punamiya, S. Kareer, Z. Liu, J. Citron, R.-Z. Qiu, X. Cai, A. Gavryushin, J. Chen, D. Liconti, L. Y. Zhu, P. Aphiwetsa, B. Li, A. Cheluva, P. Kuppili, Y. Liu, D. Patel, A. Gao, H.-Y. Chung, R. Co, R. Zbizika, J. Liu, X. Xu, H. Xiong, G. Chen, S. Oliani, C. Yang, X. Wang, J. Fort, R. Newcombe, J. Gao, J. Chong, G. Matsuda, A. Doriwala, M. Pollefeys, R. Katzschmann, X. Wang, S. Song, J. Hoffman, and D. Xu. EgoVerse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026. doi:10.48550/arXiv.2604.07607. URL https://arxiv.org/abs/2604.07607.

[43] H. Fang, J. Duan, D. Clay, S. Wang, S. Liu, W. Huang, X. Fan, W.-C. Tsai, S. Chen, Y. R. Wang, S. Xing, J. Cho, J. S. Park, A. Eftekhar, P. Sushko, K. Farley, A. Wadhwa, C. Harrison, W. Han, Y.-C. Lee, E. VanderBilt, R. Hendrix, S. Ellawela, L. Ngoo, J. Chai, Z. Ren, A. Farhadi, D. Fox, and R. Krishna. MolmoAct2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026. doi:10.48550/arXiv.2605.02881. URL https://arxiv.org/abs/2605.02881.

[44] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022. doi:10.48550/arXiv.2203.06173. URL https://arxiv.org/abs/2203.06173.

[45] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 416–426. PMLR, 2023. URL https://proceedings.mlr.press/v205/radosavovic23a.html.

[46] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, P. Abbeel, J. Malik, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 655–677, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/022ca1bed6b574b962c48a2856eb207b-Abstract-Conference.html.

[47] Z. J. Cui, H. Pan, A. Iyer, S. Haldar, and L. Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL https://arxiv.org/abs/2409.12192.

[48] A. Liang, P. Czempin, M. Hong, Y. Zhou, E. Bıyık, and S. Tu. CLAM: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025. doi:10.48550/arXiv.2505.04999. URL https://arxiv.org/abs/2505.04999.

[49] J. Yang, Y. Shi, H. Zhu, M. Liu, K. Ma, Y. Wang, G. Wu, T. He, and L. Wang. CoMo: Learning continuous latent motion from internet videos for scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 42352–42363, 2026. URL https://arxiv.org/abs/2505.17006.

[50] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, pages 306–324, 2024. doi:10.1007/978-3-031-73116-7_18. URL https://arxiv.org/abs/2405.01527.

[51] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman. TAPIR: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10061–10072, 2023. URL https://arxiv.org/abs/2306.08637.

[52] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. CoTracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, pages 18–35, 2024. doi:10.1007/978-3-031-73033-7_2. URL https://arxiv.org/abs/2307.07635.

Appendix

A Implementation Details

A.1 Architecture

FLOMO is initialized from the Wan 2.2 5B Text-Image-to-Video DiT [36]: a 30-layer transformer with hidden width d=3072, 24 attention heads (head dimension 128), feed-forward width 14,336, and a $(1, 2, 2)$ patchifier over the VAE latent grid. We use bidirectional (full) attention so every token attends to every other. On top of the shared backbone we add lightweight modality-specific heads:

- **Action head.** Each per-timestep action vector $\mathbf{a}_i \in \mathbb{R}^{d_a}$ (d_a=20) is mapped to a DiT token by a single linear projection $\mathbb{R}^{d_a} \to \mathbb{R}^d$ plus a learned per-position embedding; predicted action tokens are read back out by a single linear projection $\mathbb{R}^d \to \mathbb{R}^{d_a}$.
- **Video and scene-flow heads.** Video frames and RGB-rendered scene-flow frames are both encoded and decoded by the frozen Wan 2.2 VAE. On the trunk side they reuse the backbone's own patchify (Conv3d) and unpatchify (linear) projections rather than adding new layers, so these heads contribute no trainable parameters.
- **Conditioning.** The initial observation $\mathbf{o}_0$ is encoded by the frozen Wan VAE into $\mathbf{z}^o$, which forms a clean prefix in the token sequence that is never noised; the instruction $\boldsymbol{\tau}$ is encoded by the frozen umT5-XXL text encoder [37] (512 tokens, width 4096) and conditions the backbone through cross-attention. For the robot datasets the observation comprises three camera views (one front and two wrist), each VAE-encoded into its own prefix tokens; the human-video data uses a single view. Scene flow and video are predicted only in the front view.

We fine-tune the DiT with LoRA [38] (r=64, α=128, dropout 0), injecting adapters into the query/key/value/output projections of both self- and cross-attention and into both feed-forward linear layers of all 30 blocks; the Wan VAE and umT5 stay frozen. This leaves ≈161M trainable parameters (LoRA adapters plus the small action head) on top of the frozen 5B backbone.

A.2 Training Hyperparameters

Both stages use AdamW (β_1=0.95, β_2=0.999, weight decay 10^{-6}), gradient clipping at 1.0, bf16 mixed precision, and a constant learning rate. All inputs are processed at 256×256 resolution, giving a 16×16 Wan-VAE latent grid, and each example spans a 16-frame window: the model conditions on the initial frame and predicts the 16-step action chunk together with the video and scene flow over the window. Training uses a flow-matching objective with 1,000 timesteps. We augment every source except EgoVerse with random-resized-crop (scale $[0.85, 1.0]$; ratio $[0.98, 1.02]$) and color jitter (brightness, contrast, and saturation 0.25; hue 0.1). All models are fine-tuned from their respective epoch-28 mid-training checkpoints, using only in-house YAM data. Unless otherwise noted, fine-tuning evaluations use the epoch-3 checkpoint. Per-stage training settings are listed in Table 2.

Table 2: Per-stage training settings.

	Mid-training	Fine-tuning
Learning rate (constant)	2.8×10^{-4}	5.6×10^{-5}
Global batch size	512	256
Action loss weight λ_a	0.1	1.0
Optimizer steps	≈14,000	≈1,500

A.3 Inference

At test time, we draw the scene-flow and action tokens with a flow-matching sampler [39], integrating the learned velocity field with the UniPC ODE solver [41] under a shifted noise schedule (shift 5) for N=10 steps. We do not apply classifier-free guidance; the model is run on the conditional path only.

We execute all 16 predicted actions per planning cycle, re-planning from the next observation. Each planning cycle (one joint motion-and-action denoising of $N{=}10$ steps through the full 5B trunk) takes $\approx$0.9 s on a single NVIDIA B200 GPU; asynchronous execution overlaps planning with robot motion.

A.4 Loss Weighting and Modality Masking

The training loss (Eq. 4) uses $\lambda_f{=}1$, $\lambda_a{=}0.1$, and $\lambda_v{=}0$ by default; the video term ($\lambda_v{=}1$) is enabled only for video-prediction ablations. During fine-tuning (Stage 2) we raise the action weight to $\lambda_a{=}1$. Per-sample modality masking sets $\lambda_a{=}0$ for action-free samples (human video), allowing co-training across datasets with mismatched modality coverage.

B Dataset Details

B.1 Source Datasets and Filtering

Below we describe each source dataset and how it is filtered to the subset used in training; Table 3 summarizes the result.

EgoVerse. EgoVerse [42] is a large egocentric human-manipulation video dataset; from it we use an 828 h subset collected by Mecka AI. From this we build a task-balanced subset by round-robin over tasks (one clip per task, then a second clip per task, and so on), giving a 200 h manifest of 8,578 clips over 3,696 tasks. We hold out 9 environments (35 tasks) for validation. From the training split we randomly subsample 16-frame windows down to a fixed budget, shuffling first so the kept windows cover all tasks rather than clustering on a few. Mid-training uses this subset, which spans $\approx$81 h of unique footage; no motion filtering is applied.

MolmoAct2-BimanualYAM. MolmoAct2-BimanualYAM [43] is a bimanual-YAM robot teleoperation dataset (737 recordings, three $640{\times}360$ views at 30 fps). We drop 7 "spell-out" tasks and round-robin across the remaining 25 coarse tasks to a task-balanced selection, of which $\sim$10 h (428 clips) is used for training.

In-house YAM. Teleoperated bimanual-YAM demonstrations of three tasks: put the corn in the bowl, open the oven and place the croissant on the plate, and open the toolbox and place the screwdriver inside. Each task includes randomized initial states as well as active data collection to enable data-efficient recovery from failure modes. This subset consists of 1,029 demonstrations totaling $\approx$3.7 h. Before training, data containing motion above or below pre-defined thresholds is filtered out.

In-house Human Data. Action-free human videos of the In-house YAM tasks, recorded from a single front-facing camera without action labels, so they supervise only the video and scene-flow heads. They cover the three robot tasks plus related variations (asparagus or pepper placed in a bowl or oven, pliers into the toolbox), for 710 demonstrations ($\approx$0.7 h) across 12 splits.

In-house Play YAM. A small set of unstructured play demonstrations on the YAM robot (173 demonstrations, $\approx$0.4 h), filtered like the In-house YAM data and pooled as an auxiliary set during fine-tuning (Table 4).

Table 3: Dataset composition. "Tasks" counts distinct task labels (EgoVerse uses fine-grained activity categories, the others coarse manipulation tasks); "Clips" counts clips, videos, or demonstrations (one recording each); "Hours" is the span of unique footage used in training: the number of distinct frames appearing in any 16-frame window, divided by the source frame rate. All figures are for the subset actually used in each split.

	Train			Validation		
Dataset	**Tasks**	**Hours**	**Clips**	**Tasks**	**Hours**	**Clips**
EgoVerse	3,660	81.0	4,303	35	1.0	46
MolmoAct2-BimanualYAM	25	10.0	428	22	1.0	46
In-house YAM	4	3.7	1,029	4	0.4	113
In-house Human Data	11	0.7	710	11	0.1	83
In-house Play YAM	162	0.4	173	19	0.05	19

B.2 Scene-Flow Extraction

We run SpatialTrackerV2 [35] on T=16-frame windows with Q=4096 query points uniformly arranged at t=0. We transform tracks into the current camera frame using estimated camera poses, then compute cumulative displacement (Eq. 2). Extraction runs at roughly 2.4 seconds per 16-frame window on a single NVIDIA L40S GPU, or about 72 GPU-hours per hour of 30 FPS video.

B.3 Sampling Weights

Table 4: Dataset sampling weights across training stages.

Dataset	**Mid-training weight**	**Fine-tuning weight**
EgoVerse	0.65	0.00
MolmoAct2-BimanualYAM	0.125	0.00
In-house YAM	0.125	0.75
In-house Human Data	0.05	0.00
In-house Play YAM	0.05	0.25

C Effect of Human Video on Out-of-Distribution Generalization

Table 1 compares prediction targets (motion vs. video) under a fixed training data mixture. To determine the impact of human video in our training data mixture, we train a separate version of FLOMO with the ≈81 h EgoVerse human-video subset removed. Table 5 reports out-of-distribution success for both FLOMO and FLOMO without human video. Removing the human-video subset lowers average success from 80% to 7%, which suggests that much of the model's out-of-distribution performance comes from the human video rather than from the prediction target alone.

Table 5: Effect of human video on out-of-distribution generalization. Both models predict 3D motion and differ only in whether the ≈81 h EgoVerse human-video subset is included in training. SR = successful rollouts / total; TP = avg. fraction of subtasks completed (%).

Training data	Close Cabinet		Pull String		Highlighter		Average	
	SR	TP	SR	TP	SR	TP	SR	TP
FLOMO	**5/5**	**100**	**3/5**	**60**	**4/5**	**80**	**80%**	**80**
FLOMO (no human video)	0/5	20	1/5	50	0/5	20	7%	30

The per-subtask scores show where the model trained without human video fails. It can begin each task but rarely completes the novel motion: it contacts the cabinet on 2 of 5 trials but never closes it (0/5), grasps the string on 4 of 5 trials but pulls it to the mark on only one, and grasps the highlighter on 2 of 5 trials but never places it in the bowl.

We verified this by direct text search of the training data. The robot data, both the MolmoAct2 episodes and the RealYAM demonstrations, contains no references to cabinets, strings, or markers, and the only door-like manipulation in either is the oven door in the croissant task, a different object class than a cabinet. The EgoVerse subset, by contrast, contains all three motion classes explicitly: cabinet closing (e.g. "close cabinet door under sink"), string pulling (e.g. "pull string from spool"), and slender-object placement (e.g. "place pen lid on pen, place pen in bowl"), even though the exact word "highlighter" never appears. Since these motions are reachable only through the human video, the transfer in Table 5 must originate there.

D Extended Related Work

Visual pretraining for robot control. Beyond video-based reward and value pretraining, a related line of work studies whether large-scale visual representations can serve as general robot perception backbones. Masked visual pretraining and large-scale egocentric representation learning have been shown to improve downstream control across simulated and real robot settings ([44], [45], [46]). These works establish the value of scalable visual pretraining for control, but mainly study the quality of the observation encoder. Our work instead studies the prediction target of a pretrained world-action model: whether future dynamics should be represented as pixels, video latents, or 3D motion.

Latent dynamics and latent action pretraining. Several recent methods learn transition variables from action-free observation sequences, using latent inverse dynamics, forward dynamics, contrastive objectives, or small amounts of action-labeled data to ground learned latent actions ([47], [48], [49]). These methods show that action-free video can induce useful transition representations, but the learned variables are implicit and may depend on the choice of reconstruction, prediction, or action-grounding objective. Our work instead supervises the intermediate with an explicit geometric target, 3D scene flow, tying the representation directly to physical displacement.

Point tracks and 3D motion extraction. Point tracks provide a compact way to expose object and hand motion without predicting full RGB futures. Prior work uses predicted point tracks as action-free plans for robot control ([50]), while general-purpose trackers make dense 2D and 3D motion extraction increasingly practical from ordinary videos ([51], [52], [35]). These works provide the tracking machinery and planning perspective that make motion-based supervision feasible. Rather than using tracks directly as plans or policy inputs, we convert tracked 3D trajectories into cumulative scene flow and use this motion field as the future prediction target inside a pretrained video-generation backbone.

E Failure Modes and Limitations

We qualitatively describe the failure modes and limitations we observed when running FLOMO during our real-robot rollouts. Most failures appear in the predicted actions rather than in the predicted motion, which is usually correct.

Actions that do not follow predicted motion. The predicted motion is usually correct in direction and shape, but the action head produces a chunk that does not realize it. The action head conditions on all three camera views, so this is not a perception limit but a gap in decoding motion into actions. We suspect incomplete cross-embodiment transfer contributes: the model has seen a motion performed by a human hand in the egocentric video but has not learned to map it onto the bimanual gripper, leaving the inverse-dynamics step rather than the motion target as the bottleneck.

Errors in motion prediction. Less often, the predicted motion itself is wrong, with too little motion or motion in the wrong direction. We suspect this reflects undertraining rather than a limitation of the representation, since these cases are rare and the motion target is otherwise reliable.

Self-occlusion under front-view prediction. The model conditions on the front and both wrist cameras but predicts scene flow only in the front view. When the arms occlude the manipulated object in the front view, the predicted motion degrades. The action head still attends to the wrist views, but the motion signal it is meant to follow is lost, and if the wrist cameras also do not show the

object the rollout fails. We have not tried predicting motion in the wrist views, which could supply an alternative viewpoint.

Instruction grounding. The model follows some parts of the instruction, such as which arm to use, but grounding the target object is less reliable. When the named object is unfamiliar, or when a different object sits in a location strongly associated with picking or placing in the training data, the model may act on that object instead of the requested one.

Scene-flow extraction failures. On occlusion-heavy clips the off-the-shelf 3D point tracker [35] produces noisy or incomplete trajectories, which enter training as degraded motion targets. This is a preprocessing limitation rather than a property of the model.

F Rollout Video Visualization

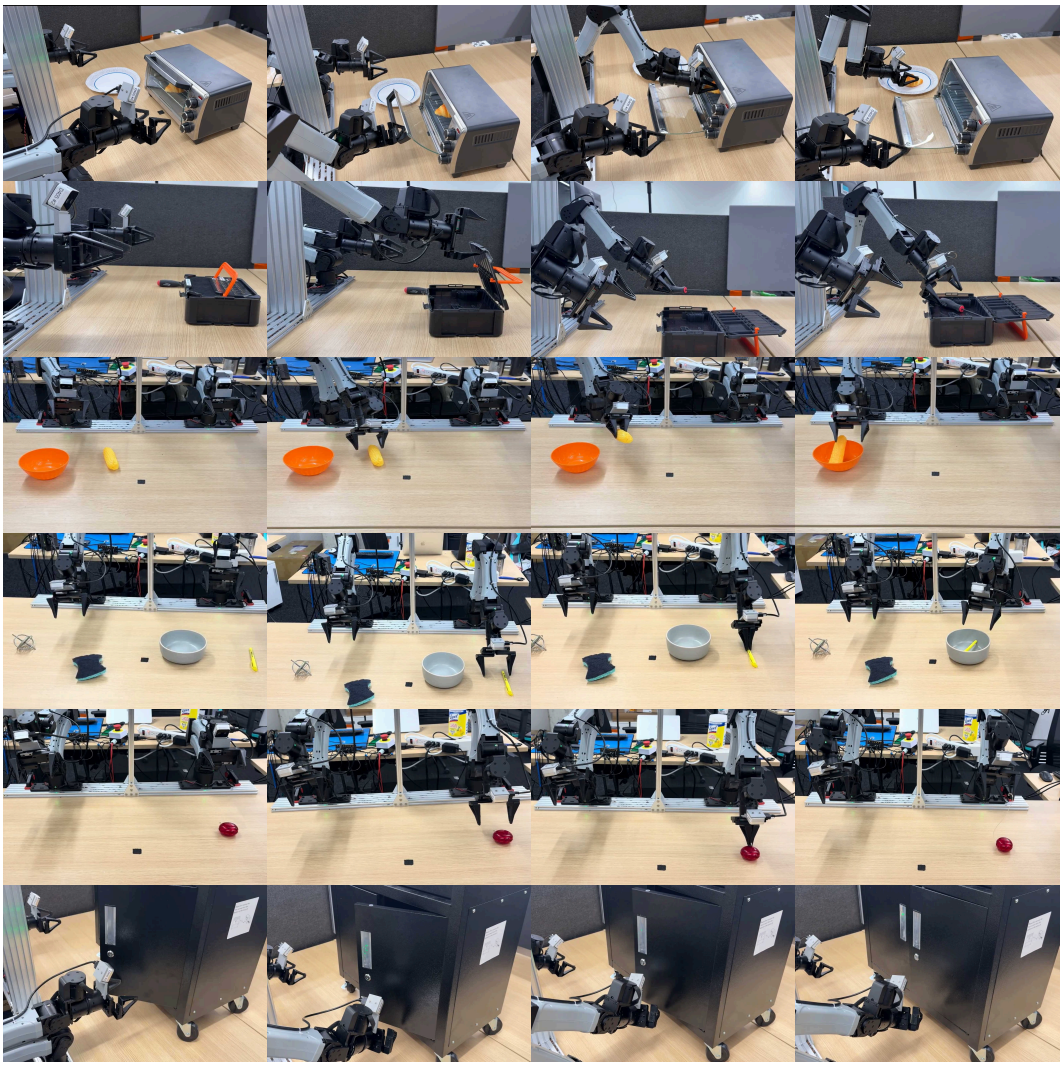

Figure 6: Rollout videos. Each row shows intermediate progression of successful FloMo rollouts of different tasks (3 in-distribution and 3 OOD). (**a**) The top row is opening the oven and placing the croissant on a plate. (**b**) The second row is opening the toolbox and placing the screwdriver inside. (**c**) The third row is placing the corn in the bowl. (**d**) The fourth row is putting the highlighter in the bowl (OOD task). (**e**) The fifth row is pulling the string on a toy (OOD task). (**f**) The last row is closing the cabinet (OOD task).