SHARED REPRESENTATIONS, NATIVE ACTIONS: LEARNING GENERALIZABLE SURGICAL MOTION FROM HETEROGENEOUS KINEMATICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning surgical tool-motion representations supports action-centric video understanding and autonomous surgery. Existing surgical video foundation models emphasize appearance and semantics, while video-only latent-action models may capture distractors without physical supervision. We present SAR-IDE (Surgical Action Representation through Inverse Dynamics Embedding), a surgical video foundation model pretrained through kinematics-supervised inverse dynamics on heterogeneous robot data. Embodiment-specific heads provide kinematic supervision for a shared visual encoder in each robot's native action space. An auxiliary tool-centric forward dynamics model connects latent actions to visible instrument motion and enables learning from videos without kinematic annotations. Across OpenH, SAR-RARP50, GraSP, and SurgPose, our frozen encoder outperforms the evaluated visual representation baselines on action regression, action recognition, and point tracking. Few-shot adaptation further transfers action prediction to an unseen robot embodiment. In real-robot experiments, augmenting policy training with model-inferred action labels improves needle-pickup success from 60% to 80% and inter-instrument transfer success from 0% to 30% over the limited-demonstration baseline. These results demonstrate the value of coupled forward–inverse dynamics pretraining for transferable surgical representations and the effectiveness of inferred action labels for downstream control.

1 INTRODUCTION

Recent work learning transferable representations from surgical videos [3; 11; 32; 44] have demonstrated promising results in action-centric application [46]. As surgical autonomy advances [9; 16; 26], OpenH [29] dataset providing large-scale robotic surgical video paired with synchronized kinematics emerged, enabling action representations to be learned from visual observations and measured robot kinematics data. Compared with general-purpose multi-robot datasets, surgical robot datasets [29] exhibit heterogeneous camera–arm geometry, with the relative placement of endoscopes and instrument varying across platforms and surgical setups. Meanwhile, machine-specific cable stretch, hysteresis, and wear, together with instrument-dependent tool-tip mappings, introduce further kinematic discrepancies [16; 29]. These differences complicate shared action representation learning from OpenH [29] dataset containing multi-embodiment surgical robots. Therefore, a motion foundation model that learns transferable action representations across heterogeneous robotic embodiments is of great importance.

Inverse dynamics modeling (IDM) learns action-aware representations from videos by predicting the action between two observations. The first line of work is latent-action models (LAMs) such as LAPA [50], combining IDM with a forward dynamics model (FDM) which predicts the future observation from current one given the predicted latent actions. This IDM-FDM pipeline can effectively learn from videos without action labels. However, LARY [23] shows that LAMs underperform general visual encoders on action regression and recognition. Moreover, FDM that predicts the whole-scene future observations may focus on action-irrelevant visual transitions such as appearance change and background tissue deformations, which impedes learning the critical action representation [24]. Different from learning latent actions, another line of work [10; 13] uses ground-truth kinematic actions to train IDM. Due to the kinematic heterogeneity, similar visible movements can correspond to different action labels, making direct learning from action targets problematic.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

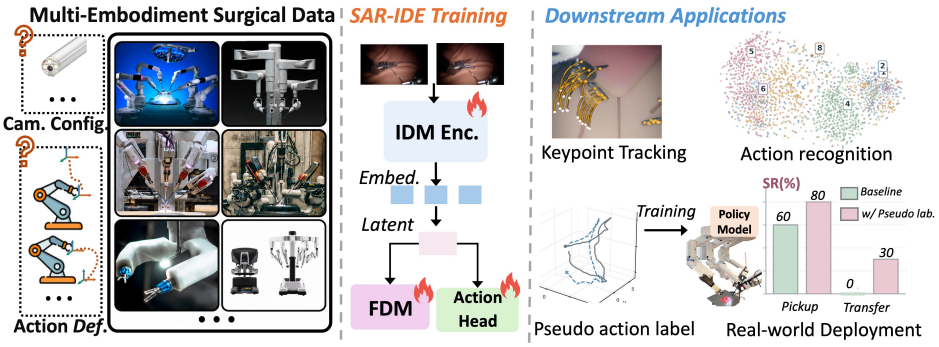

Figure 1: The SAR-IDE encoder is trained through inverse dynamics modeling on heterogeneous surgical robot data to learn enhanced motion representations for action-related downstream tasks.

To address these challenges, we propose SAR-IDE (Surgical Action Representation through Inverse Dynamics Embedding), a surgical video foundation model that learns transferable action embeddings through inverse-dynamics pretraining (Figure 1). Our key insight is that synchronized video and kinematics can teach the encoder to connect visual changes with physical instrument motion, providing motion-aware embeddings for downstream robot tasks. To help IDM pretraining learn transferable representations across different kinematic conventions, we introduce an auxiliary tool-centric FDM. Although robot actions are expressed in different coordinate systems, their effects can be supervised in observation space. By predicting future instrument masks, point tracks, and optical flow from latent actions and initial instrument prompts, the FDM provides a shared form of motion supervision that encourages implicit alignment across embodiments. Focusing these predictions on instruments further reduces interference from background changes. Instance-specific prompts also allow supervision to remain valid when annotations of other instruments are missing. This forward prediction objective enables videos without kinematic labels to contribute to pretraining, combining synchronized motion supervision and video-only learning within one framework for transferable surgical motion representations.

We pretrain SAR-IDE on approximately 462 hours of OpenH [29] video with synchronized kinematics, together with 22.4 hours of intraoperative video from SAR-RARP50 [31], GraSP [2], and Cholec80/CholecInstanceSeg [1; 40] without kinematics GT. Across extensive downstream probe experiments, our frozen encoder improves action regression, action recognition, and instrument tracking over the evaluated visual baselines. Few-shot regression further demonstrates transfer to an unseen embodiment and significantly raise the success rate for real-world policy performance. Our contributions are mainly threefold: 1) We find that IDM pre-training with heterogeneous kinematic annotations improves the transferable surgical actions representation learned by visual encoder. These improvements extend to downstream tasks and unseen embodiments; 2) We introduce SAR-IDE, a kinematics-grounded surgical video foundation model for learning generalizable instrument action representations. Our pre-training framework combines native kinematic supervision with tool-centric forward dynamics, in which the tool-centric future-state prediction provides auxiliary supervision in a shared pixel space, benefiting the shared representation learning from heterogeneous action space, while enabling learning from videos without kinematic annotations; 3) The SAR-IDE encoder achieves state-of-the-art performance among frozen visual encoders across action regression, action recognition, and instrument point tracking. Few-shot adaptation to unseen embodiments and improved needle-pickup success from 60% to 80% and inter-instrument transfer success from 0% to 30% in real-world robot experiments, further demonstrate the practical value of the learned representations.

2 Related Work

Surgical Motion and Video Pretraining. Video-pretrained surgical models learn spatiotemporal features that transfer to recognition tasks [4; 35; 49]. Motion2Vec [37] learns motion representations for action segmentation and pose imitation, while SurgMotion [46] incorporates visually derived motion cues. These studies establish visual and temporal transfer; we examine how synchronized kinematics from heterogeneous robots enrich transferable motion representations.

Learning from Heterogeneous Robot Embodiments. Open X-Embodiment [30] standardizes data formats and coarsely harmonizes actions for cross-robot policies. Octo [28] supports adaptation

to new observation and action spaces. HPT [42] shares a Transformer across embodiment-specific input stems and output heads. These approaches prioritize policy learning; we study transferable surgical motion features. TC-IDM [21] recovers executable trajectories using metric depth and camera poses. Such geometric grounding requires accurate registration, which is challenging in surgery: SurgCalib [48] documents cable-driven kinematic errors and calibration overhead. We retain native action conventions and use instrument dynamics as shared visual supervision, avoiding cross-dataset metric-frame alignment without assuming viewpoint invariance.

Action Representation Learning. Wu et al. [45] predict synchronized kinematics from optical flow for surgical gesture and skill recognition. DynaMo [8] couples IDM and FDM for encoder pretraining, with mixed gains from action supervision in downstream imitation learning. LAPA [50] learns discrete latent actions for policy pretraining. CLAM [17] learns continuous codes grounded by action-labeled play data. LAC-WM [12] uses latent actions for cross-embodiment world-model adaptation. LaWAM [7] predicts latent visual subgoals for policy guidance. WLA^3 [19] combines synchronized views, state changes, and native controls. LAOM [24] shows that inferred actions can encode distractors. LAOF [5] adds optical-flow constraints and object-region filtering. We combine heterogeneous surgical kinematics with promptable instrument-centric future prediction, and assess representation transfer through action regression, recognition, and point tracking.

3 METHOD

3.1 PROBLEM FORMULATION

SAR-IDE learns inverse dynamics across robot embodiments $\mathcal{E} = \{1, \dots, E\}$ with different kinematic conventions, action spaces, and camera configurations. A labeled sample $\xi = (e, \mathbf{o}_t, \mathbf{o}_{t+\Delta}, \mathbf{a}^{(e)}_{t+1:t+\Delta}) \sim \mathcal{D}_{\text{act}}$ contains endpoint observations separated by Δ steps and the measured trajectory connecting them, with each action $\mathbf{a}^{(e)}_{t+\ell} \in \mathbb{R}^{d_e}$. Each observation $\mathbf{o}_t = \{\mathbf{o}^{(v)}_t\}_{v \in \mathcal{V}_e}$ contains synchronized RGB views; $\mathbf{r}_{\text{view}} \in \{0,1\}^V$ marks available cameras in a common slot layout. The IDM maps direct encoder embeddings $\mathbf{H}_{t,\Delta}$ to a continuous latent action $\mathbf{z}_{t,\Delta}$ and then to native trajectories through embodiment-specific projectors. A tool-centric FDM regularizes the latent action through instrument-state prediction (Figure 2).

3.2 INVERSE DYNAMIC TRAINING

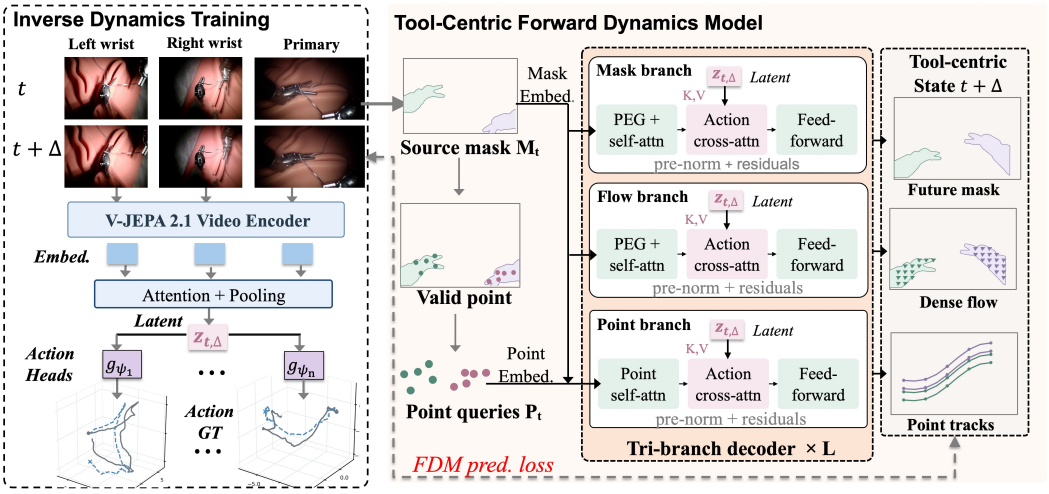

Figure 2: Overview of SAR-IDE and its training objectives.

For each camera view v, the V-JEPA 2.1 [22] encoder jointly processes the endpoint images. Its direct output tokens are $\mathbf{H}^{(v)}_{t,\Delta} \in \mathbb{R}^{N \times d}$; we denote their collection by $\mathbf{H}_{t,\Delta}$. This is the *encoder embedding*, before any IDM-specific view fusion or attentive pooling. The IDM then adds semantic view embeddings and applies masked self-attention across available camera slots. A set of Q

learned queries cross-attends to the fused tokens; the query features are averaged over valid views, concatenated, and transformed by a shared residual MLP. Denoting this entire post-encoder mapping by h_ϕ, we obtain

$$\mathbf{z}_{t,\Delta} = h_\phi(\mathbf{H}_{t,\Delta}; \mathbf{r}_{\text{view}}), \qquad \mathbf{z}_{t,\Delta} \in \mathbb{R}^{d_z}. \tag{1}$$

We use $Q = 4$ pooling queries and $d_z = 512$. The resulting $\mathbf{z}_{t,\Delta}$ is the *latent action*, distinct from the direct encoder embedding $\mathbf{H}_{t,\Delta}$.

Embodiment-specific action prediction. Each embodiment has a lightweight two-layer projector g_{ψ_e} that predicts its native trajectory, $\widehat{\mathbf{a}}^{(e)}_{t+1:t+\Delta} = g_{\psi_e}(\mathbf{z}_{t,\Delta})$. Trajectories are padded to a common width for batching; an action-validity mask $\mathbf{w}^{(e)}_{\text{act}}$ excludes padded or undefined coordinates from the per-sample loss:

$$\ell_{\text{act}}(\xi) = \frac{1}{\|\mathbf{w}^{(e)}_{\text{act}}\|_1} \sum_j w^{(e)}_{\text{act},j} \, \text{Huber}\Big(\widehat{a}^{(e)}_j - a^{(e)}_j\Big). \tag{2}$$

The targets are normalized with the training statistics of their source dataset. Only g_{ψ_e} is embodiment-specific; the video encoder, cross-view fusion, attentive pooler, and latent-action network are shared.

3.3 Tool-Centric Forward Dynamics Model Regularizer

The tool-centric FDM regularizes latent actions through a *Tri-branch decoder* for prompted instrument-state prediction. Instance-specific prompts restrict mask prediction and point tracking to selected instruments, reducing erroneous supervision from missing instance annotations.

Mask branch. Binary masks $\mathbf{M}_t$ of selected instruments replace RGB frames as spatial input, prompting the decoder to predict only their corresponding future masks $\widehat{\mathbf{M}}_{t+\Delta}$. The binary-classification loss ℓ_{mask} supervises only instances with valid correspondence. Mask-only inputs and targets exclude background appearance and focus supervision on instrument shape and location.

Point branch. We sample up to $N = 100$ valid points $\mathbf{P}_t = \{\mathbf{p}_{t,n}\}_{n=1}^N$ within the selected instance masks and use their initial positions as tracking queries, encoding each with normalized coordinates and local source-mask features. At each supervised time τ, the decoder predicts a unit direction $\widehat{\mathbf{d}}_{\tau,n}$ and non-negative magnitude $\widehat{s}_{\tau,n}$, yielding $\widehat{\mathbf{p}}_{\tau,n} = \mathbf{p}_{t,n} + \widehat{s}_{\tau,n}\widehat{\mathbf{d}}_{\tau,n}$. Trajectories span the action horizon relative to the initial query, without recursive frame-to-frame accumulation. The track loss ℓ_{track} combines direction, scale, and average-displacement errors on valid, persistently visible points.

Flow branch. To complement sparse tracks with dense motion, this branch predicts two-channel signed optical flow $\widehat{\mathbf{U}}_{t:t+\Delta-1}$ from the instrument-mask representation, conditioned on the same latent action. The loss ℓ_{flow} combines robust displacement regression and endpoint error over valid pixels of the prompted instruments. The three branch interfaces are

$$\begin{aligned} \widehat{\mathbf{M}}_{t+\Delta} &= F^{\text{mask}}_\omega(\mathbf{M}_t, \mathbf{z}_{t,\Delta}), \\ \widehat{\mathbf{P}}_{t:t+\Delta} &= F^{\text{point}}_\omega(\mathbf{M}_t, \mathbf{P}_t, \mathbf{z}_{t,\Delta}), \\ \widehat{\mathbf{U}}_{t:t+\Delta-1} &= F^{\text{flow}}_\omega(\mathbf{M}_t, \mathbf{z}_{t,\Delta}). \end{aligned} \tag{3}$$

The Tri-branch decoder uses Transformer blocks based on LAPA [50], with spatial self-attention, cross-attention to latent-action tokens, and feed-forward layers.

3.4 Joint Training with Missing Supervision

The training collection $\mathcal{D}$ contains partially overlapping subsets with action, mask, track, or flow supervision. For a sample $\xi \sim \mathcal{D}$, let r_{act}, r_{mask}, r_{track}, and r_{flow} indicate the availability of each target. To train SAR-IDE, we jointly optimize

$$\mathcal{L} = \mathbb{E}_{\xi\sim\mathcal{D}}\big[r_{\text{act}}\ell_{\text{act}} + \lambda_{\text{mask}} r_{\text{mask}}\ell_{\text{mask}} + \lambda_{\text{track}} r_{\text{track}}\ell_{\text{track}} + \lambda_{\text{flow}} r_{\text{flow}}\ell_{\text{flow}}\big]. \tag{4}$$

Every loss is normalized over its valid elements, so missing targets contribute neither loss nor gradient to the corresponding branch. Kinematics-rich OpenH samples train the IDM and, where annotations are available, the tool-centric FDM. Surgical videos without robot state set $r_{\text{act}} = 0$ and train the shared representation through instrument motion alone. Kinematic supervision therefore grounds $\mathbf{z}_{t,\Delta}$ in explicit robot motion, while the tool-centric FDM extends representation learning to videos for which no compatible action vector is available.

4 Experimental Setup

We evaluate transfer to action regression, action recognition, and keypoint tracking, and test the contribution of tool-centric FDM regularization through objective ablations.

Compared representations. We compare against V-JEPA 2.1 [22], DINOv3-L [36], DINO-Video-L [47], SurgMotion-G [46], EndoFM [44], EndoMamba [38], EndoViT [3], EndoSSL [11], SurgeNet [14], VideoMAEv2 [41], and GR00T-H [27], using the baselines compatible with each task's inputs and outputs. The visual encoder of SAR-IDE is initialized from the same pretrained checkpoint used by the V-JEPA 2.1 [22] baseline. Encoder and latent-action results are reported separately; internal variants are compared in Section 5.5.

Datasets and Implementation Details. We pretrain SAR-IDE using OpenH [29] video with synchronized kinematics and intraoperative video without kinematic annotations (see Appendix D for dataset composition and Appendix D.2 for durations and data curation). Few-shot action regression uses MedbotWorld [39] and SMARTS [51], both held out from pretraining. The 50 demonstrations for downstream control were collected separately on the same commercial endoscopic surgical platform as MedbotWorld [39]. Action recognition uses RARP50, GraSP, and the SMARTS [51] and TUD [34] subsets of OpenH, with the latter included in the ablation study. Keypoint tracking uses SurgPose. Representation probes freeze the pretrained features and use shared task targets and splits. Appendix D.4 details the recognition splits, label mapping, probe configurations, and checkpoint selection.

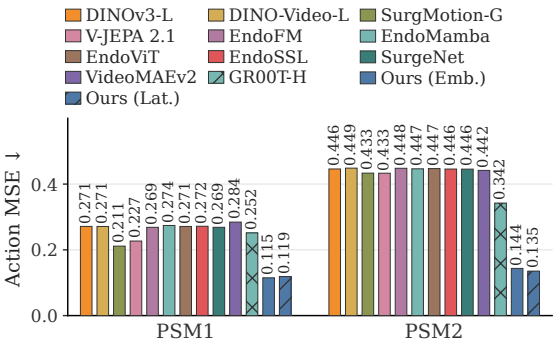

Figure 3: Action MSE (↓) for each PSM on SMARTS [51].

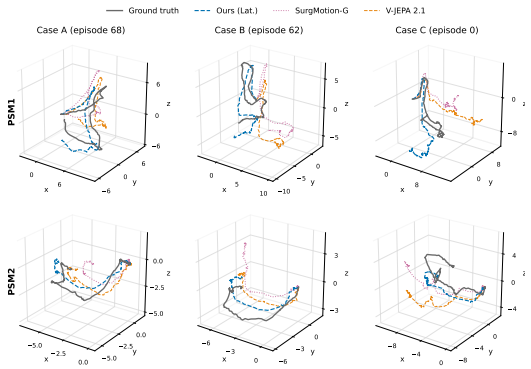

Figure 4: Accumulative rollouts using SAR-IDE (Lat.) on MedbotWorld [39].

5 Experimental Evaluation

We investigate whether the learned motion representation transfers across robot embodiments, retains semantic and dense motion information, and benefits from forward-dynamics regularization.

5.1 Few-Shot Action Regression

We test whether motion representations expose explicit kinematics under limited supervision. We split 70 episodes from MedbotWorld [39] into five adaptation episodes and 65 held-out test episodes to evaluate transfer to an unseen robot embodiment. We additionally evaluate transfer to SMARTS [51] (Figure 3), a domain excluded from pretraining, with five training procedures and 91 test procedures (5:91); all segments from a procedure remain in the same split. Comparison settings and evaluation details are provided in Appendix D.3.

Table 1 compares our encoder embeddings (Emb.) and latent actions (Lat.) on MedbotWorld [39]. Both representations substantially improve overall action-regression accuracy over the pretrained visual baselines. Together with the SMARTS [51] results, this trend suggests that inverse-dynamics pretraining makes kinematic information more accessible and supports few-shot transfer to a new domain or embodiment.

Method	PSM1			PSM2			Average		
	Trans. (mm)	Rot. (°)	Jaw (°)	Trans. (mm)	Rot. (°)	Jaw (°)	Trans. (mm)	Rot. (°)	Jaw (°)
V-JEPA 2.1 [22]	0.949	5.349	8.243	0.455	4.783	5.221	0.702	5.066	6.732
DINOv3-L [36]	1.254	6.034	9.257	0.703	5.373	7.041	0.979	5.704	8.149
DINO-Video-L [47]	1.223	6.287	12.052	0.671	5.825	7.620	0.947	6.056	9.836
SurgMotion-G [46]	0.989	5.318	7.988	0.465	4.763	5.345	0.727	5.041	6.667
GR00T-H [27]	1.368	6.381	8.661	0.847	5.735	6.553	1.108	6.058	7.607
EndoFM [44]	1.159	5.975	11.244	0.602	5.423	7.228	0.881	5.699	9.236
EndoMamba [38]	1.302	5.083	17.317	0.692	4.264	8.513	0.997	4.674	12.915
EndoViT [3]	1.373	6.119	16.195	0.741	5.529	8.831	1.057	5.824	12.513
EndoSSL [11]	1.299	5.092	17.034	0.685	4.220	8.567	0.992	4.656	12.801
SurgeNet [14]	1.296	5.924	12.586	0.695	5.292	8.193	0.996	5.608	10.390
VideoMAEv2 [41]	1.089	5.359	9.770	0.579	4.868	5.906	0.834	5.114	7.838
SAR-IDE (Emb.)	0.750	4.941	5.815	0.415	4.394	**4.404**	0.583	4.668	**5.110**
SAR-IDE (Lat.)	**0.671**	**4.158**	**5.494**	**0.391**	**3.439**	5.148	**0.531**	**3.799**	5.321

Table 1: MedbotWorld [39] few-shot action prediction from frozen representations; Average reports the arithmetic mean of PSM1 and PSM2 for each metric.

5.2 Real World Robot Policy Evaluation

Scene and robot setup. We follow the task setup of SutureBot [9], but collect demonstrations and evaluate policies on a different commercial surgical platform, anonymized for review (Figure 5, left). Episodes start with a needle on a suturing pad and both instruments above the workspace. Three RGB streams provide the left endoscopic view and left and right wrist-camera views. Camera placement, pad placement, and instrument starting positions remain consistent between collection and deployment, while initial needle placement varies.

Tasks and success criteria.

SutureBot [9] evaluates grasping and handoff together as needle pickup; we report the two stages separately within the continuous task illustrated in Figure 6. **Pickup** requires the left gripper to grasp the needle near its tip and lift it clear of the pad; success requires retaining the needle after lifting. **Transfer** begins only after successful pickup, from the instrument and needle configuration reached in that stage, with no reset or needle repositioning. The right gripper must grasp the needle near its base, after which the left gripper releases it. Transfer succeeds only when the needle remains securely held by the right gripper after the left gripper's release. Successful sequences typically finish within 25 s. Each trial has a shared 60 s timeout for the complete pickup–transfer sequence, with no timer reset between stages. A stage is counted as unsuccessful if the needle is dropped during that stage or its success condition is not reached by the timeout. Both pickup and transfer success rates use all 10 evaluation trials per condition as the denominator. A failed pickup also counts as a failed transfer.

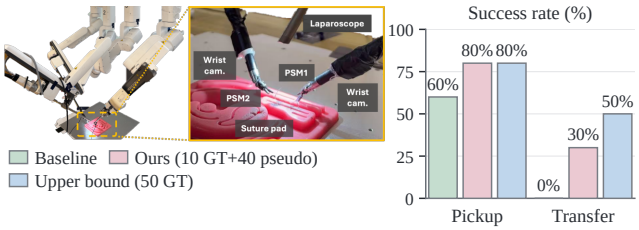

Figure 5: Robot setup (left) and success rates over 10 trials per condition (right).

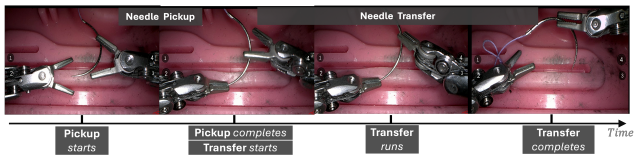

Figure 6: Illustration of the downstream robotic needle pickup and transfer task.

Evaluation details. We compare three supervision conditions using a pool of 50 demonstration episodes. The baseline trains on 10 episodes with ground-truth (GT) actions, and the fully supervised upper bound uses GT actions for all 50 episodes. For action-label augmentation, we fine-tune an MLP action head on the 10 GT episodes and use SAR-IDE together with this head to infer action labels for the remaining 40 episodes. The policy is then trained on the combined 10 GT-action and 40 pseudo-labeled trajectories.

Implementation details. As in action probing, the head predicts 15-step relative-action chunks from successive 16-frame video sequences. Each trajectory is initialized from its GT starting action pose; subsequent chunks use the preceding predicted endpoint as their reference, yielding an absolute trajectory through cumulative rollout (Appendix E). We convert the reconstructed trajectories to the observation-relative Cartesian action representation of GR00T-H [27] for policy training. All three policy conditions use the GR00T N1.6 [25] model version and initialize from the same OpenH-pretrained GR00T-H [27] weights before task-specific fine-tuning. The policy predicts chunks with actions spaced at 30 Hz, matching the robot control rate; the 55–75 ms single-call inference latency refers to generating a chunk, not an individual action. Model, optimization, and deployment details are provided in Appendix J.

Results. Figure 5 (right) shows that augmentation with model-inferred action labels improves pickup and transfer success over the 10-episode baseline. It matches the fully supervised upper bound on pickup and enables successful transfers where the baseline achieves none, although a gap remains to the upper bound on transfer. These results support the feasibility of using inferred action labels to improve downstream control when only a small set of action-labeled demonstrations is available.

5.3 Surgical Action Recognition

All recognition methods, including Emb. and Lat., use frozen representations with the same attentive classifier and evaluation protocol within each dataset. RARP50 and SMARTS [51] measure fine-grained segment-level recognition, while GraSP measures frame-centric multi-label action presence (Table 2).

5.3.1 Fine-Grained Action Recognition

RARP50 and SMARTS provide case-disjoint segment-level action labels. RARP50 is seen only through tool-centric FDM supervision and provides no kinematic targets, whereas SMARTS is held out entirely from our pretraining. We evaluate frozen representations using the protocol in Appendix D.4. For the representation visualization, we additionally include Turin, an OpenH subset used during pretraining with synchronized kinematics, so that the three datasets cover kinematically supervised, visually supervised, and unseen domains.

Table 2: Surgical action recognition (%, ↑): accuracy, macro-F1, and GraSP mAP.

Method	SAR-RARP50		SMARTS		GraSP
	Acc.	F1	Acc.	F1	mAP
V-JEPA 2.1 [22]	85.00	65.50	83.55	76.69	39.85
DINOv3-L [36]	74.32	46.78	66.45	56.61	31.78
DINO-Video-L [47]	69.32	47.92	51.32	38.27	30.90
SurgMotion-G [46]	80.00	58.50	73.03	61.90	37.51
GR00T-H [27]	74.32	52.10	88.82	86.23	28.91
EndoFM [44]	58.41	36.56	55.92	47.14	31.73
EndoMamba [38]	36.59	20.08	28.95	14.00	22.39
EndoViT [3]	59.77	38.26	40.79	33.82	30.29
EndoSSL [11]	45.68	26.56	34.21	17.21	25.36
SurgeNet [14]	56.59	32.15	57.89	45.90	28.21
VideoMAEv2 [41]	67.05	47.49	53.95	47.12	31.65
SAR-IDE (Emb.)	**86.59**	**66.27**	**91.45**	**89.18**	**43.16**
SAR-IDE (Lat.)	78.64	53.01	90.79	87.01	32.35

Table 2 shows that SAR-IDE (Emb.) achieves the highest accuracy and F1-score on both RARP50 and SMARTS. The gain on RARP50 indicates that tool-centric visual supervision transfers to fine-grained action recognition even without kinematic labels for that dataset. The stronger result on the unseen SMARTS domain further shows that the benefit is not restricted to videos observed during pretraining. The compact latent representation remains competitive on SMARTS but is weaker on RARP50, indicating that the encoder retains action-discriminative visual structure that is not fully preserved by the action bottleneck. Figure 7 further visualizes the frozen encoder geometry with t-SNE on the two OpenH action subsets and RARP50. Across these complementary supervision regimes, our representation forms more coherent action-specific neighborhoods than the comparison encoders. The improvement is most apparent for actions that occur in similar procedural phases and involve similar instrument appearances, indicating that the learned representation captures motion-dependent action cues beyond coarse phase or instrument identity.

5.3.2 GraSP Action Presence Recognition

GraSP [2] contains robot-assisted radical-prostatectomy videos with frame-level annotations of 14 non-exclusive atomic actions. We formulate a video-only action-presence task: for each annotated keyframe, the target is the union of the actions performed by all visible instruments, without using

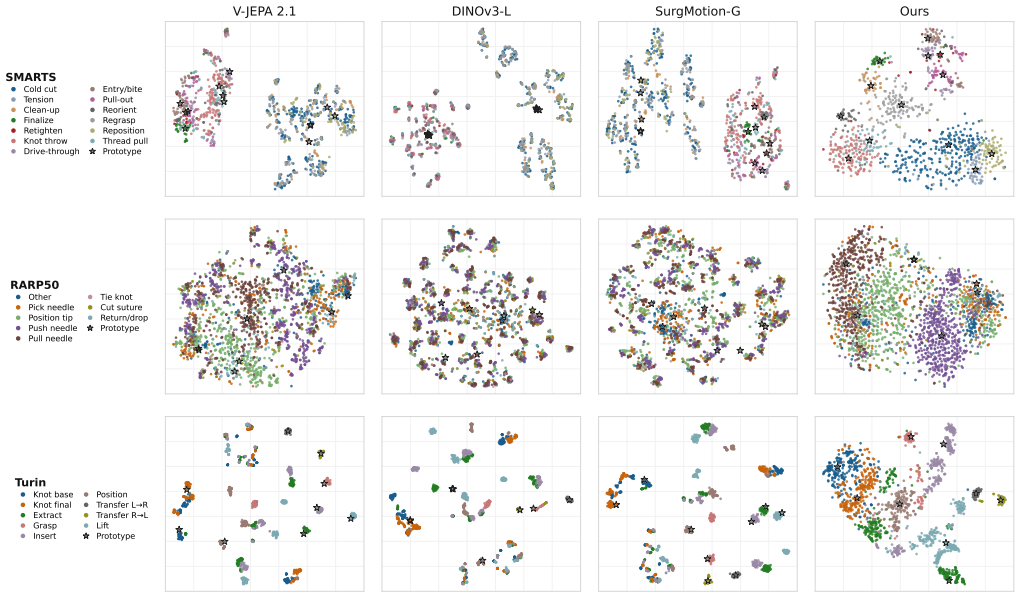

Figure 7: t-SNE visualization of frozen encoder representations for surgical action recognition.

instrument boxes, masks, types, surgical phases, or steps as probe inputs. All representations are frozen and evaluated using the same attentive multi-label probe; Appendix D.5 provides the protocol details.

Table 2 summarizes GraSP mAP; Appendix D.5 provides complete class-wise results. The encoder representation of SAR-IDE achieves the strongest action-presence result, indicating that tool-centric motion pretraining improves the accessibility of fine-grained surgical action cues under a shared probe. The latent bottleneck is less effective on this semantic, spatially grounded task, suggesting that part of the information useful for action presence remains distributed across encoder tokens rather than being fully retained in the compact latent action.

5.4 Surgical Point Tracking

Table 3: SurgPose endpoint EPE (↓) in 384-grid pixels at five temporal gaps (frames); the first group contains specialized zero-shot references, and bold compares representation probes.

Model	4	8	12	16	30
CoTracker3 [15]	1.152	2.396	3.978	6.088	19.844
SEA-RAFT [43]	1.404	3.059	5.455	8.821	24.605
DINOv3-L [36]	1.875	3.356	5.061	7.219	18.769
DINO-Video-L [47]	4.529	7.846	11.035	14.042	25.675
VideoMAEv2 [41]	1.717	3.798	6.345	9.631	26.030
EndoFM [44]	2.995	5.202	7.666	10.533	24.375
EndoMamba [38]	3.044	5.527	8.245	11.387	25.972
EndoViT [3]	2.521	4.378	6.475	8.881	20.835
EndoSSL [11]	2.863	5.169	7.674	10.541	24.030
SurgeNet [14]	2.792	4.883	7.155	9.823	23.421
SurgMotion-G [46]	2.032	3.724	5.577	7.793	19.966
GR00T-H [27]	3.771	5.955	8.108	10.425	21.123
V-JEPA 2.1 [22]	1.450	2.685	4.154	6.036	15.955
SAR-IDE	**1.295**	**2.245**	**3.226**	**4.331**	**10.069**

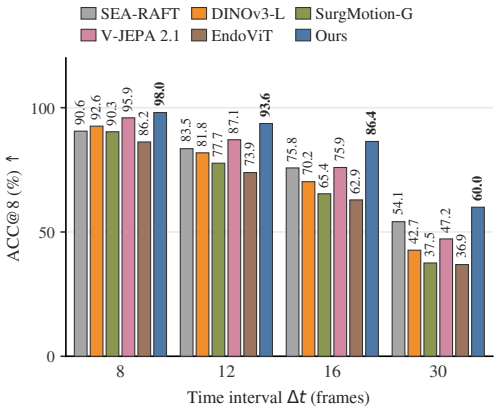

Figure 8: SurgPose accuracy at 8 pixels (%, ↑): SAR-IDE compared with the best four baselines and SEA-RAFT.

We train a common DPT-based [33] displacement decoder on frozen multi-level encoder features, sampling endpoint gaps of 1–30 frames. Evaluation uses stride-one windows from 24 held-out cases. All methods use only the two endpoint frames as visual input and are scored on the same points visible at both endpoints. CoTracker3 [15] and SEA-RAFT [43] serve as specialized zero-shot references

under this sparse-input protocol. EPE and ACC@8 are reported in a common 384×384 coordinate system.

Table 3 shows that SAR-IDE has the lowest EPE among the representation probes at every interval, with a clearer advantage over V-JEPA 2.1 [22] as temporal distance grows. Figure 8 shows that SAR-IDE also leads the representation probes in ACC@8 at 8–30-frame gaps. Together, these results support improved transfer of instrument-motion features, particularly over longer temporal gaps.

Appendix F provides complete results, evaluation details, comparisons with specialized trackers, and qualitative examples (Figure 10).

5.5 Ablation Studies

Table 4: Ablation on action regression, action recognition, and keypoint tracking. PSM1 and PSM2 report MSE in OpenH action space. Recognition reports test accuracy and macro-F1 averaged over SMARTS [51] and TUD [34], while tracking reports SurgPose Acc@8 and Acc@16 averaged over temporal intervals.

	Action regression		Action recognition		Keypoint tracking	
Model	PSM1	PSM2	Acc. ↑	F1 ↑	Acc@8 ↑	Acc@16 ↑
IDM	0.072	0.055	86.53	86.65	82.16	94.88
w/o Kine.	0.908	0.846	79.60	75.99	80.48	94.06
Latent FDM	0.054	0.040	85.92	82.98	82.95	94.94
SAR-IDE	**0.050**	**0.038**	**90.06**	**88.79**	**84.52**	**95.11**

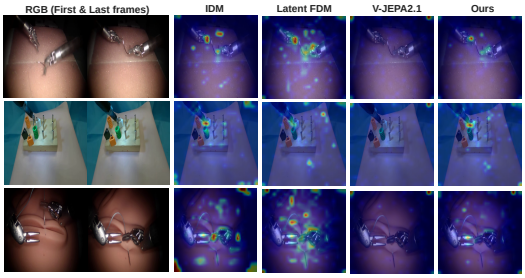

Figure 9: Encoder attention heatmaps across OpenH embodiments.

Table 4 examines kinematic supervision and the choice of forward-prediction target. Latent FDM uses V-JEPA 2.1 [22] features as its initial-state input and future reconstruction target, predicting whole-scene features, including instruments and background, conditioned on the inferred latent action. Our tool-centric FDM instead predicts instrument masks, point tracks, and flow. Removing kinematic supervision markedly weakens action regression. Latent FDM improves regression over kinematics-only IDM but underperforms SAR-IDE across regression, recognition, and tracking. These trends support the complementary roles of kinematic grounding and instrument-focused future prediction. Evaluation details are provided in Appendix D.6.

In the OpenH attention examples in Figure 9, Latent FDM shows more scattered background responses, while SAR-IDE focuses more strongly on instrument tips, jaws, and contact regions. This pattern suggests that whole-scene reconstruction retains action-irrelevant visual changes, whereas tool-centric targets encourage instrument-focused motion features. The task-conditioned Grad×Act analysis in Appendix G likewise shows stronger instrument alignment along the full action-prediction pathway.

6 Conclusion

We presented SAR-IDE, a surgical video foundation model that learns transferable motion-aware embeddings through inverse-dynamics pretraining. Our central finding is that synchronized kinematics from heterogeneous robots can strengthen a shared visual encoder even when their action coordinates differ. Embodiment-specific heads provide supervision in native action spaces, while the auxiliary tool-centric FDM links inferred actions to instrument motion in observation space and enables learning from videos without kinematic labels. Frozen-representation evaluations demonstrate substantial gains in action-regression accuracy, improved surgical action recognition and point tracking, and few-shot transfer to an unseen embodiment. Ablations support the complementary roles of kinematic grounding and instrument-focused future prediction. These representations also provide useful action labels for policy learning: adding 40 pseudo-labeled demonstrations to 10 GT-action demonstrations raises needle-pickup success from 60% to 80% and inter-instrument transfer success from 0% to 30%, measured over 10 trials per condition on a real surgical robot. Together, these results show how inverse-dynamics pretraining with heterogeneous kinematics and unlabeled video can improve surgical representations and support downstream robot learning.

A Ethics Statement

This study uses existing surgical-video and robot-kinematics datasets, with sources and curation documented in Appendices D and D.2. Some sources contain intraoperative recordings; these are distinct from our separately collected robot demonstrations. Our real-world policy evaluation consists of bench-top needle pickup and transfer on a suturing pad, rather than clinical interventions. Access to and reuse of each source dataset remain subject to its provider's conditions. The reported results evaluate representation learning and controlled robot tasks; they do not establish clinical safety or effectiveness. Patient-facing use would require separate validation and appropriate oversight.

B Reproducibility Statement

Section 3 specifies the model architecture, action representations, and training objectives. Sections 4 and 5 describe the downstream tasks and evaluation settings. Appendix D documents the pretraining data, while Appendix D.2 details annotation generation and source-level split separation. The few-shot regression protocol in Appendix D.3 specifies adaptation splits, frozen components, head training, and error definitions; Appendix E defines the action coordinates and trajectory reconstruction. Complete tracking results and evaluation coverage appear in Appendix F. Appendix J reports the demonstration allocation and model and training configurations for downstream robot policies. Dataset access follows the original providers' conditions in Appendix D.3.

C AI Use Statement

We used LLM-based tools for literature search, manuscript polishing, and the preparation and revision of plotting code and figures. Writing assistance was limited to language refinement, including grammar, readability, and clarity. LLM assistance did not extend to method implementation, experimental design, or interpretation of results.

The checks performed on AI-assisted figure edits included comparing numerical values with the supplied experimental results, verifying axis ranges and label placement in the plotting code, and inspecting the compiled manuscript pages. Models used to produce training annotations and inferred action labels are described separately in Appendices D.2 and J. The authors take responsibility for the manuscript, code, reported results, and conclusions, including material prepared with LLM assistance.

REFERENCES

[1] Oluwatosin Alabi, Ko Ko Zayar Toe, Zijian Zhou, Charlie Budd, Nicholas Raison, Miaojing Shi, and Tom Vercauteren. CholecInstanceSeg: A tool instance segmentation dataset for laparoscopic surgery. *Scientific Data*, 12(1):825, 2025. doi: 10.1038/s41597-025-05163-w. 2, 15

[2] Nicolás Ayobi, Santiago Rodríguez, Alejandra Pérez, Isabela Hernández, Nicolás Aparicio, Eugénie Dessevres, Sebastián Peña, Jessica Santander, Juan Ignacio Caicedo, Nicolás Fernández, and Pablo Arbeláez. Pixel-wise recognition for holistic surgical scene understanding. *arXiv preprint arXiv:2401.11174*, 2024. 2, 7, 15

[3] Dominik Batić, Felix Holm, Ege Özsoy, Tobias Czempiel, and Nassir Navab. EndoViT: pretraining vision transformers on a large collection of endoscopic images. *International Journal of Computer Assisted Radiology and Surgery*, 19:1085–1091, 2024. 1, 5, 6, 7, 8, 17, 18, 19, 20

[4] Leonardo Borgioli, Neil Getty, Wenli Xiu, Jessica Cassiani, Alvaro Ducas, Carlos Agustin Orda, Hira Waris, Fangfang Xia, Rick Stevens, Pier Cristoforo Giulianotti, and Milos Zefran. OmniRAS: Standardizing foundation model training and evaluation in robot-assisted surgery. *arXiv preprint arXiv:2608.31048*, 2026. 2

[5] Xizhou Bu, Jiexi Lyu, Fulei Sun, Ruichen Yang, Zhiqiang Ma, and Wei Li. LAOF: Robust latent action learning with optical flow constraints. *arXiv preprint arXiv:2511.16407*, 2025. 3

[6] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719. 16

[7] Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. LaWAM: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026. 3

[8] Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. *arXiv preprint arXiv:2409.12192*, 2024. 3

[9] Jesse Haworth, Juo-Tung Chen, Nigel Nelson, Ji Woong Kim, Masoud Moghani, Chelsea Finn, and Axel Krieger. SutureBot: A precision framework & benchmark for autonomous end-to-end suturing. *arXiv preprint arXiv:2510.20965*, 2025. 1, 6, 22, 24

[10] Yufan He, Pengfei Guo, Mengya Xu, Zhaoshuo Li, Andriy Myronenko, Dillan Imans, Bingjie Liu, Dongren Yang, Mingxue Gu, Yongnan Ji, Yueming Jin, Ren Zhao, Baiyong Shen, and Daguang Xu. SurgWorld: Learning surgical robot policies from videos via world modeling. *arXiv preprint arXiv:2512.23162*, 2025. URL https://arxiv.org/abs/2512.23162v2. 1

[11] Roy Hirsch, Mathilde Caron, Regev Cohen, Amir Livne, Ron Shapiro, Tomer Golany, Roman Goldenberg, Daniel Freedman, and Ehud Rivlin. Self-supervised learning for endoscopic video analysis. In *Medical Image Computing and Computer Assisted Intervention*, pp. 569–578, 2023. 1, 5, 6, 7, 8, 17, 18, 19, 20

[12] Huang Huang, Sriram Yenamandra, Arjun Majumdar, Elie Aljalbout, Tushar Nagarajan, Jimmy Yang, Akshara Rai, Michael Rabbat, Li Fei-Fei, Jiajun Wu, Tingfan Wu, and Franziska Meier. Cross-embodiment robot foundation world models with latent actions. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=S30Y1sBz34. 3

[13] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. URL https://arxiv.org/abs/2505.12705. 1

[14] Tim J. M. Jaspers, Ronald L. P. D. de Jong, Yiping Li, Carolus H. J. Kusters, Franciscus H. A. Bakker, Romy C. van Jaarsveld, Gino M. Kuiper, Richard van Hillegersberg, Jelle P. Ruurda, Willem M. Brinkman, Josien P. W. Pluim, Peter H. N. de With, Marcel Breeuwer, Yasmina Al Khalil, and Fons van der Sommen. Scaling up self-supervised learning for improved surgical foundation models. *Medical Image Analysis*, 108:103873, 2026. doi: 10.1016/j.media.2025.103873. 5, 6, 7, 8, 17, 18, 19, 20

[15] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 8, 16, 20, 23

[16] Ji Woong Kim, Tony Z. Zhao, Samuel Schmidgall, Anton Deguet, Marin Kobilarov, Chelsea Finn, and Axel Krieger. Surgical robot transformer (SRT): Imitation learning for surgical tasks. *arXiv preprint arXiv:2407.12998*, 2024. 1, 18

[17] 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. 3

[18] Haofeng Liu, Erli Zhang, Junde Wu, Mingxuan Hong, and Yueming Jin. Surgical SAM 2: Real-time segment anything in surgical video by efficient frame pruning. *arXiv preprint arXiv:2408.07931*, 2024. 16

[19] Peidong Liu, Zhiyuan Xiang, Mingyang Li, Wenhao Li, Jiale Zhang, Jiahao Sun, and Jiawei Li. WLA^3: World latent action modeling for semantics, dynamics, and kinematics. *arXiv preprint arXiv:2609.15870*, 2026. 3

[20] Meta Superintelligence Labs. SAM 3.1: Object multiplex. Official model release notes, March 2026. URL https://github.com/facebookresearch/sam3/blob/main/RELEASE_SAM3p1.md. Released March 27, 2026. 16

[21] Weishi Mi, Yong Bao, Xiaowei Chi, Xiaozhu Ju, Zhiyuan Qin, Kuangzhi Ge, Kai Tang, Peidong Jia, Shanghang Zhang, and Jian Tang. TC-IDM: Grounding video generation for executable zero-shot robot motion. *arXiv preprint arXiv:2601.18323*, 2026. 3

[22] Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. 3, 5, 6, 7, 8, 9, 17, 18, 19, 20

[23] Dujun Nie, Fengjiao Chen, Qi Lv, Jun Kuang, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. LARY: A latent action representation yielding benchmark for generalizable vision-to-action alignment. *arXiv preprint arXiv:2604.11689*, 2026. 1, 26

[24] Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025. 1, 3

[25] NVIDIA. GR00T-N1.6-3B. Model card, 2025. URL https://huggingface.co/nvidia/GR00T-N1.6-3B. 7, 24

[26] NVIDIA. Cosmos-H-Surgical-Simulator. Model card, 2026. URL https://huggingface.co/nvidia/Cosmos-H-Surgical-Simulator. 1

[27] NVIDIA. GR00T-H. Model card, 2026. URL https://huggingface.co/nvidia/GR00T-H. 5, 6, 7, 8, 17, 18, 19, 20, 24

[28] Octo Model Team. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 2

[29] Open-H-Embodiment Consortium. Open-H-Embodiment: A large-scale dataset for enabling foundation models in medical robotics. *arXiv preprint arXiv:2604.21017*, 2026. 1, 2, 5, 15, 16

[30] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023. 2

[31] Dimitrios Psychogyios, Emanuele Colleoni, Beatrice Van Amsterdam, Chih-Yang Li, Shu-Yu Huang, Yuchong Li, Fucang Jia, Baosheng Zou, Guotai Wang, Yang Liu, Maxence Boels, Jiayu Huo, Rachel Sparks, Prokar Dasgupta, Alejandro Granados, Sébastien Ourselin, Mengya Xu, An Wang, Yanan Wu, Long Bai, Hongliang Ren, Atsushi Yamada, Yuriko Harai, Yuto Ishikawa, Kazuyuki Hayashi, Jente Simoens, Pieter DeBacker, Francesco Cisternino, Gabriele Furnari, Alex Mottrie, Federica Ferraguti, Satoshi Kondo, Satoshi Kasai, Kousuke Hirasawa, Soohee Kim, Seung Hyun Lee, Kyu Eun Lee, Hyoun-Joong Kong, Kui Fu, Chao Li, Shan An, Stefanie Krell, Sebastian Bodenstedt, Nicolas Ayobi, Alejandra Perez, Santiago Rodriguez, Juanita Puentes, Pablo Arbelaez, Omid Mohareri, and Danail Stoyanov. SAR-RARP50: Segmentation of surgical instrumentation and action recognition on robot-assisted radical prostatectomy challenge. *arXiv preprint arXiv:2401.00496*, 2024. 2, 15

[32] Sanat Ramesh, Vinkle Srivastav, Deepak Alapatt, Tong Yu, Aditya Murali, Luca Sestini, Chinedu Innocent Nwoye, Idris Hamoud, Saurav Sharma, Antoine Fleurentin, Georgios Exarchakis, Alexandros Karargyris, and Nicolas Padoy. Dissecting self-supervised learning methods for surgical computer vision. *Medical Image Analysis*, 88:102844, 2023. 1

[33] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021. 8

[34] Ariel Rodriguez, Rayan Younis, Lorenzo Mazza, Ortrun Hellig, Martin Wagner, and Stefanie Speidel. TUD & NCT Dresden Robot Assistant Dataset. Open-H-Embodiment dataset, 2026. URL `https://huggingface.co/datasets/nvidia/PhysicalAI-Robotics-Open-H-Embodiment/blob/main/Surgical/tud/260131_tundra_dataset/grasping_retraction/README.md`. TUNDRA dataset. CC BY 4.0. Accessed September 25, 2026. 5, 9, 18

[35] Samuel Schmidgall, Ji Woong Kim, Jeffrey Jopling, and Axel Krieger. General surgery vision transformer: A video pre-trained foundation model for general surgery. *arXiv preprint arXiv:2403.05949*, 2024. 2

[36] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. 5, 6, 7, 8, 17, 18, 19, 20

[37] Ajay Kumar Tanwani, Pierre Sermanet, Andy Yan, Raghav Anand, Mariano Phielipp, and Ken Goldberg. Motion2Vec: Semi-supervised representation learning from surgical videos. *arXiv preprint arXiv:2006.00545*, 2020. 2

[38] Qingyao Tian, Huai Liao, Xinyan Huang, Bingyu Yang, Dongdong Lei, Sebastien Ourselin, and Hongbin Liu. EndoMamba: An efficient foundation model for endoscopic videos. *arXiv preprint arXiv:2502.19090*, 2025. 5, 6, 7, 8, 17, 18, 19, 20

[39] toumai-eco. MedbotWorld. Hugging Face dataset, 2026. URL `https://huggingface.co/datasets/toumai-eco/MedbotWorld`. CC BY-NC 4.0. Accessed September 25, 2026. 5, 6, 16, 17, 22, 26

[40] Andru P. Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy. EndoNet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, 2017. 2, 15

[41] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. VideoMAE V2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023. 5, 6, 7, 8, 17, 18, 19, 20

[42] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *arXiv preprint arXiv:2409.20537*, 2024. 3

[43] Yihan Wang, Lahav Lipson, and Jia Deng. SEA-RAFT: Simple, efficient, accurate raft for optical flow. *arXiv preprint arXiv:2405.14793*, 2024. 8, 16, 19, 20

[44] Zhao Wang, Chang Liu, Shaoting Zhang, and Qi Dou. Foundation model for endoscopy video analysis via large-scale self-supervised pre-train. In *Medical Image Computing and Computer Assisted Intervention*, pp. 101–111, 2023. 1, 5, 6, 7, 8, 17, 18, 19, 20

[45] Jie Ying Wu, Aniruddha Tamhane, Peter Kazanzides, and Mathias Unberath. Cross-modal self-supervised representation learning for gesture and skill recognition in robotic surgery. *International Journal of Computer Assisted Radiology and Surgery*, 16(5):779–787, 2021. doi: 10.1007/s11548-021-02343-y. 3

[46] Jinlin Wu, Felix Holm, Chuxi Chen, An Wang, Yaxin Hu, Xiaofan Ye, Zelin Zang, Miao Xu, Lihua Zhou, Huai Liao, Danny T. M. Chan, Ming Feng, Wai S. Poon, Hongliang Ren, Dong Yi, Nassir Navab, Gaofeng Meng, Jiebo Luo, Hongbin Liu, and Zhen Lei. SurgMotion: A video-native foundation model for universal understanding of surgical videos. *arXiv preprint arXiv:2602.05638*, 2026. 1, 2, 5, 6, 7, 8, 17, 18, 19, 20

[47] Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From foundation to application: Improving VLA models in practice. *arXiv preprint arXiv:2607.06403*, 2026. 5, 6, 7, 8, 17, 18, 19, 20

[48] Zijian Wu, Shuojue Yang, Yu Chung Lee, Eitan Prisman, Yueming Jin, and Septimiu E. Salcudean. SurgCalib: Gaussian splatting-based hand-eye calibration for robot-assisted minimally invasive surgery. *arXiv preprint arXiv:2603.08983*, 2026. 3

[49] Shu Yang, Fengtao Zhou, Leon Mayer, Fuxiang Huang, Yiliang Chen, Yihui Wang, Sunan He, Yuxiang Nie, Xi Wang, Ömer Sümer, Yueming Jin, Huihui Sun, Shuchang Xu, Alex Qinyang Liu, Zheng Li, Jing Qin, Jeremy YuenChun Teoh, Lena Maier-Hein, and Hao Chen. Large-scale self-supervised video foundation model for intelligent surgery. *arXiv preprint arXiv:2506.02692*, 2025. 2

[50] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June 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*, 2025. 1, 3, 4

[51] Haoying Zhou, Peter Kazanzides, Chang Liu, Junlin Wu, Yimeng Wu, and Hongjun Wu. SurgSyncExt: Time-synchronized multi-modal dataset for surgical robotics. Open-H-Embodiment dataset, 2025. URL `https://huggingface.co/datasets/nvidia/PhysicalAI-Robotics-Open-H-Embodiment/blob/main/Surgical/jhu/lcsr/smarts/README.md`. SMARTS dataset. CC BY 4.0. Accessed September 25, 2026. 5, 7, 9, 16, 17, 18, 26

[52] Haoying Zhou, Chang Liu, Yimeng Wu, Junlin Wu, Zijian Wu, Yu Chung Lee, Sara Martuscelli, Septimiu E. Salcudean, Gregory S. Fischer, and Peter Kazanzides. SurgSync: Time-synchronized multi-modal data collection framework and dataset for surgical robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL `https://arxiv.org/abs/2603.06919`. 17

D Pretraining Data and Evaluation Details

D.1 Pretraining Data Composition

Our completed kinematics-supervised IDM pretraining runs use 38 distinct OpenH [29] dataset subsets across nine embodiment configuration groups (Table 5). These groups distinguish acquisition sites and camera setups, including a monocular variant. The declared training partitions of these sources contain approximately 462 hours of video. This duration sums unique episode lengths after metadata-level exclusions and before runtime validity, clutch, and training-window filtering, counting SutureBot once across its two configurations.

We pretrain SAR-IDE for 12,000 optimizer steps on 32 GPUs, with approximately 33 hours of total training wall time.

Table 5: OpenH datasets used for kinematics-supervised IDM pretraining: nine embodiment configuration groups and 38 distinct dataset subsets.

Embodiment group	Included datasets	Entries
CMR Versius	Hysterectomy; inguinal hernia; prostatectomy	3
JHU IMERSE dVRK	ARCADE cholecystectomy; Cao cautery; needle pickup/handover; tissue lift; wound closure; SutureBot; NephFat	7
JHU IMERSE dVRK (mono)	SutureBot (monocular configuration)	1
Stanford dVRK	Needle transfer; tissue retraction; peg transfer	3
Obuda dVRK	FRS Dome 1; needle threading 1/2; peg transfer 1/2; rollercoaster 1; sea spike 1/2/3; pork 1; skin phantom 1	11
Hamlyn dVRK	Suturing 1/2; peg transfer; needle grasp/handover; knot tying; tissue lifting; tissue retraction	7
UCSD dVRK	Surgical learning datasets 1/2	2
UCB dVRK	Debridement	1
Turin MITIC	Ex-vivo; plastic pad; plastic pad (3DMED); plastic tube	4
Total training entries (nine groups)		39
Distinct datasets after deduplication		38

SutureBot is registered under both JHU groups, yielding 39 training entries but 38 distinct datasets. Configured sources that contributed no training samples are excluded.

Intraoperative video without kinematics. The training-partition inventory contains 22.4 hours of intraoperative surgical video from SAR-RARP50 [31], GraSP [2], and Cholec80 [40], with the latter paired with instrument-instance annotations from CholecInstanceSeg [1]. Without synchronized robot kinematics, these samples contribute through latent-action learning with tool-centric forward prediction rather than supervised kinematic regression. We do not use their phase/action labels for pretraining. Learning is therefore unsupervised with respect to action labels, while instrument masks provide visual supervision for the tool-centric FDM. Training-only clip counts and durations are reported in Table 6.

D.2 Annotation and Data Curation

Dataset durations. The full OpenH video–kinematics inventory contains approximately 462 hours in its declared training partitions, counted before runtime window filtering. Table 6 reports the portion with valid visual annotations, alongside the intraoperative video sources. The annotated OpenH subset contains 27.42 hours across 30 of the 38 subsets; these hours are already included in the 462-hour inventory. SAR-RARP50, GraSP, and Cholec contribute 3.17, 14.69, and 4.55 hours of annotated training-partition video, respectively, totaling 22.41 hours. Durations sum clip lengths at each OpenH subset's frame rate and at 10 Hz for the intraoperative clips, counting each primary-view timeline once without duplicating camera views or overlapping windows. Clip counts describe continuous segments, not independent surgeries or repeated training draws.

Split separation. For the intraoperative inventory, we match converted clips to their original case or video IDs and retain the 35 SAR-RARP50 and six GraSP cases used for downstream training, together with ten Cholec training videos. These source cases and videos are disjoint from the corresponding validation and test partitions. Exclusion is at the source-video level, so all clips and derived annotations from a held-out source are excluded together; withholding action labels alone would not remove visual overlap.

Table 6: Training-only visual annotation coverage, estimated from valid clip lengths. Hours count each source timeline once; validation and test data are excluded.

Collection	Clips	Frames	Hours
OpenH (annotated subset)	7 198	2 955 157	27.42
SAR-RARP50	1 163	114 132	3.17
GraSP	5 413	528 946	14.69
Cholec80 / CholecInstanceSeg	1 817	163 753	4.55
Intraoperative subtotal	8 393	806 831	22.41
Total annotated training coverage	15 591	3 761 988	49.83

Instrument segmentation. We use SAM-family video segmenters: SAM 3 [6] for OpenH and a surgical-domain adaptation of SAM 3.1 [6; 20] for Cholec. We follow the mask-prompted propagation paradigm used in Surgical SAM 2 [18]. Keyframe masks anchor instrument identities, and temporal propagation supplies masks for intervening frames. For Cholec, we map the sparse CholecInstanceSeg annotations to the corresponding Cholec80 frames and densify approximately 1 Hz annotations to 10 Hz. We retain instrument IDs shared by neighboring keyframes, re-anchor propagation at each keyframe, and preserve the annotated endpoint masks. OpenH retains its dataset-specific frame rates. Propagated masks are pseudo-labels rather than additional manual ground truth; invalid clips and unreliable instance correspondences are excluded from visual supervision.

Point tracking. CoTracker3 [15] supplies instrument-point trajectories for both OpenH and Cholec. Queries are sampled inside each instrument mask, with increased density near a geometric tip proxy while retaining samples along the instrument body. We estimate the dominant mask axis by singular value decomposition of centered foreground coordinates and use its image-center-facing end as the tip proxy. Visibility and instance-validity masks restrict supervision to reliable trajectory segments. These point trajectories are distinct from the dense optical-flow annotations.

Idle detection and removal. For Cholec, we identify idle intervals from instrument motion without phase or action labels. At one-second intervals, we measure the median endpoint displacement over 0.6 s (seven frames at 10 Hz) using flow-derived instrument-point tracks from SEA-RAFT [43]. Only points that remain valid within the same instrument instance throughout the interval contribute. A median displacement below four pixels in the annotation-image coordinates marks a low-motion interval; missing or invalid tracks are not themselves evidence of inactivity. Within each continuous source sequence, we bridge isolated one-second gaps between low-motion intervals and remove the resulting idle runs lasting at least two seconds. Remaining active intervals are divided into 50–100-frame clips (5–10 s), discarding fragments shorter than 50 frames. The Cholec duration in Table 6 is measured after this filtering.

Dense flow and temporal alignment. SEA-RAFT [43] estimates optical flow between adjacent RGB frames; source instrument masks restrict the pixels retained for flow supervision. We align annotations by source-frame index and join temporally adjacent segments, removing duplicated boundary frames. Within each continuous clip, stride-one windows assemble aligned masks, point trajectories, and successive flow fields, preserving point-query identities and avoiding temporal gaps. This per-step assembly provides dense temporal supervision; it does not imply recursive rollout by the point decoder.

D.3 Few-Shot Action-Regression Protocol

Splits and comparison scope. MedbotWorld [39] evaluates transfer to an unseen embodiment: we partition its 70 episodes into five adaptation episodes and 65 test episodes using seed 42. SMARTS [51] evaluates transfer to a held-out domain within OpenH [29], using a procedure-disjoint 5:91 train/test split. With split seed 42, we select two training procedures from P1, two from

P2, and one from P3; all segments of a procedure stay together. The resulting training and test sets contain 104 and 983 segments, respectively. This regression split is distinct from the segment-level action-recognition split. Figure 3 compares V-JEPA 2.1 [22], DINOv3-L [36], DINO-Video-L [47], SurgMotion-G [46], EndoFM [44], EndoMamba [38], EndoViT [3], EndoSSL [11], SurgeNet [14], VideoMAEv2 [41], GR00T-H [27], SAR-IDE (Emb.), and SAR-IDE (Lat.).

Dataset sources. MedbotWorld [39] is distributed through Hugging Face under CC BY-NC 4.0 with gated file access; we cite the dataset page as its source. SMARTS [51] is the SurgSyncExt contribution to OpenH, distributed under CC BY 4.0 and collected using the synchronized acquisition framework SurgSync [52].

Frozen representations and heads. Encoder-level comparisons freeze the pretrained backbone and train an attentive action decoder on its output tokens. For SAR-IDE (Emb.), only the encoder weights are transferred; the fusion, pooling, and action-head modules are initialized anew. For SAR-IDE (Lat.), the encoder and pretrained latent-action pathway remain frozen, and only a newly initialized embodiment-specific output projector is trained. The two variants therefore measure encoder transfer and latent-action transfer, respectively, rather than identical-head comparisons. DINO-Video-L [47] is the video teacher released with LingBot-VLA 2.0 [47]; we use its pretrained video encoder as a frozen representation. For GR00T-H [27], both regression datasets use its frozen final visual-language embeddings with the action decoder instead of its original policy head.

SMARTS inputs and optimization. All methods use windows defined by endoscopic frames at offsets 0 and 5 and predict action offsets 1–5, spanning 0.5 seconds at 10 Hz. The two wrist-view slots are blank and masked. DINOv3-L [36] extracts target-frame patch tokens, while DINO-Video-L [47] supplies causally encoded endpoint tokens from the frame pair. Encoder probes share a two-layer attention, attentive pooling, and MLP head design, with backbone-specific input widths. Inputs are 384×384 for V-JEPA 2.1 [22], DINOv3-L [36], and SAR-IDE, and 256×256 for DINO-Video-L [47] and SurgMotion-G [46]. Heads are trained for 1,000 optimizer steps with effective batch size 32, using learning rates 3×10^{-5} for encoder probes and 3×10^{-4} for the latent-action projector. Every method uses its fixed step-1,000 checkpoint, without test-set checkpoint selection. Evaluation uses stride five and the same 18,425 valid test windows. Each chunk is independently anchored to the ground-truth initial action, with no accumulated rollout.

SMARTS action space and error definitions. Each arm contributes three relative translation coordinates, six relative rotation coordinates, and one absolute jaw coordinate, following Appendix E. Translation and jaw are standardized using training-only, horizon-dependent statistics shared across methods; rotation-6D coordinates are left unchanged. Normalized Pose MSE averages squared errors over the nine pose coordinates and evaluated time steps, while normalized Jaw MSE averages the single jaw channel separately, for PSM1 and PSM2. Padding is excluded. These are model-space errors, not millimeter or angular errors; quaternion sign alignment does not enter the rotation-6D calculation. An MSE pooled over all ten channels measures combined action error and must not be interpreted as pose-only error.

D.4 Action-Recognition Evaluation Protocol

Splits and labels. RARP50 uses its official case-disjoint train/validation/test split of 1,545/234/440. SMARTS [51] uses 709/130/152 segments, with three knot-throw labels merged into 13 classes. For both datasets, probe configurations and checkpoints are selected on validation and evaluated once on the test set.

Representations and probes. Encoder probes use direct backbone outputs ($\mathbf{H}$ for ours), and latent-action probes use $\mathbf{z}$. Both are frozen, projected to a shared 512-dimensional width, and passed to the same attentive classifier. All methods follow the same dataset-specific splits, probe-training protocol, and validation-based checkpoint selection. GR00T-H [27] uses its frozen native VL tokens with a fixed task prompt and no robot-state input. SMARTS [51] uses the 709/130/152 train/validation/test split, with checkpoint selection by 15-class validation macro-F1 and post-hoc 13-class mapping of predicted and target labels.

D.5 Complete GraSP Action-Presence Results

Table 7 reports mAP and all 14 class-wise AP values for the methods in Table 2.

The official case-disjoint benchmark comprises eight development procedures with 2,324 annotated keyframes and five held-out test procedures with 1,125 keyframes. Each sample uses nine temporally ordered RGB frames from a 32-frame neighborhood around the annotated keyframe, producing eight adjacent frame pairs.

We freeze every representation, project its tokens to 512 dimensions, and train the same attentive multi-label probe with 14 sigmoid outputs and binary cross-entropy. For each action class, average precision is computed over all held-out keyframes, and the reported mAP is the unweighted mean of the 14 class-wise AP values.

Table 7: GraSP action-presence recognition. mAP is followed by the AP of each action class. *Open-S* denotes Open Something; higher is better.

Method	mAP	Caut.	Close	Cut	Grasp	Hold	Open	Open-S	Pull	Push	Rel.	Still	Suct.	Travel	Other
V-JEPA 2.1 [22]	39.85	56.99	14.43	17.80	12.28	85.13	10.94	10.40	23.95	**46.48**	**15.03**	97.99	**72.80**	79.86	13.79
DINOv3-L [36]	31.78	26.32	11.73	12.74	7.11	83.52	5.10	7.84	15.02	28.69	1.89	95.50	70.01	63.16	16.22
DINO-Video-L [47]	30.90	41.38	12.98	16.44	8.08	80.28	5.06	4.36	13.87	26.50	1.32	94.94	65.09	59.91	2.33
SurgMotion-G [46]	37.51	47.89	15.79	24.94	11.34	84.77	6.75	9.59	25.09	34.36	7.80	97.76	67.48	79.71	11.86
GR00T-H [27]	28.91	26.21	6.82	14.16	7.08	**86.95**	4.55	7.86	15.87	27.90	1.69	94.94	34.21	60.95	15.49
EndoFM [44]	31.73	31.93	14.07	12.31	9.23	78.24	5.67	3.74	14.12	29.45	3.81	95.43	54.28	66.73	25.27
EndoMamba [38]	22.39	11.59	8.87	7.14	5.53	71.30	5.04	2.47	8.97	15.33	1.23	89.05	24.17	59.87	2.90
EndoViT [3]	30.29	33.04	7.92	14.03	5.21	77.84	4.90	5.40	10.62	27.51	1.29	94.19	59.40	58.90	23.82
EndoSSL [11]	25.36	28.09	7.42	8.14	4.19	76.83	5.70	3.03	8.37	19.39	1.28	90.73	43.50	56.24	2.06
SurgeNet [14]	28.21	22.87	8.62	10.12	5.29	76.13	5.83	11.73	12.46	21.90	1.44	93.95	40.10	58.37	**26.08**
VideoMAEv2 [41]	31.65	41.04	9.92	14.15	5.42	76.07	6.14	8.77	17.40	32.40	3.84	96.32	42.75	77.10	11.73
SAR-IDE (Emb.)	**43.16**	**61.63**	**20.25**	**38.75**	**18.60**	86.83	**18.18**	**31.75**	**28.69**	45.88	11.61	**98.54**	59.61	**81.82**	2.10
SAR-IDE (Lat.)	32.35	28.61	16.51	24.91	15.30	76.50	6.32	6.41	28.44	36.29	4.83	97.00	30.66	76.90	4.15

D.6 Ablation Evaluation Protocol

For the OpenH action-regression ablation, all models receive the same ground-truth-anchored chunks from 30 jointly covered test datasets spanning six embodiments and predict offsets 1–15 from frames 0 and 15. MSE is computed before denormalization over all valid translation, 6D rotation, and gripper channels for each PSM. For recognition, we use SMARTS [51] and TUD [34] because they provide temporally localized action labels, whereas many other OpenH subsets annotate workflow or phase, or define tasks closely coupled to scene context. The latter denotes the grasping-and-retraction subset of the TUD & NCT Dresden Robot Assistant (TUNDRA) dataset [34], released through OpenH under CC BY 4.0. All representations are frozen and use the same attentive classifier and dataset-specific evaluation protocol. SMARTS [51] uses 13 classes after merging the three knot-throw labels, and TUD [34] uses five classes; test accuracy and macro-F1 are averaged equally across the two datasets. Tracking uses the protocol in Section 5.4, with ACC@8 and ACC@16 averaged equally over gaps of 8, 12, 16, and 30 frames, excluding the four-frame interval.

E Action-Space Implementation for SAR-IDE

Relative pose. We follow the hybrid-relative formulation of SRT [16], using the Cartesian pose conversion in GR00T-H [27] with an action-based reference. For each arm, let $(\mathbf{x}_k, \mathbf{R}_k, q_k)$ denote the recorded action at time k: end-effector position, orientation, and the available jaw/gripper command. For a chunk beginning at t, the unnormalized targets are

$$\begin{aligned} \Delta\mathbf{x}_{t,k} &= \mathbf{x}_{t+k} - \mathbf{x}_t, & \Delta\mathbf{R}_{t,k} &= \mathbf{R}_t^\top \mathbf{R}_{t+k}, \\ \mathbf{a}_{t+k} &= \left[\Delta\mathbf{x}_{t,k}^\top,\ \rho(\Delta\mathbf{R}_{t,k})^\top,\ q_{t+k}\right]^\top, & k &= 1, \ldots, \Delta. \end{aligned} \tag{5}$$

Translation is a direct difference in the dataset's native Cartesian axes; rotation is expressed relative to the first action's end-effector orientation. The 6D representation $\rho(\mathbf{R})$ concatenates the first two columns of $\mathbf{R}$. The gripper channel q_{t+k} remains absolute, retaining its native jaw-angle or command convention. Other arm-joint measurements remain state channels, not joint-displacement targets in

this Cartesian action space. Each arm contributes nine pose coordinates and, when available, its gripper channel; arm outputs are concatenated. Translation and gripper targets are normalized using source-dataset training statistics, while the 6D rotation coordinates are unchanged.

Action-anchored chunks and rollout. We use the first action in each chunk as the reference; the standard GR00T-H conversion uses the observation pose. In our 16-frame setting, action offset 0 supplies the common reference for all supervised offsets 1–15. For long-trajectory reconstruction, we initialize at the first action pose and recover each chunk as

$$\widehat{\mathbf{x}}_{t+k} = \widehat{\mathbf{x}}_t + \widehat{\Delta\mathbf{x}}_{t,k}, \qquad \widehat{\mathbf{R}}_{t+k} = \widehat{\mathbf{R}}_t \widehat{\Delta\mathbf{R}}_{t,k}, \tag{6}$$

where predictions are denormalized and relative rotation matrices are recovered from 6D outputs by Gram–Schmidt orthonormalization. The predicted endpoint anchors the next chunk. An observation-based reference would include action–observation offsets in each chunk's targets; repeatedly composing these offsets can distort the accumulated trajectory. Action anchoring avoids reintroducing this reference mismatch at successive chunk boundaries.

F Complete SurgPose Point-Tracking Results

Tables 8 and 10 report complete EPE and threshold accuracies under the protocol in Section 5.4. ACC@τ measures the percentage of annotated point pairs with endpoint error strictly below $\tau \in \{4, 8, 16\}$ pixels on the common 384-grid. Table 9 gives the shared evaluation coverage.

Table 8: SurgPose results on a common 384-grid. Each cell reports EPE (↓) / ACC@8 (%, ↑). The first row is a specialized zero-shot reference; bold compares representation probes. The dash denotes unavailable common-grid ACC@8.

Model	4	8	12	16	30
SEA-RAFT [43]	1.404 / 96.80	3.059 / 90.56	5.455 / 83.51	8.821 / 75.77	24.605 / 54.14
DINOv3-L [36]	1.875 / 99.27	3.356 / 92.58	5.061 / 81.84	7.219 / 70.23	18.769 / 42.67
DINO-Video-L [47]	4.529 / 84.98	7.846 / 64.05	11.035 / 49.91	14.042 / 41.07	25.675 / 23.83
VideoMAEv2 [41]	1.717 / 98.66	3.798 / 87.62	6.345 / 73.10	9.631 / 60.68	26.030 / 32.86
EndoFM [44]	2.995 / 96.03	5.202 / 81.40	7.666 / 66.87	10.533 / 54.78	24.375 / 28.18
EndoMamba [38]	3.044 / 94.90	5.527 / 78.91	8.245 / 64.90	11.387 / 53.36	25.972 / 28.70
EndoViT [3]	2.521 / 97.74	4.378 / 86.21	6.475 / 73.90	8.881 / 62.90	20.835 / 36.89
EndoSSL [11]	2.863 / 96.10	5.169 / 80.91	7.674 / 66.77	10.541 / 55.57	24.030 / 30.37
SurgeNet [14]	2.792 / 96.83	4.883 / 82.89	7.155 / 69.46	9.823 / 58.00	23.421 / 32.43
SurgMotion-G [46]	2.032 / –	3.724 / 90.32	5.577 / 77.67	7.793 / 65.36	19.966 / 37.53
GR00T-H [27]	3.771 / 92.50	5.955 / 75.52	8.108 / 60.61	10.425 / 48.43	21.123 / 24.78
V-JEPA 2.1 [22]	1.450 / 99.68	2.685 / 95.93	4.154 / 87.11	6.036 / 75.94	15.955 / 47.21
w/o Kinematics	1.360 / 99.64	2.495 / 96.54	3.686 / 89.70	4.941 / 80.63	10.618 / 55.04
SAR-IDE	**1.295 / 99.81**	**2.245 / 98.05**	**3.226 / 93.64**	**4.331 / 86.41**	**10.069 / 59.96**

Columns denote temporal gaps Δt in frames.

Table 9: SurgPose evaluation coverage shared by all methods at all five temporal gaps. A visible point pair is one annotated query point that is visible at both endpoints of a sliding window.

Δt	Sliding windows	Visible point pairs
4	23,928	267,647
8	23,832	265,706
12	23,736	263,793
16	23,640	261,920
30	23,304	255,659

Evaluation uses points visible at both endpoints, without predicted-visibility filtering. Native-256 errors from SurgMotion-G [46] and DINO-Video-L [47] are scaled by $383/255$ before computing EPE and threshold accuracies; percentages are not rescaled. SurgMotion-G lacks common-grid accuracies at four frames. DINOv3-L [36] and DINO-Video-L use native-input preprocessing.

For EndoFM [44], EndoMamba [38], EndoViT [3], EndoSSL [11], SurgeNet [14], VideoMAEv2 [41], and GR00T-H [27], only the common DPT head is trained: ten training cases, 5,000 updates,

Table 10: SurgPose ACC@τ (%, ↑) on the common 384-grid. Each model has rows for $\tau = 4, 8, 16$ pixels; columns denote temporal gaps in frames. The first group contains specialized zero-shot references; bold compares representation probes, and dashes indicate unavailable accuracy.

Model	τ	4	8	12	16	30
CoTracker3 [15]	4	96.81	90.93	85.39	80.16	61.58
	8	98.33	94.05	89.93	85.12	66.91
	16	99.30	95.98	92.57	89.02	72.29
SEA-RAFT [43]	4	93.32	83.97	75.29	67.51	47.04
	8	96.80	90.56	83.51	75.77	54.14
	16	99.11	95.00	89.75	83.63	62.16
DINOv3-L [36]	4	91.34	71.13	54.32	42.61	21.72
	8	99.27	92.58	81.84	70.23	42.67
	16	99.98	99.39	96.12	89.91	67.26
DINO-Video-L [47]	4	56.53	33.93	22.46	17.03	8.24
	8	84.98	64.05	49.91	41.07	23.83
	16	98.77	87.86	77.20	69.26	50.34
VideoMAEv2 [41]	4	91.73	68.65	50.68	38.33	17.30
	8	98.66	87.62	73.10	60.68	32.86
	16	99.90	97.82	90.92	81.47	55.20
EndoFM [44]	4	75.92	51.30	36.00	26.25	11.25
	8	96.03	81.40	66.87	54.78	28.18
	16	99.81	96.67	89.25	80.54	55.15
EndoMamba [38]	4	75.51	50.85	36.05	26.73	12.24
	8	94.90	78.91	64.90	53.36	28.70
	16	99.74	95.11	86.60	78.17	53.91
EndoViT [3]	4	82.97	60.99	45.71	35.15	16.66
	8	97.74	86.21	73.90	62.90	36.89
	16	99.90	97.90	92.26	85.42	62.56
EndoSSL [11]	4	77.90	53.18	37.79	27.84	12.31
	8	96.10	80.91	66.77	55.57	30.37
	16	99.81	96.41	88.58	80.22	55.95
SurgeNet [14]	4	79.25	55.11	39.89	30.08	14.45
	8	96.83	82.89	69.46	58.00	32.43
	16	99.83	97.35	90.63	82.72	58.47
SurgMotion-G [46]	4	–	66.35	48.86	36.98	17.69
	8	–	90.32	77.67	65.36	37.53
	16	–	99.10	95.49	88.99	63.89
GR00T-H [27]	4	63.25	38.95	26.15	18.90	7.89
	8	92.50	75.52	60.61	48.43	24.78
	16	99.84	96.86	89.97	81.60	56.54
V-JEPA 2.1 [22]	4	95.75	79.73	62.70	49.98	25.64
	8	99.68	95.93	87.11	75.94	47.21
	16	99.99	99.67	98.01	93.23	70.48
w/o Kinematics	4	96.11	81.89	66.01	54.57	31.75
	8	99.64	96.54	89.70	80.63	55.04
	16	99.98	99.77	99.10	97.04	80.34
SAR-IDE	4	**97.28**	**85.75**	**71.50**	**59.47**	**34.88**
	8	**99.81**	**98.05**	**93.64**	**86.41**	**59.96**
	16	**99.99**	**99.91**	**99.53**	**98.10**	**82.89**

effective batch size eight, and 24 held-out cases. SurgeNet uses SurgeNetXL (CAFormer-S18), and VideoMAEv2 uses the released distilled ViT-B/16 checkpoint. GR00T-H uses image-token grids from four frozen VL layers; endpoint grids are spatially aligned and channel-concatenated for the DPT head.

The zero-shot references use fixed CoTracker3 [15] (scaled-offline) and SEA-RAFT [43] (Spring-M) weights with 384×384 inputs; SEA-RAFT samples forward flow at each query. CoTracker3 has the lowest EPE at four frames, while SAR-IDE has lower EPE at gaps of 8–30. CoTracker3 retains higher ACC@8 at 30 frames (Table 10), showing that mean error and threshold accuracy capture different aspects of tracking quality.

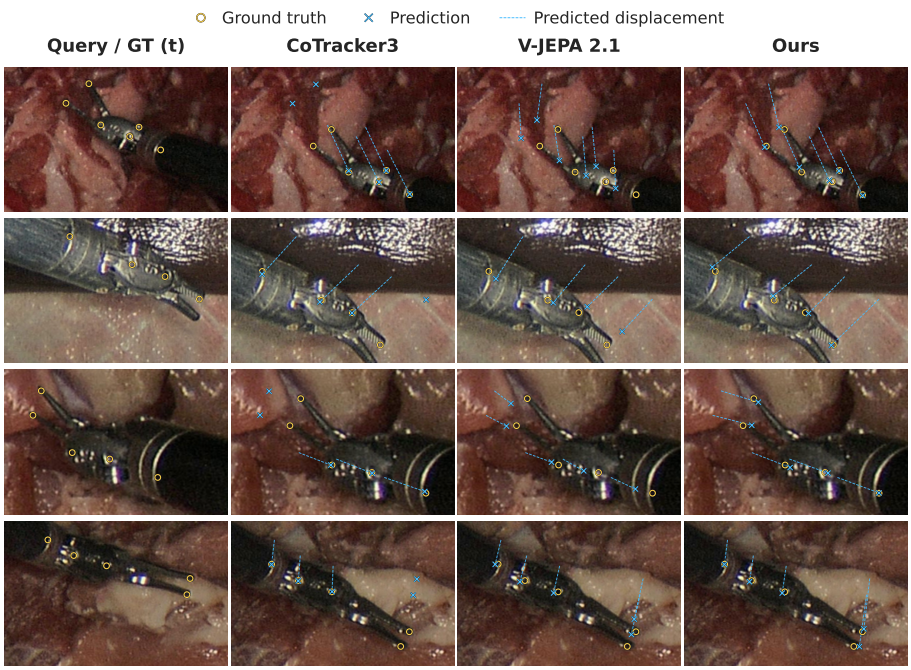

Figure 10: Instrument-centric examples on held-out SurgPose videos at gaps of 12 or 16 frames.

Qualitative point localization. Figure 10 illustrates instrument-point localization on held-out videos at gaps of 12 or 16 frames.

G Instrument-Aligned Action Attribution

We use Grad×Act to analyze which visual tokens support the predicted actions. This is a task-conditioned attribution analysis of the full action pathway, rather than a direct visualization of encoder self-attention.

Protocol. We select four OpenH training subsets jointly covered by all three models. Every model receives the same frame pair (offsets 0 and 15), episode, step, and view mask. Inputs are resized to 384×384, yielding a 24×24 token grid. Grad×Act is computed for the grouped PSM1/PSM2 translation and jaw outputs. SAM3 instrument masks are used only to measure attribution alignment and are not model inputs.

Table 11: OpenH subsets used for the matched Grad×Act analysis.

Training subset	FPS	Episodes	View mask
JHU needle pickup/handover	30	192	[T, T, T]
Stanford needle transfer	30	498	[T, F, F]
Hamlyn peg transfer	30	207	[T, T, T]
Turin MITIC plastic pad	30	394	[T, F, F]

The protocol produces 128 candidate anchor–target groups. After removing four anchors without valid masks, the analysis contains 28 anchors and 112 matched action-target cases, or 336 attribution maps across the three models.

Compared with IDM and Latent FDM, SAR-IDE assigns more attribution to instrument regions and their immediate neighborhoods, and its attribution peak more often falls on an instrument. Since SAR-IDE does not have the most concentrated attribution map, the stronger alignment reflects greater instrument selectivity rather than uniformly sharper responses.

Table 12: Instrument alignment of action attribution. "Nearby" dilates the instrument support by one token on the 24×24 grid. Top-5% mass measures attribution concentration and is not an alignment metric.

Model	On instrument	Instrument nearby	Peak on instrument	Top-5% mass
IDM	27.4%	54.6%	67.0%	0.568
Latent FDM	21.3%	48.2%	68.8%	0.404
SAR-IDE	**37.3%**	**65.0%**	**92.0%**	0.523

These results support the interpretation that SAR-IDE produces more instrument-aligned action attribution. They do not isolate encoder attention because Grad×Act also reflects the downstream action head.

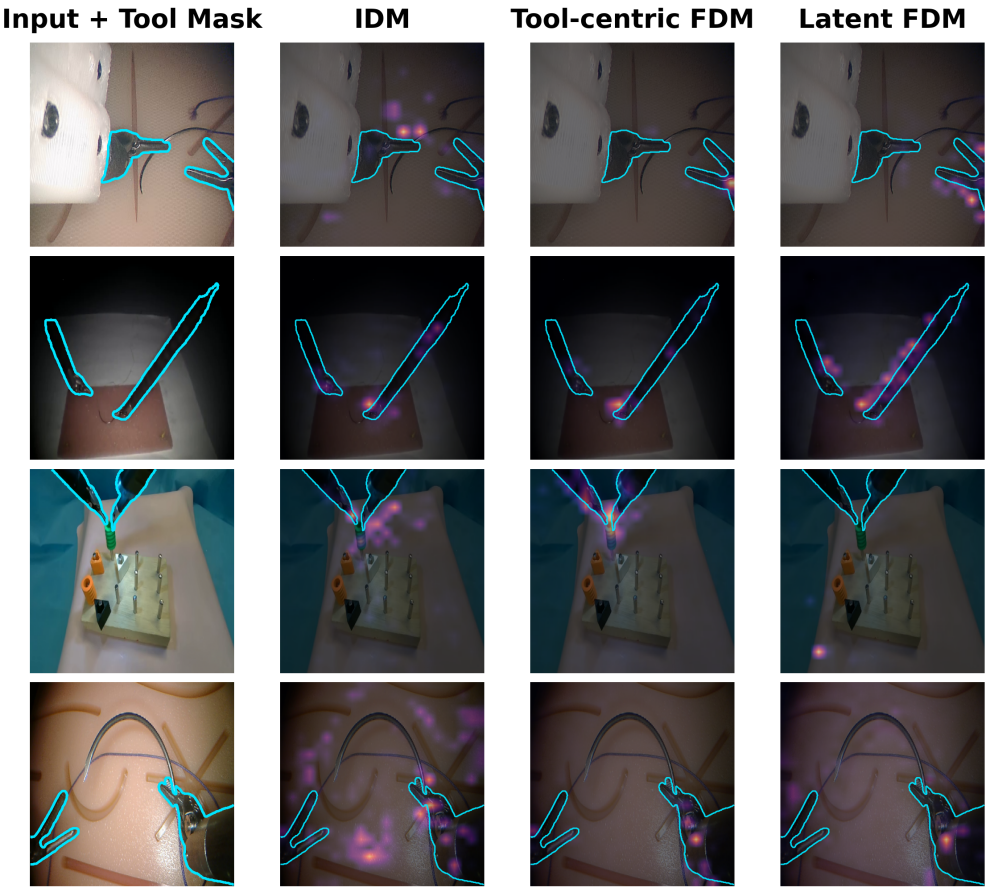

Figure 11: Matched Grad×Act maps for PSM1 translation across four OpenH embodiments. Cyan contours show the SAM3 instrument masks used for evaluation. The examples follow the fixed sampling protocol.

H Visualization of FDM Output

Figure 12 illustrates FDM outputs for seven examples from MedbotWorld [39], Hamlyn, Turin, and SutureBot [9]. IDM infers the conditioning latent action from both observed endpoints, A and $A+16$. The first three sources contribute test records, while the last contributes training records.

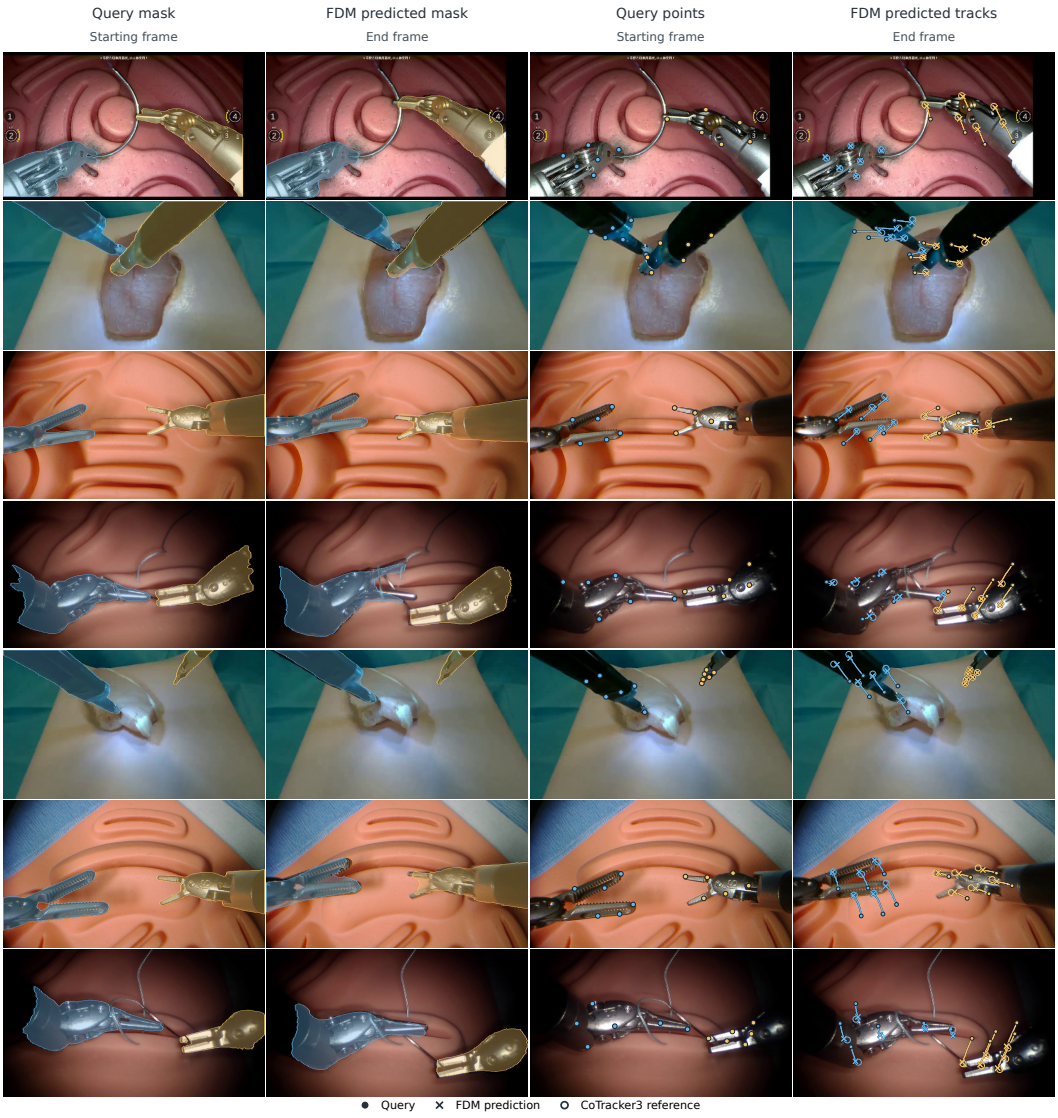

Figure 12: FDM output visualization. Each row is one case. From left to right: SAM3 query mask at A, FDM-predicted mask over $A + 16$, query points at A, and FDM-predicted tracks over $A + 16$. Blue and gold distinguish the two input instrument slots, with six query points per slot. Filled circles mark queries, crosses mark the last native FDM outputs, and hollow circles mark CoTracker3 [15] pseudo-reference endpoints. Solid paths contain the 16 native outputs (training span A–$A + 15$); CoTracker3 uses 17 observed frames A–$A + 16$. Endpoint positions provide a qualitative comparison across these horizons.

I Cross-Sample Visualization of High-Similarity Latent Actions

Figure 13 visualizes JHU IMERSE–Hamlyn sample pairs selected for high cosine similarity between their latent-action vectors from SAR-IDE. The paired transitions show related tissue-lifting and retraction behaviors across recording setups. We use latent similarity to identify examples for qualitative inspection of the observed instrument motion.

Both datasets use dVRK, but their selected recordings have opposite screen-side PSM assignments: JHU places PSM2 on the left and PSM1 on the right, whereas Hamlyn places PSM1 on the left and PSM2 on the right. The similar visible behaviors in these high-similarity pairs suggest that the learned latent actions associate related instrument motion across these recording configurations.

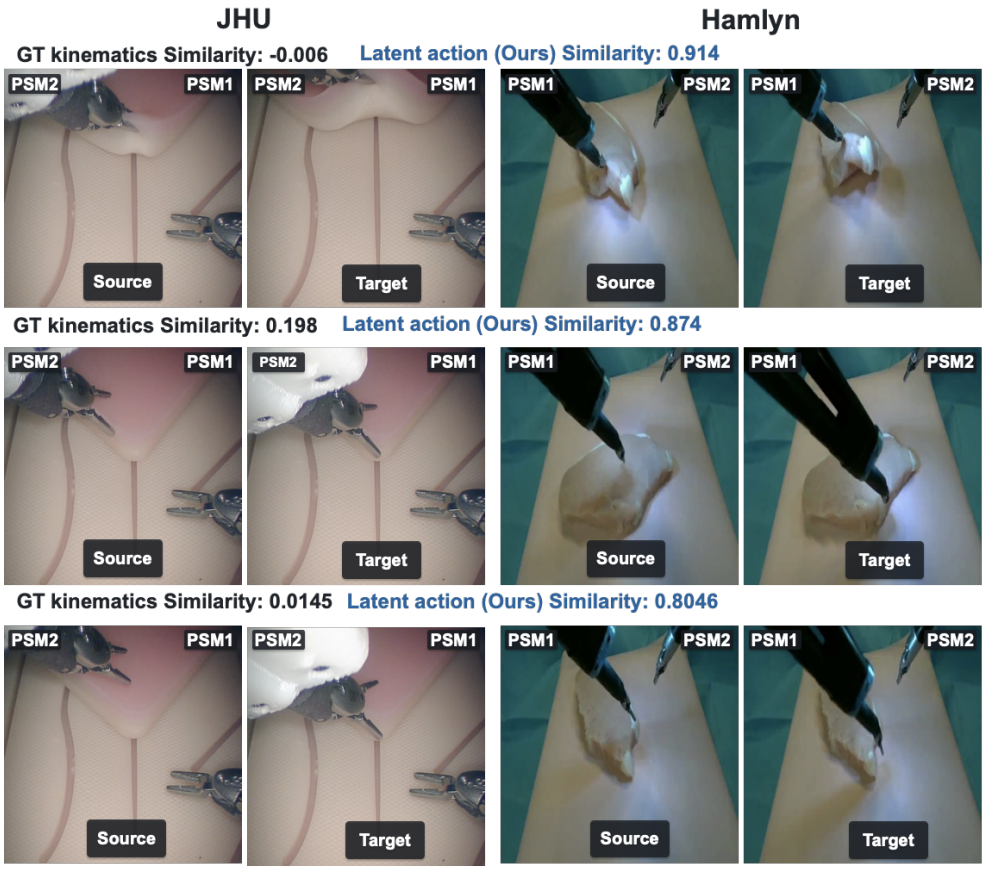

Figure 13: Cross-sample visualizations with high latent-action similarity. Each row pairs JHU IMERSE (left two panels) with Hamlyn (right two panels); columns show source and target observations. PSM labels denote the screen-side assignments. The selected pairs illustrate related tissue-lifting and retraction behaviors across recording setups.

J Real-World Policy Implementation

Demonstration collection and policy evaluation follow the task setup of SutureBot [9] on a different commercial surgical platform, anonymized for review.

Implementation details. All three real-world policy conditions use the GR00T N1.6 [25] model version, initialized with the same GR00T-H [27] weights pretrained on OpenH before task-specific fine-tuning. We freeze SAR-IDE and train a randomly initialized, embodiment-specific two-layer MLP with ReLU on the 10 action-labeled demonstration episodes. The frozen model encodes two endpoint frames (offsets 0 and 15) from the endoscope and two wrist cameras; the MLP maps its 512-dimensional latent action to a 15-step action chunk. Its predictions provide action supervision for 40 additional demonstrations, which are combined with the 10 GT-action episodes to train the policy. Absolute trajectories reconstructed by cumulative rollout (Appendix E) are re-encoded for policy training following GR00T-H [27]: translation differences and 6D relative rotations use the current observation pose as their reference, while gripper commands remain absolute. Tables 13 and 14 summarize the model, training, and deployment configuration.

Table 13: Model and training configuration for real-world policy learning.

Setting	Action head	Policy
Model parameters		
Model / version	MLP	GR00T N1.6
Pretrained checkpoint	SAR-IDE (frozen)	GR00T-H (OpenH-pretrained)
VLM backbone	–	`nvidia/Eagle-Block2A-2B-v2`
Image resolution	384×384	256×256
Observation horizon	2 endpoint frames	1
State / action dimensions	– / 40 (padded)	128 / 128
Action representation	Relative action, ROT6D	Relative action, ROT6D
Prediction horizon	15 steps	50 steps
Hidden dimension	512	1024
Number of layers	2	32
Inference sampling steps	–	4
Training parameters		
Trainable components	MLP	Projector + diffusion model + VLLN
GT-labeled episodes	10	10 / 10 / 50
Pseudo-labeled episodes	0	0 / 40 / 0
Training GPU count	2	4
Global batch size	32	32
Optimizer	AdamW	AdamW
Learning rate	3×10^{-4}	1×10^{-4}
Learning-rate schedule	Linear warmup + cosine decay	Linear warmup + cosine decay
Weight decay	1×10^{-5}	1×10^{-5}
Warmup steps	30	600
Training steps	600	12,000

The head has 20 valid action coordinates padded to 40. Policy episode counts are ordered as baseline / pseudo-label augmentation / fully supervised upper bound.

Table 14: Inference cost and deployment settings for real-world evaluation.

Component	Single-call latency (ms)	Parameters
SAR-IDE + head (offline labeling)	–	–
GR00T N1.6 policy (online control)	55–75	∼3.2B
Deployment GPU	Single NVIDIA A100 40GB	
Action frequency within chunk (Hz)	30	
Robot control frequency (Hz)	30	
Executed action-chunk length	8 steps	
Evaluation trials per condition	10	

The 30 Hz action frequency specifies the temporal spacing of actions within a predicted chunk, not the number of model inference calls per second. The 55–75 ms latency measures one policy call that generates a chunk; the robot executes actions at 30 Hz. Both pickup and transfer success rates are computed over all 10 trials per condition.

K Discussion

K.1 Sharing Motion Representations without Unifying Native Action Spaces

Heterogeneous robot datasets need not share a single explicit action parameterization to contribute to representation learning. In SAR-IDE, the visual encoder and latent-action pathway are shared across embodiments, while embodiment-specific projectors predict trajectories in their native action spaces (Section 3.2). This separates the representation of observed change from its numerical expression in a particular robot's coordinates. The shared pathway can therefore benefit from multiple sources of kinematic supervision without first resolving every difference in coordinate frames, arm ordering, or action dimensionality. It does not remove the need to specify the target embodiment's action convention when decoding executable actions.

Appendix I visualizes selected JHU–Hamlyn pairs with high latent-action similarity. Their related tissue-lifting and retraction behaviors suggest that the shared representation associates similar in-

strument motion across dVRK recording configurations with different screen-side arm assignments. These qualitative examples complement the unseen-embodiment adaptation in Table 1. Adapting the representation to a new robot requires training its output projector in the target action space.

K.2 Complementary Roles of Kinematic and Tool-Centric Supervision

Kinematic and tool-centric objectives constrain different aspects of motion learning. Inverse dynamics anchors the representation to measured trajectories, including translation, orientation, and gripper state. Tool-centric forward prediction connects the latent action to changes in the observed instruments through mask, point, and flow targets. It also supplies a training signal for videos without synchronized robot actions. Thus, the visual objective complements physical grounding rather than serving as a substitute for it.

Table 4 supports this complementarity: removing kinematics strongly degrades action regression, while the full model improves over kinematics-only IDM and whole-scene feature reconstruction with Latent FDM across the reported tasks. Latent FDM also shows more scattered background responses in Figure 9, consistent with weaker instrument selectivity. The attribution results provide a separate diagnostic. In Table 12, SAR-IDE assigns more attribution to instruments than IDM despite a less concentrated overall map. These Grad$\times$Act measurements characterize instrument alignment along the full action pathway on the sampled training subsets.

K.3 When to Use Encoder Embeddings or Latent Actions

The two outputs serve different downstream needs. Direct encoder embeddings $\mathbf{H}$ retain spatially indexed visual tokens before IDM-specific fusion and pooling. They are the natural interface when a head needs local appearance, instrument–tissue context, or spatial correspondence, as in action recognition, multi-label action presence, and dense tracking. The latent action $\mathbf{z} \in \mathbb{R}^{512}$ instead summarizes an endpoint transition through a pretrained motion bottleneck. It is useful when the desired interface is a compact motion code for few-shot action recovery or offline action labeling. The four pooling queries in our architecture are internal to this mapping; they are fused into one vector per frame pair, not retained as four output latent tokens. Compactness reduces the representation presented to the head, not the cost of running the shared encoder.

Our results favor a task-dependent choice. Encoder embeddings outperform latent actions on RARP50, SMARTS [51], and GraSP recognition under the same attentive-classifier protocol (Tables 2 and 7). Both representations improve overall action-regression accuracy on MedbotWorld [39] relative to the pretrained visual baselines (Table 1). A compact motion representation need not preserve every cue required for semantic recognition to be useful for action recovery.

Token budget matters when interpreting these comparisons. LARY [23] evaluates visual encoders and latent-action models with nine-frame inputs and a common attentive-probe design after dimensional projection. However, some compact latent models expose substantially fewer spatial tokens; LARY also studies sequence-length and quantization bottlenecks. Equal feature width does not equalize token count or retained spatial information. Compression is therefore a plausible contributor to weaker recognition, not an established sole explanation for the gap, and LARY's quantization findings do not directly characterize our continuous latent. Moreover, our few-shot Emb./Lat. variants transfer different modules and train different heads (Appendix D.3). Isolating the bottleneck itself would require matching token budgets, head capacity, and transferred components. We interpret the present results as evaluations of complementary transfer interfaces, not a universal ranking of embeddings versus latent actions.