From Articulated Kinematics to Routed Visual Control for Action-Conditioned Surgical Video Generation

Anonymous Author(s)
Affiliation
Address
email

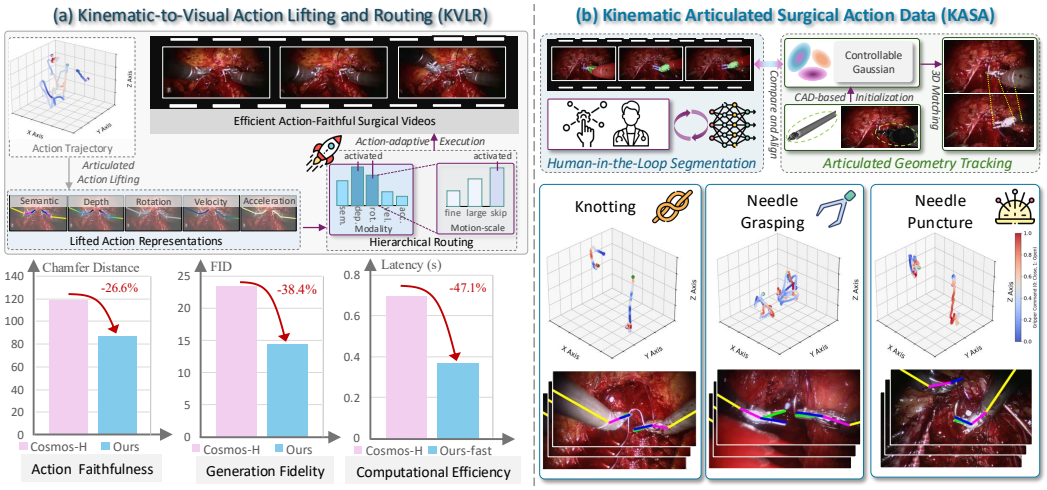

Figure 1: (a) KVLR transforms low-dimensional articulated kinematics into five image-aligned control modalities. A hierarchical routing mechanism selectively activates the most relevant modalities and motion scales, improving action faithfulness, visual fidelity, and computational efficiency. (b) KASA provides articulated action annotations curated from real robotic surgical videos.

Abstract

Action-conditioned surgical video generation is a critical yet highly challenging problem for robotic surgery. The core difficulty is that low-dimensional control vectors must precisely govern complex image-space evolution. In this work, we propose a **kinematic-to-visual** lifting paradigm that converts articulated kinematics into a unified set of five image-aligned control modalities. Building on this representation, we introduce a **hierarchically routed visual control** framework that selectively activates the most relevant control modalities and motion scales. Instead of uniformly applying all control signals, our model performs hierarchical routing to dynamically allocate conditioning capacity. We further design kinematic-prior-guided routing loss functions to ensure physically meaningful, temporally stable, and efficient expert utilization. **To improve efficiency**, we propose a budgeted training and inference scheme that leverages routing-induced sparsity. By selectively discarding low-significance control pathways during training and execution, our approach enables adaptive computation that is complementary to standard distillation. We additionally construct **a new benchmark** with curated articulated annotations, obtained through human-in-the-loop semantic labeling and differentiable pose tracking, providing realistic supervision for action-conditioned surgical video generation. **Extensive experiments** demonstrate that our method consistently improves action faithfulness, visual fidelity, and cross-domain generalization over diverse baselines. Moreover, our efficient variant achieves substantial reductions in latency while maintaining strong control accuracy.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

1 Introduction

Robotic surgery is entering an era where perception and control must be tightly integrated, making action-conditioned surgical video generation a promising enabling tool [8, 25, 6, 3, 48, 34, 45]. Given a desired surgical action sequence, the ability to synthesize realistic and physically consistent visual outcomes has broad applications [3, 39, 17]. For instance, surgeons or learning systems can simulate "what-if" scenarios (see Fig. 6) under different manipulation strategies. However, as illustrated in Fig. 1 (a) bottom, this problem is fundamentally challenging: it requires simultaneously achieving high visual realism and precise action faithfulness, where low-dimensional articulated control signals must govern complex, high-dimensional image-space evolution.

The core difficulty lies in how to effectively introduce control signals into video generation [47, 21, 61]. Text-based conditioning [8, 17, 25], while flexible, is insufficient for surgical settings due to its ambiguity and lack of geometric grounding. Prior efforts have explored dense visual control signals (*e.g.*, depth, segmentation, or flow) [6, 3, 39, 48], which improve controllability but rely on expensive annotations or unavailable inputs in real robotic systems. This motivates the need for a unified representation that is both physically grounded and visually aligned.

We propose KVLR, a **K**inematic-to-**V**isual **L**ifting and **R**outing method (Sec. 2.2) that transforms articulated kinematics into visual control. Specifically, we lift low-dimensional articulated kinematics into a pixel-aligned representation consisting of five interpretable modalities (semantics, depth, rotation, velocity, and acceleration), forming a unified visual control basis (Fig. 1 (a) top). This kinematic-to-visual action field enables the model to explicitly encode where the tool is, how it is oriented, and how it moves. Compared to prior works [6, 39, 17], this representation is lightweight and can be directly derived from articulated actions commonly available in robotic settings.

Building on this representation, KVLR allows a hierarchical routing mechanism (Sec. 2.3) that dynamically allocates control capacity. As illustrated in Fig. 1 (a) top, instead of uniformly applying all control modalities [57, 52, 11, 59, 29], our model performs modality-level routing to select the most relevant control modalities, and further applies motion-scale routing to specialize computation for fine manipulation, large transport motion, or static regions. To ensure that routing reflects physical structure rather than arbitrary patterns, we design kinematic-prior-guided loss functions (Sec. 2.4) that enforce physically meaningful expert utilization and temporal consistency.

An important product of hierarchical routing is the emergence of controllable sparsity, which we leverage for efficient generation. As shown in Fig. 1 (a) top and Sec. 2.5, routing naturally identifies low-significance control pathways that can be skipped without degrading performance. We exploit this property through a budgeted training and inference scheme, combining action-aware distillation with significance-driven execution. This enables the model to selectively discard unnecessary computation, achieving substantial efficiency gains. Empirically, our efficient variant reduces latency by 47.1% while maintaining strong control accuracy (Fig. 1 (a) bottom).

To support this task, we further construct KASA, a new benchmark for **K**inematic **A**rticulated **S**urgical **A**ction. As illustrated in Fig. 1 (b) and Sec. 3, KASA provides curated annotations that bridge real surgical videos and articulated control signals. Our pipeline combines human-in-the-loop segmentation with differentiable pose tracking, enabling the extraction of part-aware geometry and temporally consistent trajectories without requiring external robot logs. The dataset covers diverse surgical actions such as knotting, needle grasping, and puncture, capturing both large-scale motion and fine-grained manipulation, and serves as a realistic testbed for evaluating structured control.

Extensive experiments (Sec. 4) demonstrate that KVLR consistently improves action faithfulness, visual fidelity, and generalization. As summarized in Fig. 1 (a) bottom, our method achieves 26.6% Chamfer Distance decrease and 38.4% FID decrease compared to strong baselines. KVLR consistently outperforms text-conditioned, raw-action-conditioned, and dense-visual-conditioned methods. Furthermore, KVLR shows favorable zero-shot transfer on unseen scenes.

In summary, this work makes the following contributions: (1) A kinematic-to-visual lifting paradigm that transforms articulated kinematics into unified, image-aligned control modalities; (2) A hierarchical routing method that dynamically allocates control across modalities and motion scales; (3) An action-adaptive efficient generation scheme that leverages routing-induced sparsity for significant computational acceleration; (4) KASA, a new benchmark with articulated annotations from real surgical videos for evaluating action-conditioned generation; (5) Comprehensive experiments demonstrating improved action faithfulness, visual fidelity, efficiency, and generalization.

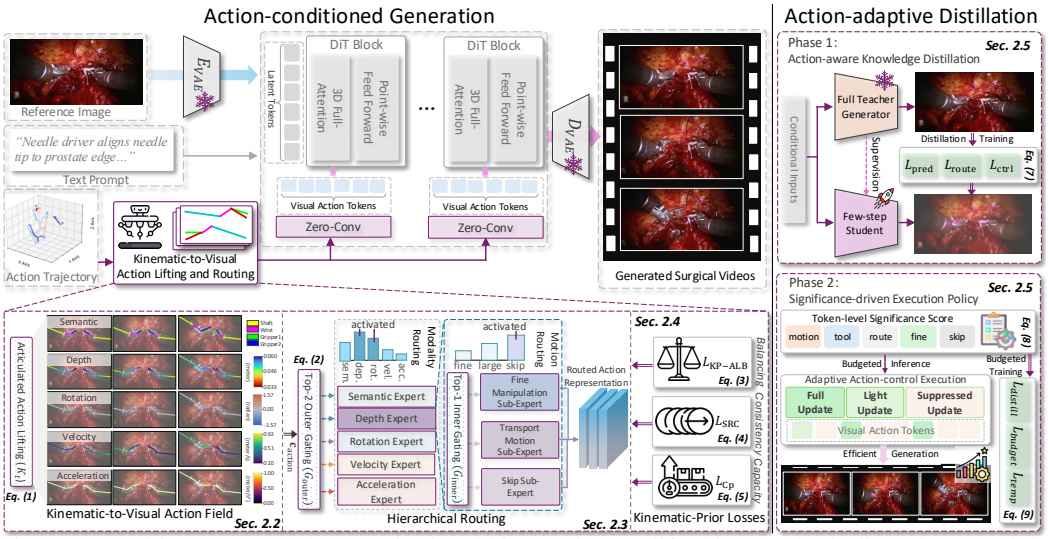

Figure 2: Architecture overview. Articulated kinematics are lifted into a pixel-aligned KVA-Field, then hierarchically routed across physical modalities and motion scales. This paradigm improves action faithfulness under heterogeneous dynamics and supports action-adaptive efficient generation.

2 Methodology

2.1 Framework Overview

As shown in Fig. 2 top left, given a reference image I^{ref}, a text prompt y, and an articulated action sequence $\mathrm{a}_{1:T}$ from the surgical robot, our goal is to generate a video $\hat{\mathrm{x}}_{1:T}$ conditioned on the specified surgical actions: $\hat{\mathrm{x}}_{1:T} = \mathcal{F}(I^{\text{ref}}, y, \mathrm{a}_{1:T})$. As illustrated in Fig. 2 bottom left, we first lift articulated kinematics into a pixel-aligned *K*inematic-to-*V*isual *A*ction *Field* (*KVA-Field*), which encodes multi-modal cues directly in image space (Sec. 2.2). We then apply *hierarchical routing* to allocate conditional computation according to physical modalities and motion scales (Sec. 2.3). The kinematic-prior-guided routing loss functions are further designed to ensure physically meaningful, temporally stable, and efficient expert utilization (Sec. 2.4). This design reveals intrinsic conditional sparsity, which is later leveraged for action-adaptive efficient generation (Sec. 2.5).

2.2 Kinematic-to-Visual Action Field

To bridge the gap between low-dimensional control and dense visual synthesis, we lift articulated instrument actions into a pixel-aligned *Kinematic-to-Visual Action Field (KVA-Field)*. We model the surgical tool with a part-aware articulated structure composed of a shaft, a wrist, and two gripper components (as shown in Fig. 2 bottom left). For each frame t, the articulated action is represented as $a_t = [p_t, r_t, q_t^{\text{sw}}, q_t^{\text{lg}}, q_t^{\text{rg}}] \in \mathbb{R}^9$. Here, $p_t \in \mathbb{R}^3$ and $r_t \in \mathbb{R}^3$ denote the wrist translation and wrist orientation, respectively, with r_t expressed in axis-angle form. The scalar variables q_t^{sw}, q_t^{lg}, and q_t^{rg} denote the shaft-to-wrist, wrist-to-left-gripper, and wrist-to-right-gripper joint states. Together, these variables are turned into a compact articulated action representation $K_t \in \mathbb{R}^{H\times W\times 9}$, with channels corresponding to part semantics, depth, orientation, velocity, and acceleration:

$$K_t(h,w) = [s_t(h,w), d_t(h,w), \rho_t(h,w), v_t(h,w), \alpha_t(h,w)]\,. \tag{1}$$

Here, $s_t \in \mathbb{R}^3$ denotes part-aware semantic channels for shaft, wrist, and gripper; $d_t \in \mathbb{R}$ is rendered depth; $\rho_t \in \mathbb{R}$ is a local orientation descriptor; $v_t \in \mathbb{R}^3$ is projected velocity; and $\alpha_t \in \mathbb{R}$ is acceleration magnitude. v_t and α_t are computed from consecutive articulated states: $v_t = \frac{\Phi(M_t)-\Phi(M_{t-1})}{\Delta t}$, $\alpha_t = \frac{\|v_t - v_{t-1}\|_2}{\Delta t}$. Here, M_t is the articulated tool state and $\Phi(\cdot)$ denotes its camera-projected feature representation on the image plane. **The resulting field encodes not only where the instrument is, but also how it is oriented and how it moves.** By explicitly separating semantic, geometric, and dynamic factors in image space, the KVA-Field provides the structured representation on top of which our hierarchical routing allocates conditional computation.

2.3 Hierarchical Routing

To account for the spatial heterogeneity and heavy-tailed distribution of surgical motion, we introduce the *Hierarchical Routing strategy*. Unlike standard Mixture-of-Experts (MoE) [36, 62] architectures

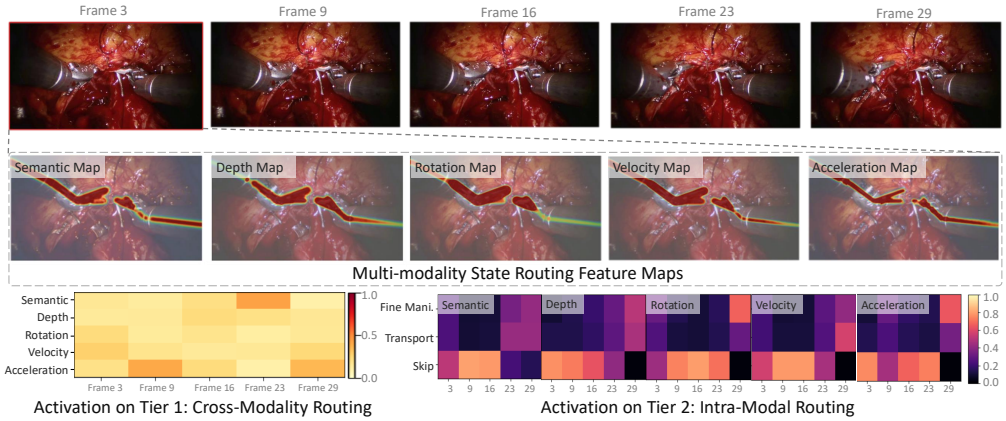

Figure 3: Visualization of hierarchical routing. We render the lifted action cues and show the learned Tier-1 modality and Tier-2 motion-scale activations. The routing patterns vary across motion phases, indicating action-dependent allocation of expert computation.

that route based solely on token content, our design employs a two-tier hierarchy. As shown in Fig. 2 bottom left, this architecture decouples the routing decision into two complementary axes: (1) *Which physical modality is dominant at this spatial location?* (Tier 1), and (2) *What scale of motion dynamics should be extracted within that modality?* (Tier 2). This hierarchical disentanglement allows the model to adaptively allocate computational resources to complex instrument dynamics.

Tier 1: Multi-modality routing. The outer layer consists of 5 modality-specific expert branches $\mathcal{E}_{\text{mod}} = \{E_i\}_{i=1}^{5}$. Each modality expert E_i is structurally constrained to process only its corresponding physical channels from the 9-channel input KVA-Field K_t. As shown in Fig. 2 bottom left, routing at this level is governed by an outer gating network G_{outer}. We encode K_t with a lightweight shared action module $\mathcal{A}_{\text{act}}$ to obtain a holistic gating feature $\mathbf{c}_{\text{action}} = \mathcal{A}_{\text{act}}(K_t)$. G_{outer} is then computed as:

$$G_{\text{outer}}(\mathbf{c}_{\text{action}}, t) = \text{Softmax}\left(\text{Conv}_{1\times1}\left(\text{Concat}[\text{Pool}(\mathbf{c}_{\text{action}}), \tau(t)]\right)\right), \tag{2}$$

where Pool and $\tau(t)$ denote spatial average pooling and the diffusion timestep embedding, respectively. To ensure training stability, we employ a scheduling strategy [36, 62] that transitions from dense softmax fusion to sparse top-k routing (default $k = 2$).

Tier 2: Motion-scale routing. As illustrated in Fig. 2 bottom left, within each modality expert, we introduce a secondary layer to handle diverse motion dynamics. Each modality branch is composed of three sub-experts controlled by a top-1 inner gating mechanism G_{inner}: **(1)** Fine Manipulation Sub-Expert: Utilizes 1×1 and dilated 3×3 convolutions to capture high-frequency details required for precision tasks like knotting or grasping. **(2)** Transport Motion Sub-Expert: Employs large-kernel (7×7) depthwise convolutions and global context pooling to model fast tool sweeps and large displacements. **(3)** Skip Sub-Expert: An identity mapping branch that allows zero-cost computation for static instrument regions, preventing unnecessary feature transformation.

Routing fusion. The inner router uses a 1×1 convolution to map the modality-specific feature vector at each spatial location to motion-scale logits. This produces an independent fine, large, or skip decision for each token, allowing the selected operator to adapt to the local kinematic state. The output of each modality branch is the selected sub-expert feature weighted by its inner-routing confidence. The final routed action representation is obtained by fusing the selected expert outputs.

Hierarchical injection. We then inject the fused control features into the video generator through zero-initialized additive layers on top of the DiT backbone, as shown in Fig. 2 top left. This preserves the pretrained generative prior at initialization while allowing the model to progressively learn action conditioning. In this way, the routing strategy allocates conditional computation according to both physical modality and motion complexity. Fig. 3 shows that the learned routing is motion-dependent: low-motion regions favor semantic and skip pathways, while larger motions increasingly activate dynamic modalities and transport-oriented computation, matching the design of hierarchical routing.

2.4 Kinematic-Prior Learning Loss Functions

The routing module in Sec. 2.3 should reflect the physical structure of surgical motion rather than purely statistical expert usage. However, as illustrated in Tab. 7, previous MoE [43, 10, 60] objectives, such as uniform load balancing, are poorly matched to the heavy-tailed and heterogeneous dynamics of articulated manipulation. We therefore introduce *kinematic-prior learning* loss functions to make hierarchical routing physically meaningful, temporally stable, and compatible with efficient inference.

Kinematic-prior adaptive load balancing. Instead of enforcing uniform expert utilization, we guide the modality router with a dynamic physical signal derived from the KVA-Field. The **motivation** is that different frames are dominated by different physical factors: (1) static tool regions may be dominated by semantic and depth cues, (2) wrist articulation by rotation cues, and (3) rapid tool motion by velocity or acceleration cues. Thus, the target expert usage should depend on the physical signal present in the current action field rather than being fixed across all samples. Let $K_t^{\text{sem}} \in \mathbb{R}^{H\times W\times 3}$, $K_t^{\text{dep}} \in \mathbb{R}^{H\times W}$, $K_t^{\text{rot}} \in \mathbb{R}^{H\times W}$, $K_t^{\text{vel}} \in \mathbb{R}^{H\times W\times 3}$, and $K_t^{\text{acc}} \in \mathbb{R}^{H\times W}$ denote the feature fields of depth, semantic, rotation, velocity and acceleration channels of the KVA-Field K_t at frame t. For each spatial token $u \in \Omega$, where Ω is the image-token domain, we compute modality-wise physical magnitudes as $e_t(u) = \left[\|K_t^{\text{sem}}(u)\|_2,\ |K_t^{\text{dep}}(u)|,\ |K_t^{\text{rot}}(u)|,\ \|K_t^{\text{vel}}(u)\|_2,\ |K_t^{\text{acc}}(u)|\right]^\top$. These terms respectively measure part-presence strength, camera-depth magnitude, articulated rotation magnitude, projected motion magnitude, and acceleration magnitude. We aggregate them over spatial tokens to obtain the frame-level physical energy vector $\mathcal{E}_i(K_t) = \frac{1}{|\Omega|}\sum_{u\in\Omega} e_{t,i}(u),\ i \in \mathcal{M}$, where $\mathcal{M} = \{\text{sem}, \text{dep}, \text{rot}, \text{vel}, \text{acc}\}$ is the set of modality experts.

The normalized physical signal is then $\pi_i(K_t) = \frac{\mathcal{E}_i(K_t)}{\sum_{j\in\mathcal{M}}\mathcal{E}_j(K_t)}$. This signal defines the desired routing mass for each modality expert according to the physical content of the current action field. Let P_i be the mean soft routing probability assigned to expert i, and let f_i be the fraction of tokens whose top-ranked routing assignment selects expert i. Following sparse MOE load estimation, we define the realized routing load as $\ell_i = f_i P_i$ and minimize:

$$\mathcal{L}_{\text{KP-ALB}} = \sum_{i\in\mathcal{M}} \left(\ell_i - \pi_i(K_t)\right)^2. \tag{3}$$

This objective encourages the router to allocate capacity according to physically meaningful action cues, rather than forcing all modality experts to be used uniformly.

Spatiotemporal routing consistency. Existing MoE diffusion models process tokens within a batch-level global pool, ignoring the temporal correlation inherent in sequential video frames [10, 60]. We introduce a Spatiotemporal Routing Consistency (SRC) loss to enforce temporal coherence in the continuous routing distributions, localized to moving instruments:

$$\mathcal{L}_{\text{SRC}} = \frac{1}{T-1}\sum_{t=1}^{T-1}\sum_{h,w} M_{\text{tool}}^{(t)}(h,w)\ \|\text{R}_t(h,w) - \text{R}_{t-1}(h,w)\|_2^2, \tag{4}$$

where $\text{R} \in \mathbb{R}^{H\times W\times N}$ denotes the continuous routing probabilities at frame t, and $M_{\text{tool}}^{(t)} \in \{0,1\}^{H\times W}$ is the binary spatial mask indicating surgical tool presence (derived from K_t^{sem}). By masking the loss to instrument regions, we avoid penalizing routing changes in static background areas while stabilizing expert selection for dynamic tools.

Capacity predictor loss. To enable efficient inference without computing full top-k routing at runtime, we train a lightweight capacity predictor [36, 62] to anticipate routing decisions. The predictor receives detached action features and outputs raw logits $\hat{R}$ for each expert. We optimize via Binary Cross-Entropy against the actual routing mask R^* generated by the gating networks:

$$\mathcal{L}_{\text{CP}} = \text{BCE}\left(\sigma(\hat{R}), \text{sg}(R^*)\right), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function. During inference, the predictor's outputs are thresholded using EMA-updated per-expert quantiles, enabling dynamic routing with $\mathcal{O}(1)$ complexity per spatial token. The thresholds are updated every training step to track the evolving routing distribution, preventing expert starvation from positive-feedback loops. More implementation details are provided in Sec. D.5.

Overall objective. The final objective combines the generation loss with the above routing priors:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Flow}} + \lambda_1\mathcal{L}_{\text{KP}-\text{ALB}} + \lambda_2\mathcal{L}_{\text{SRC}} + \lambda_3\mathcal{L}_{\text{CP}}, \tag{6}$$

where $\mathcal{L}_{\text{Flow}}$ is the standard flow-matching loss [30], and $\lambda_{1:3}$ are weighting coefficients tuned as $\lambda_1 = 0.01$, $\lambda_2 = 0.005$, $\lambda_3 = 0.01$. More details on each loss function are presented in Sec. D.5.

2.5 Action-adaptive Efficient Generation

The structured control pathway improves action faithfulness and reveals where dense conditional computation is necessary. We exploit this property to build an *action-adaptive efficient generation* scheme.

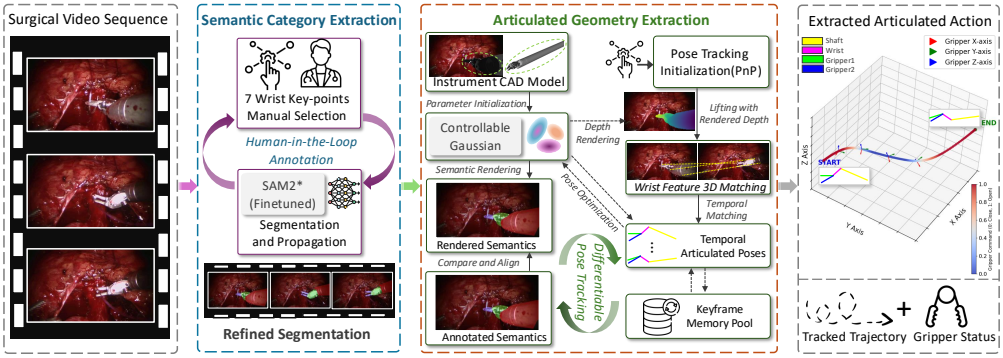

Figure 4: Data construction pipeline. Given robotic surgical videos, we obtain articulated action supervision through human-in-the-loop semantic annotation and differentiable pose tracking, producing lightweight yet visually grounded control signals for action-conditioned video generation.

Phase 1: Action-aware distillation. As shown in Fig. 2 top right, we first distill a few-step student generator from the full teacher under the same conditioning inputs. To preserve action-aware behavior, the student is supervised not only on the final prediction, but also on the internal control structure:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{pred}} + \lambda_r \mathcal{L}_{\text{route}} + \lambda_c \mathcal{L}_{\text{ctrl}}, \tag{7}$$

where $\mathcal{L}_{\text{pred}}$ matches the teacher prediction, $\mathcal{L}_{\text{route}}$ aligns routing behavior, and $\mathcal{L}_{\text{ctrl}}$ preserves structured action-control features. More details on each loss function are provided in Sec. D.6. This transfers the teacher's control policy into a few-step generator instead of distilling appearance alone.

Phase 2: Significance-driven execution. We then derive a token-level action significance score from the lifted action field and routing signals:

$$s(u) = w_m e_{\text{motion}}(u) + w_t m_{\text{tool}}(u) + w_r c_{\text{route}}(u) + w_f q_{\text{fine}}(u) - w_s p_{\text{skip}}(u), \tag{8}$$

where u denotes a latent token; e_{motion}, m_{tool}, c_{route}, q_{fine}, and p_{skip} denote normalized motion intensity, tool presence, routing confidence, fine-motion preference, and skip probability. The non-negative weights w_m, w_t, w_r, w_f, w_s control the relative contribution of each term. At inference, $s(u)$ partitions latent tokens into three execution modes. As shown in Fig. 2 bottom-right, high-significance tokens perform a full control update by evaluating the KVA encoder, routers, and selected experts. Medium-significance tokens perform a light update by reusing cached routing decisions and only recomputing the final residual adapter. Low-significance tokens suppress the current control update and reuse the previous cached control residual. A frame-level refresh interval (detailed in Sec. D.6) prevents cache drift by forcing full updates at fixed temporal intervals.

Budget-aware objective. We optimize the efficient path with:

$$\mathcal{L}_{\text{eff}} = \mathcal{L}_{\text{distill}} + \lambda_b \mathcal{L}_{\text{budget}} + \lambda_t \mathcal{L}_{\text{temp}}, \tag{9}$$

where $\mathcal{L}_{\text{budget}}$ enforces a target compute budget and $\mathcal{L}_{\text{temp}}$ stabilizes reused or lightly updated regions over time. This objective reduces generation cost while preserving action-conditioned control. More details are provided in the Sec. D.6 of the appendix.

3 KASA Benchmark Construction

Benchmark overview. A key obstacle in action-conditioned surgical video generation is the lack of benchmarks pairing real surgical videos with explicit articulated controls. Existing works [8, 17, 25, 6, 3, 39] typically provide coarse language descriptions or dense visual annotations, but not lightweight action supervision aligned with visual generation. We construct the Kinematic Action-centric Surgical (*KASA*) benchmark for articulated tool-action-conditioned generation, associating each video with video-reconstructed annotations. These compact, visually grounded controls are compatible with robotic settings and aligned with structured control signals for kinematic-to-visual action modeling.

Task-driven sequence selection. We curate KASA from the SAR-RARP dataset [38], collecting 1,047 surgical scenes with 105,175 frames. The benchmark focuses on three representative sub-actions of *needle grasping*, *needle puncture*, and *knotting*, which together cover a broad range of surgical dynamics, from large transport motion to fine local manipulation. Videos are downsampled to 30 FPS, and each sequence contains 55-235 frames. KASA exhibits articulated motion, frequent self-occlusion, and complex instrument-tissue interactions, providing a challenging test platform for evaluating both action faithfulness and structured control under heterogeneous surgical dynamics.

Human-in-the-loop semantic annotation. To obtain reliable part-aware supervision, we adopt a Human-in-the-loop Semantic Annotation (HSA) procedure (Fig. 4 left). We first estimate part-level

Table 1: Action faithfulness on KASA. All methods are finetuned under the same split.

Method	Knotting			NeedleGrasping			NeedlePuncture		
	CD ↓	TI ↑	AF ↓	CD ↓	TI ↑	AF ↓	CD ↓	TI ↑	AF ↓
Text-conditioned Generation									
Wan-2.2	169.35±3.12	0.56±0.02	6.54±0.45	135.80±2.85	0.57±0.02	26.95±1.20	98.15±2.15	0.65±0.02	5.31±0.38
HunyuanVideo	162.81±2.95	0.57±0.01	6.07±0.50	131.63±2.60	0.59±0.02	26.05±1.15	94.11±1.95	0.67±0.01	4.82±0.35
OpenSora-2.0	173.42±3.40	0.53±0.02	7.02±0.62	140.56±3.10	0.55±0.02	28.35±1.40	102.67±2.45	0.63±0.02	5.72±0.45
Cosmos-H-Surgical	150.26±2.80	0.59±0.02	4.88±0.35	119.38±2.45	0.61±0.02	23.64±1.05	85.30±1.80	0.71±0.01	3.44±0.25
Raw-action-conditioned Generation									
Wan-2.2*	135.48±2.55	0.68±0.02	5.23±0.40	108.64±2.15	0.70±0.02	21.56±0.95	78.52±1.65	0.78±0.02	4.25±0.30
HunyuanVideo*	130.25±2.40	0.70±0.01	4.86±0.35	105.31±2.05	0.72±0.01	20.84±0.85	75.29±1.50	0.80±0.01	3.86±0.25
OpenSora-2.0*	138.74±2.65	0.65±0.02	5.62±0.45	112.45±2.25	0.68±0.02	22.68±1.10	82.14±1.75	0.76±0.02	4.58±0.35
Cosmos-H-Surgical*	120.21±2.34	0.74±0.01	3.91±0.22	95.51±1.86	0.76±0.02	18.91±0.74	68.24±1.41	0.84±0.01	2.75±0.18
Visual-action-conditioned Generation									
SurgSora†	119.50±2.10	0.75±0.02	3.85±0.20	94.80±1.65	0.77±0.01	18.50±0.65	67.50±1.50	0.85±0.02	2.65±0.15
OpenSora-2.0†	113.80±1.95	0.84±0.02	1.51±0.15	91.10±1.80	0.87±0.02	12.20±0.55	61.40±1.35	0.87±0.01	1.88±0.12
KVLR-fast(ours)	114.32±1.91	0.85±0.01	1.54±0.14	91.85±1.48	0.88±0.01	12.45±0.51	63.45±1.12	0.86±0.01	1.95±0.12
KVLR(ours)	**110.57**±1.62	**0.88**±0.01	**1.18**±0.10	**89.22**±1.35	**0.91**±0.01	**10.67**±0.43	**60.90**±0.96	**0.89**±0.01	**1.59**±0.09

Table 2: Generation Fidelity on KASA. PSNR/SSIM are averaged over generated frames, FID is computed on generated frames as images, and FVD on 17-frame clips.

Method	Knotting				NeedleGrasping				NeedlePuncture			
	PSNR ↑	SSIM ↑	FID ↓	FVD ↓	PSNR ↑	SSIM ↑	FID ↓	FVD ↓	PSNR ↑	SSIM ↑	FID ↓	FVD ↓
Text-conditioned Generation												
Wan-2.2	16.20±0.25	0.55±0.02	45.30±2.40	420.50±18.50	15.30±0.22	0.52±0.02	48.20±2.60	450.30±20.10	16.80±0.26	0.58±0.02	40.10±2.10	310.40±15.50
HunyuanVideo	16.50±0.22	0.57±0.01	42.10±2.15	395.20±16.20	15.60±0.20	0.54±0.01	44.50±2.35	415.80±18.40	17.10±0.24	0.60±0.01	37.20±1.95	285.20±14.20
OpenSora-2.0	16.10±0.28	0.54±0.02	47.50±2.55	445.80±19.80	15.10±0.25	0.51±0.02	50.40±2.80	470.20±22.50	16.60±0.25	0.56±0.02	42.30±2.25	335.60±16.80
Cosmos-H-Surgical	17.20±0.20	0.62±0.01	36.50±1.90	325.40±14.50	16.40±0.18	0.59±0.01	38.50±2.10	355.50±15.80	17.80±0.22	0.65±0.01	31.50±1.75	245.70±12.40
Raw-action-conditioned Generation												
Wan-2.2*	17.80±0.24	0.66±0.02	31.60±1.85	280.20±13.50	16.90±0.21	0.63±0.02	33.10±2.05	310.80±14.80	18.20±0.25	0.68±0.02	26.30±1.65	200.50±11.50
HunyuanVideo*	18.10±0.22	0.68±0.01	28.40±1.75	255.40±12.80	17.30±0.19	0.65±0.01	30.20±1.95	285.90±13.60	18.60±0.23	0.71±0.01	23.40±1.55	175.40±10.80
OpenSora-2.0*	17.50±0.26	0.64±0.02	34.20±2.10	305.60±15.20	16.60±0.24	0.61±0.02	36.40±2.25	335.70±16.50	17.90±0.27	0.66±0.02	29.10±1.85	225.80±12.20
Cosmos-H-Surgical*	18.90±0.23	0.72±0.01	24.46±1.81	225.72±12.35	18.20±0.18	0.70±0.01	25.87±1.94	251.26±15.18	19.14±0.20	0.75±0.01	19.72±1.67	142.97±10.74
Visual-action-conditioned Generation												
SurgSora†	18.85±0.21	0.73±0.02	24.15±1.50	222.40±11.50	18.35±0.15	0.71±0.01	25.40±1.65	248.50±13.20	19.25±0.18	0.76±0.01	19.35±1.45	140.60±9.80
OpenSora-2.0†	20.75±0.25	0.79±0.01	16.80±1.60	168.90±10.80	19.90±0.20	0.75±0.02	18.95±1.75	185.20±14.50	20.75±0.15	0.83±0.01	12.90±1.30	97.50±8.90
KVLR-fast(ours)	20.45±0.17	0.78±0.01	18.25±1.12	185.32±10.46	19.68±0.15	0.76±0.01	20.15±1.18	205.40±12.33	20.55±0.16	0.81±0.01	14.60±0.82	112.50±7.64
KVLR(ours)	**21.09**±0.14	**0.82**±0.01	**14.78**±0.89	**154.56**±8.72	**20.17**±0.13	**0.80**±0.01	**16.60**±0.91	**168.94**±9.85	**21.22**±0.12	**0.85**±0.01	**11.75**±0.68	**87.10**±5.96

masks for the instrument shaft, wrist, and grippers using a SAM2 model [40] finetuned on SAR-RARP, and then refine the propagated predictions through interactive human correction. This step removes part confusion, boundary leakage, and temporal inconsistency, yielding clean semantic labels that are crucial for disentangling different physical factors in our structured control representation.

Differentiable pose tracking. To recover continuous articulated geometry, we further introduce Differentiable Pose Tracking (DPT), which estimates temporally consistent pose trajectories directly from videos without relying on external kinematic logs or hardware tracking (Fig. 4 right). DPT uses a render-and-compare formulation initialized from instrument CAD priors and aligns rendered part-aware geometry with the observed video evidence over time. The resulting articulated trajectories provide the geometric and motion supervision, enabling downstream construction of semantic, depth, rotation, and motion-aligned control signals for action-conditioned video generation. To further validate KASA, Sec. 4.2 and Sec. E report independent-evaluator checks and annotation quality validations.

4 Experiment

4.1 Experimental Setup

Training. We instantiate our models based on OpenSora-2.0 [61]. The models are trained on NVIDIA H20 GPUs with a batch size of 24 using AdamW [33]. The learning rate is set to 1×10^{-5} with weight decay of 0.01. We split KASA into 85% training and 15% validation sets. The videos are generated at a default resolution of 576×1024 with clip lengths aligned with the ground-truth sequences.

Baselines. Our full model uses 50 denoising steps, while *KVLR-fast* denotes the distilled efficient variant with a 4-step student. We compare with Wan-2.2 [47], HunyuanVideo [21], OpenSora-2.0 [61], Cosmos-H-Surgical [17], and SurgSora [6], all finetuned on KASA under the same split and input protocol. Models marked with '*' use ControlNet-style injection (same parameter scale as ours) with raw-action (articulated trajectory vectors) conditions, while models with '†' leverage dense visual conditions as SurgSora [6] (*i.e.*, depth, semantic, and flow maps). The results are reported with the mean and standard deviation over three runs. Additional details are provided Sec. D of the appendix.

4.2 Main Results

Action faithfulness. We evaluate Chamfer Distance (CD), Temporal IoU (TI), and Area Flicker (AF) using an independently trained Grounded SAM [41], comparing predicted tool masks with

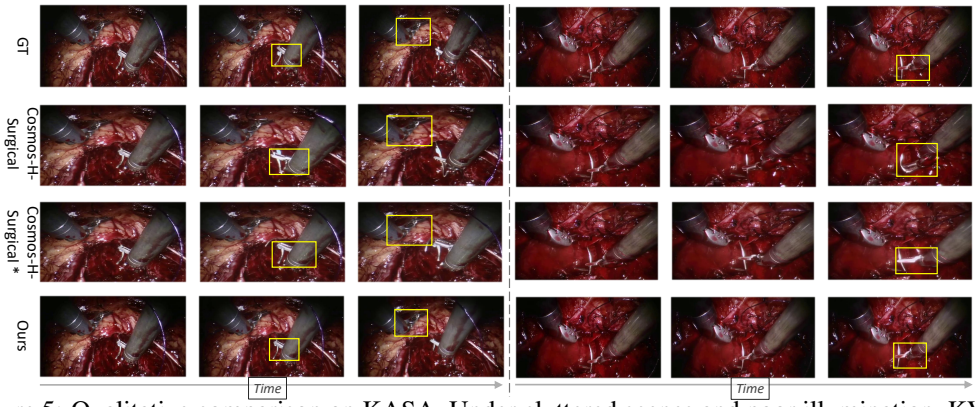

Figure 5: Qualitative comparison on KASA. Under cluttered scenes and poor illumination, KVLR better preserves tool states and action-consistent evolution than Cosmos-H-Surgical*.

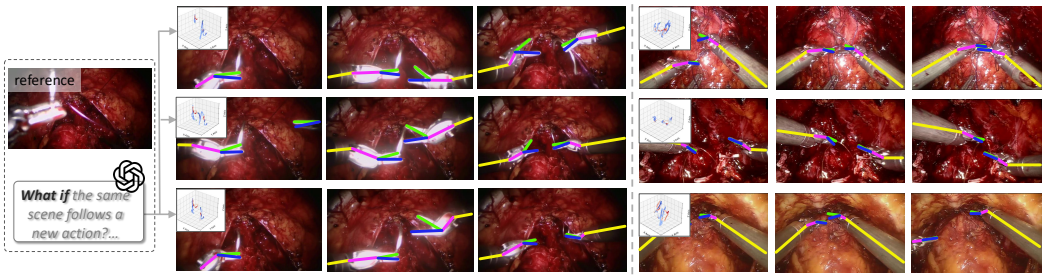

Figure 6: Generalization results. (a) Controllable synthesis under user-specified actions with the same reference. (b) Qualitative transfer to unseen scenes with different appearances and motion patterns.

ground-truth action masks; details are in Sec. D.1. As shown in Tab. 1, text-only baselines struggle to follow surgical actions, and raw-action injection improves control but remains limited. Dense-visual-condition baselines further improve action precision using semantic, depth, and flow maps, but such inputs are often unavailable in robotic settings and still underperform KVLR. *KVLR-fast* retains compelling control fidelity, showing that action alignment is largely preserved under reduced computation.

Generation fidelity. Tab. 2 shows consistent gains in visual fidelity. Direct action injection improves over text-only generation, while structured visual-action-condition baselines further benefit from dense scene-level guidance. KVLR achieves the best overall performance, particularly on *Knotting* and *NeedleGrasping*, where realistic tool-tissue evolution requires accurate local motion control. This suggests that kinematic-to-visual lifting and routing provide benefits beyond simply introducing dense visual conditions. Fig. 5 further qualitatively confirms that KVLR better preserves action-consistent evolution and interaction outcomes under cluttered scenes and poor illumination.

Computational efficiency. Tab. 3 reports latency and FLOPs on H20 GPUs with 17-frame clips. Full KVLR prioritizes action faithfulness and fidelity, whereas KVLR-fast targets efficient inference. At both resolutions, KVLR-fast achieves the lowest latency and FLOPs while retaining strong action alignment, showing that routing-derived significance exposes useful sparsity for adaptive acceleration.

Table 3: Computational efficiency evaluation.

Method	288×512		576×1024	
	Latency (s)↓	FLOPs↓ (PF/clip)	Latency (s)↓	FLOPs↓ (PF/clip)
Wan-2.2*	2.38	3.42	11.85	12.18
HunyuanVideo*	6.12	9.15	30.54	32.45
Cosmos-H-Surgical*	0.70	1.91	4.32	7.55
SurgSora†	3.85	5.42	18.25	19.65
OpenSora-2.0†	5.12	7.55	25.60	26.85
KVLR (Ours)	4.64	6.94	23.39	24.77
KVLR-fast (Ours)	**0.37**	**0.55**	**1.88**	**1.98**

Cross-dataset generalization. We evaluate zero-shot transfer on GraSP [46, 2], which differs in tissue appearance, illumination, and viewpoint. Kinematics are extracted by our DPT pipeline without human correction, introducing realistic tracking noise. Without fine-tuning, KVLR maintains favorable faithfulness and fidelity over the evaluated baselines (Tab. 4), suggesting improved robustness to domain shift. This trend is consistent with the qualitative transfer results in Fig. 6 (b) and the "what-if" action control examples in Fig. 6 (a).

Table 4: Cross-dataset generalization on GraSP.

Method	CD↓	TI↑	AF↓	PSNR↑	FID↓
Cosmos-H-Surgical*	134.8±3.9	0.68±0.03	8.42±0.58	16.9±0.3	28.3±1.8
SurgSora†	127.5±3.4	0.71±0.02	7.15±0.45	17.6±0.2	24.0±1.5
OpenSora-2.0†	115.3±3.1	0.77±0.01	4.78±0.38	18.9±0.2	17.0±1.3
KVLR-fast (Ours)	117.6±2.8	0.76±0.02	5.05±0.31	18.7±0.2	19.1±1.1
KVLR (Ours)	**111.9**±2.5	**0.79**±0.01	**4.28**±0.26	**19.2**±0.1	**16.7**±0.9

Table 6: Architecture ablation under parameter-matched control branches.

Method	Total Params	CD ↓	TI ↑	FID ↓	PSNR ↑	Latency (s) ↓
Text-only baseline	11.00B	138.88	0.57	93.31	12.80	**4.59**
+ Raw action vector + direct condition	12.06B	111.11	0.69	79.56	15.08	4.61
+ KVA-Field + direct condition	12.06B	96.45	0.78	45.20	17.65	4.68
+ KVA-Field + Tier-1 routing	12.05B	91.30	0.84	26.85	19.34	4.66
+ KVA-Field + Tier-1 + Tier-2 routing	12.06B	**86.89**	**0.89**	**14.37**	**20.82**	4.64

Trajectory-level verification. Rather than only computing the mask-overlap metrics, we provide an additional trajectory-level evaluation that extracts the visible tool centerline and distal endpoint from each generated video using an independently trained Grounded SAM-based tracker [41], and compares them with the projected trajectory induced by the reference articulated action. As shown in Tab. 5, KVLR reduces both centerline and endpoint errors and improves motion-direction accuracy over Cosmos-H-Surgical*, indicating that its gains are not specific to the default mask-overlap protocol. Action-perturbation checks in appendix F.2 further confirm that the generated motion follows the specified action trajectory.

Table 5: Trajectory-level verification. Ctr. and End. are centerline and distal-endpoint errors in pixels; Dir. is motion-direction accuracy(%).

Method	Ctr.↓	End.↓	Dir.↑
Cosmos-H-Surgical*	11.83	15.62	71.4
KVLR-fast (Ours)	8.05	10.24	82.1
KVLR (Ours)	**7.21**	**9.13**	**84.8**

4.3 Ablation Study

Architecture design. Tab. 6 ablates KVLR under parameter-matched control branches. Despite the matched capacity, direct conditioning with raw action yields limited gains, suggesting that the bottleneck is the fundamental representational mismatch between low-dimensional articulated kinematics and dense visual evolution. Replacing raw actions with the KVA-Field bridges this gap, driving a substantial performance leap (*e.g.*, CD drops to 96.45). Adding hierarchical routing dynamically allocates this fixed parameter budget across varying physical modalities and motion scales, further boosting action faithfulness without inflating latency, showing that action lifting and structured routing are complementary. Qualitative comparisons on different settings are provided in Fig. 8 of the appendix.

Kinematic-prior losses. Tab. 7 shows that the three priors are complementary. Replacing KP-ALB with uniform load balancing [43, 10, 60] mainly hurts spatial alignment and fidelity, while removing SRC causes the largest drop in temporal stability; removing CP also degrades all metrics, indicating that predictable routing is important for stable control. Overall, the best results require kinematic-prior-guided regularization throughout the routing process. More qualitative comparisons on the effect of different losses are provided in Fig. 9 of the appendix.

Table 7: Kinematic-prior ablation.

Method	CD ↓	TI ↑	AF ↓	FID ↓
Full w/o KP-ALB	94.30	0.85	5.20	21.65
Full w/o SRC	90.12	0.81	8.75	17.50
Full w/o CP	89.45	0.87	5.12	16.20
Full model	**86.89**	**0.89**	**4.48**	**14.37**

Efficient generation. Tab. 8 separates generic few-step acceleration from action-aware efficiency. Under the same 4-step budget, DMD2 [56] reduces latency to 0.52 s but degrades control, increasing CD from 86.89 to 98.45. By contrast, our action-aware student achieves the same latency and FLOPs while preserving much stronger control fidelity, with CD 87.50. Spatial adaptive execution further reduces latency/FLOPs to 0.42 s/0.63 PF by skipping low-significance control updates, and temporal caching reaches 0.37 s/0.55 PF with moderate degradation. These results indicate that few-step distillation provides the main acceleration, while action-aware distillation and routing-derived execution better preserve control under reduced computation. A more detailed breakdown analysis is provided in Sec. F.4.

Table 8: Efficient generation ablation.

Method	CD ↓	FID ↓	Latency ↓	FLOPs ↓
Teacher (full generator, 50 steps)	**86.89**	**14.37**	4.64	6.94
Generic distillation (DMD2)	98.45	19.82	0.52	0.78
Action-aware student (few-step only)	87.50	15.60	0.52	0.78
+ spatial adaptive execution	88.65	16.55	0.42	0.63
+ temporal cache (Final efficient)	89.87	17.67	**0.37**	**0.55**

5 Conclusion

This work presents KVLR, a structured framework for action-conditioned surgical video generation from articulated inputs. KVLR lifts kinematics into a pixel-aligned KVA-Field and routes control across modalities and motion scales, improving measured action alignment and generation quality under complex surgical dynamics. We also introduce KASA, a benchmark with articulated annotations from real robotic surgical videos. Experiments show that KVLR outperforms evaluated baselines, while KVLR-fast reduces inference cost with largely preserved action control. Nevertheless, generated videos should be viewed as research aids for simulation and data augmentation rather than clinical evidence. Additional discussions on limitations and broader impact are in Sec. C.

References

[1] Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

[2] Nicolás Ayobi, Alejandra Pérez-Rondón, Santiago Rodríguez, and Pablo Arbeláes. Matis: Masked-attention transformers for surgical instrument segmentation. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2023. doi: 10.1109/ISBI53787.2023.10230819.

[3] Diego Biagini, Nassir Navab, and Azade Farshad. Hierasurg: Hierarchy-aware diffusion model for surgical video generation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 310–319. Springer, 2025.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[5] Rui Chen, Zehuan Wu, Yichen Liu, Yuxin Guo, Jingcheng Ni, Haifeng Xia, and Siyu Xia. Unimlvg: Unified framework for multi-view long video generation with comprehensive control capabilities for autonomous driving, 2025. URL `https://arxiv.org/abs/2412.04842`.

[6] Tong Chen, Shuya Yang, Junyi Wang, Long Bai, Hongliang Ren, and Luping Zhou. Surgsora: Object-aware diffusion model for controllable surgical video generation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 521–531. Springer, 2025.

[7] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[8] Joseph Cho, Samuel Schmidgall, Cyril Zakka, Mrudang Mathur, Dhamanpreet Kaur, Rohan Shad, and William Hiesinger. Surgen: Text-guided diffusion model for surgical video generation. *arXiv preprint arXiv:2408.14028*, 2024.

[9] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. `https://github.com/facebookresearch/slowfast`, 2020.

[10] Zhengcong Fei, Mingyuan Fan, Changqian Yu, Debang Li, and Junshi Huang. Scaling diffusion transformers to 16 billion parameters. *arXiv preprint arXiv:2407.11633*, 2024.

[11] Mingju Gao, Kaisen Yang, Huan ang Gao, Bohan Li, Ao Ding, Wenyi Li, Yangcheng Yu, Jinkun Liu, Shaocong Xu, Yike Niu, Haohan Chi, Hao Chen, Hao Tang, Yu Zhang, Li Yi, and Hao Zhao. Pam: A pose-appearance-motion engine for sim-to-real hoi video generation, 2026. URL `https://arxiv.org/abs/2603.22193`.

[12] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024.

[13] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2025.

[14] Junhao Ge, Zuhong Liu, Longteng Fan, Yifan Jiang, Jiaqi Su, Yiming Li, Zhejun Zhang, and Siheng Chen. Unraveling the effects of synthetic data on end-to-end autonomous driving. *arXiv preprint arXiv:2503.18108*, 2025.

[15] Jiazhe Guo, Yikang Ding, Xiwu Chen, Shuo Chen, Bohan Li, Yingshuang Zou, Xiaoyang Lyu, Feiyang Tan, Xiaojuan Qi, Zhiheng Li, et al. Dist-4d: Disentangled spatiotemporal diffusion with metric depth for 4d driving scene generation. *arXiv preprint arXiv:2503.15208*, 2025.

[16] Yufan He, Pengfei Guo, Mengya Xu, Zhaoshuo Li, Andriy Myronenko, Dillan Imans, Bingjie Liu, Dongren Yang, Mingxue Gu, Yongnan Ji, et al. Surgworld: Learning surgical robot policies from videos via world modeling. *arXiv preprint arXiv:2512.23162*, 2025.

[17] 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. Cosmos-h-surgical: Learning surgical robot policies from videos via world modeling, 2026. URL `https://arxiv.org/abs/2512.23162`.

[18] Tianshuai Hu, Xiaolu Liu, Song Wang, Yiyao Zhu, Ao Liang, Lingdong Kong, Guoyang Zhao, Zeying Gong, Jun Cen, Zhiyu Huang, et al. Vision-language-action models for autonomous driving: Past, present, and future. *arXiv preprint arXiv:2512.16760*, 2025.

[19] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024.

[20] Lingdong Kong, Wesley Yang, Jianbiao Mei, Youquan Liu, Ao Liang, Dekai Zhu, Dongyue Lu, Wei Yin, Xiaotao Hu, Mingkai Jia, et al. 3d and 4d world modeling: A survey. *arXiv preprint arXiv:2509.07996*, 2025.

[21] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[22] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11971–11981, 2025.

[23] Chenxin Li, Hengyu Liu, Yifan Liu, Brandon Y Feng, Wuyang Li, Xinyu Liu, Zhen Chen, Jing Shao, and Yixuan Yuan. Endora: Video generation models as endoscopy simulators. In *International conference on medical image computing and computer-assisted intervention*, pages 230–240. Springer, 2024.

[24] Samuel Li, Pujith Kachana, Prajwal Chidananda, Saurabh Nair, Yasutaka Furukawa, and Matthew Brown. Rig3r: Rig-aware conditioning for learned 3d reconstruction. *arXiv preprint arXiv:2506.02265*, 2025.

[25] Wei Li, Ming Hu, Guoan Wang, Lihao Liu, Kaijing Zhou, Junzhi Ning, Xin Guo, Zongyuan Ge, Lixu Gu, and Junjun He. Ophora: a large-scale data-driven text-guided ophthalmic surgical video generation model. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 425–435. Springer, 2025.

[26] Ao Liang, Lingdong Kong, Tianyi Yan, Hongsi Liu, Wesley Yang, Ziqi Huang, Wei Yin, Jialong Zuo, Yixuan Hu, Dekai Zhu, et al. Worldlens: Full-spectrum evaluations of driving world models in real world. *arXiv preprint arXiv:2512.10958*, 2025.

[27] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, et al. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. *arXiv preprint arXiv:2501.18590*, 2025.

[28] Chih-Hao Lin, Zian Wang, Ruofan Liang, Yuxuan Zhang, Sanja Fidler, Shenlong Wang, and Zan Gojcic. Controllable weather synthesis and removal with video diffusion models. *arXiv preprint arXiv:2505.00704*, 2025.

[29] Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model, 2024.

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

[31] Liu Liu, Xiaofeng Wang, Guosheng Zhao, Keyu Li, Wenkang Qin, Jiaxiong Qiu, Zheng Zhu, Guan Huang, and Zhizhong Su. Robotransfer: Geometry-consistent video diffusion for robotic visual policy transfer. *arXiv preprint arXiv:2505.23171*, 2025.

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[34] Ning Ma, Shu Yang, Yizhao Zhou, Chaoyang Zhang, Jian Chen, and Xiaoman He. Open-world surgical video generation via dual-visual diffusion and dual-annealed generation. *Neural Networks*, page 108281, 2025.

[35] Sabina Martyniak, Joanna Kaleta, Diego Dall'Alba, Michal Naskrket, Szymon Płotka, and Przemysław Korzeniowski. Simuscope: Realistic endoscopic synthetic dataset generation through surgical simulation and diffusion models. *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4268–4278, 2024. URL `https://api.semanticscholar.org/CorpusID:274445994`.

[36] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293, 2014.

[37] Open-H-Embodiment Consortium Nigel Nelson, Juo-Tung Chen, Jesse Haworth, Xinhao Chen, Lukas Zbinden, and etc. Dianye Huang. Open-h-embodiment: A large-scale dataset for enabling foundation models in medical robotics, 2026. URL `https://api.semanticscholar.org/CorpusID:287702178`.

[38] Dimitrios Psychogyios, Emanuele Colleoni, Beatrice Van Amsterdam, Chih-Yang Li, Shu-Yu Huang, Yuchong Li, Fucang Jia, Baosheng Zou, Guotai Wang, Yang Liu, et al. Sar-rarp50: Segmentation of surgical instrumentation and action recognition on robot-assisted radical prostatectomy challenge. *arXiv preprint arXiv:2401.00496*, 2023.

[39] Sampath Rapuri, Lalithkumar Seenivasan, Dominik Schneider, Roger Soberanis-Mukul, Yufan He, Hao Ding, Jiru Xu, Chenhao Yu, Chenyan Jing, Pengfei Guo, Daguang Xu, and Mathias Unberath. Saw: Toward a surgical action world model via controllable and scalable video generation, 2026. URL `https://arxiv.org/abs/2603.13024`.

[40] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[41] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[43] Usha Ruby, Vamsidhar Yendapalli, et al. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10), 2020.

[44] Saumya Saksena. R3d-18 for ucf-101 action recognition. `https://huggingface.co/dronefreak/r3d-18-ucf101`, 2024.

[45] Mehmet Kerem Turkcan, Mattia Ballo, Filippo Filicori, and Zoran Kostic. Towards suturing world models: Learning predictive models for robotic surgical tasks. *arXiv preprint arXiv:2503.12531*, 2025.

[46] Natalia Valderrama, Paola Ruiz, Isabela Hernández, Nicolás Ayobi, Mathilde Verlyck, Jessica Santander, Juan Caicedo, Nicolás Fernández, and Pablo Arbeláez. Towards holistic surgical scene understanding. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 442–452, Cham, 2022. Springer Nature Switzerland.

[47] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei

Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[48] Rongsheng Wang, Junying Chen, Ke Ji, Zhenyang Cai, Shunian Chen, Yunjin Yang, and Benyou Wang. Medgen: Unlocking medical video generation by scaling granularly-annotated medical videos. *arXiv preprint arXiv:2507.05675*, 2025.

[49] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[50] Jiawei Xu, Kai Deng, Zexin Fan, Shenlong Wang, Jin Xie, and Jian Yang. Ad-gs: Object-aware b-spline gaussian splatting for self-supervised autonomous driving. *arXiv preprint arXiv:2507.12137*, 2025.

[51] Jiazhi Yang, Kashyap Chitta, Shenyuan Gao, Long Chen, Yuqian Shao, Xiaosong Jia, Hongyang Li, Andreas Geiger, Xiangyu Yue, and Li Chen. Resim: Reliable world simulation for autonomous driving. *arXiv preprint arXiv:2506.09981*, 2025.

[52] Xiuyu Yang, Bohan Li, Shaocong Xu, Nan Wang, Chongjie Ye, Zhaoxi Chen, Minghan Qin, Yikang Ding, Zheng Zhu, Xin Jin, et al. Orv: 4d occupancy-centric robot video generation. *arXiv preprint arXiv:2506.03079*, 2025.

[53] Zeyu Yang, Zijie Pan, Yuankun Yang, Xiatian Zhu, and Li Zhang. Driving view synthesis on free-form trajectories with generative prior, 2025. URL `https://arxiv.org/abs/2412.01717`.

[54] Zhuoran Yang, Xi Guo, Chenjing Ding, Chiyu Wang, Wei Wu, and Yanyong Zhang. Instadrive: Instance-aware driving world models for realistic and consistent video generation. In *ICCV*, 2025.

[55] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[56] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024.

[57] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023.

[58] Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. *arXiv e-prints*, pages arXiv–2503, 2025.

[59] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2023.

[60] Youwei Zheng, Yuxi Ren, Xin Xia, Xuefeng Xiao, and Xiaohua Xie. Dense2moe: Restructuring diffusion transformer to moe for efficient text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18661–18670, 2025.

[61] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

[62] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

[63] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024.

Appendix

A Formal Motivation

This section of the appendix provides a more discussions on the main design choices in KVLR.

A.1 Why Pixel-Aligned Lifting Helps

The first design choice in KVLR is to lift sparse articulated actions into a dense image-aligned control field before injecting them into the video generator. This subsection formalizes the intuition that raw-action conditioning must learn both spatial alignment and control-feature prediction, while the lifted field externalizes part of the alignment problem.

Let u denote a spatial-temporal token and let $z_t^\star(u)$ be an ideal local control feature sufficient for action-conditioned generation at token u:

$$z_t^\star(u) = \phi^\star(\xi_t(u)), \tag{10}$$

where $\xi_t(u)$ denotes the local physical state relevant to token u, such as part identity, depth, local orientation, projected motion, and motion change.

Under raw-action conditioning, the model predicts a token-wise control feature directly from the global articulated action:

$$\hat{z}_t^{\mathrm{raw}}(u) = g_{\mathrm{raw}}(a_t, u). \tag{11}$$

This requires the learned branch to implicitly approximate both the projection from the global articulated state to token u and the mapping from the resulting local physical state to generator features.

In contrast, KVLR first constructs a pixel-aligned action field,

$$K_t(u) = L_u(a_t), \tag{12}$$

where $L_u(\cdot)$ denotes the deterministic lifting operation at token u. The local control feature is then predicted as:

$$\hat{z}_t^{\mathrm{lift}}(u) = g_{\mathrm{lift}}(K_t(u)). \tag{13}$$

The lifted representation separates the spatial-alignment step from the learned control-feature mapping.

To make this intuition explicit, suppose that the lifted field recovers the relevant local physical state up to an abstract error $\varepsilon_{\mathrm{lift}}$, while a learned projection from raw actions recovers it up to $\varepsilon_{\mathrm{proj}}$. Also suppose that $\phi^\star$ is L_ϕ-Lipschitz and that the learned mappings have approximation errors $\varepsilon_{\mathrm{app}}^{\mathrm{lift}}$ and $\varepsilon_{\mathrm{app}}^{\mathrm{raw}}$. A direct triangle-inequality decomposition gives:

$$\left\|\hat{z}_t^{\mathrm{lift}}(u) - z_t^\star(u)\right\| \lesssim L_\phi \varepsilon_{\mathrm{lift}} + \varepsilon_{\mathrm{app}}^{\mathrm{lift}}, \tag{14}$$

whereas raw-action conditioning yields:

$$\|\hat{z}_t^{\mathrm{raw}}(u) - z_t^\star(u)\| \lesssim L_\phi \varepsilon_{\mathrm{proj}} + \varepsilon_{\mathrm{app}}^{\mathrm{raw}}. \tag{15}$$

These quantities are the decomposition that clarifies the design assumption behind KVLR: pixel-aligned lifting is expected to help when the hand-constructed action field reduces the spatial-alignment burden more than it restricts the representation. This is consistent with the parameter-matched ablation in the main paper, where replacing raw action vectors with the KVA-Field provides a substantial gain before adding hierarchical routing.

A.2 When Hierarchical Routing Is Useful

The second design choice in KVLR is to route lifted action features across physical modalities and motion scales. We interpret this routing mechanism as an inductive bias rather than as a universally stronger function class. A sufficiently large monolithic conditioner can, in principle, approximate heterogeneous control mappings. The motivation for routing is that, under a fixed parameter and computation budget, surgical scenes contain spatial-temporal regimes with different dominant factors: static or low-motion regions often require semantic and geometric consistency, while high-motion regions require dynamic cues and larger motion operators.

Let $\kappa_t(u)$ denote an unobserved control regime for token u, and let $f^{\star}_{\kappa_t(u)}$ be the corresponding local control map. A monolithic conditioner approximates all regimes using a shared operator:

$$\hat{z}_t^{\mathrm{mono}}(u) = f_{\mathrm{mono}}(K_t(u)). \tag{16}$$

A routed conditioner instead selects an expert or sub-expert according to the local control regime:

$$\hat{z}_t^{\mathrm{route}}(u) = f_{\hat{\kappa}_t(u)}(K_t(u)), \tag{17}$$

where $\hat{\kappa}_t(u)$ denotes the selected routing branch.

If the average within-regime approximation error is denoted by $\bar{\varepsilon}$, and if the probability of selecting an unsuitable branch is denoted by η, then the routed error can be informally decomposed as

$$\mathbb{E}_{t,u}\left[\left\|\hat{z}_t^{\mathrm{route}}(u) - z_t^{\star}(u)\right\|\right] \lesssim \bar{\varepsilon} + C\eta, \tag{18}$$

where C is a bounded mismatch constant measuring the cost of choosing an unsuitable expert. This expression separates two intuitive error sources: expert approximation error and routing error. It is not meant to provide an estimable guarantee, because $\bar{\varepsilon}$, η, and C are not measured in the actual model.

This decomposition motivates two aspects of KVLR. First, hierarchical routing can be useful when different physical regimes are easier to approximate with specialized operators than with a single shared pathway. Second, the benefit of specialization depends on routing quality, which motivates the kinematic-prior objectives introduced in the main method. In practice, the value of this inductive bias is evaluated empirically through the parameter-matched ablation comparing raw-action injection, KVA-Field injection, Tier-1 routing, and Tier-1 plus Tier-2 routing.

A.3 Interpretation of Kinematic-Prior Regularization

The kinematic-prior objectives are introduced to bias the router toward physically plausible and temporally stable assignments. They constrain observable aspects of routing behavior that are relevant to surgical motion.

Kinematic-prior adaptive load balancing. The KP-ALB objective aligns modality usage with the physical signal strength in the KVA-Field:

$$\mathcal{L}_{\text{KP-ALB}} = \sum_{i \in \mathcal{M}} \left(\ell_i - \pi_i(K_t)\right)^2, \tag{19}$$

where $\pi_i(K_t)$ denotes the normalized physical prior of modality i, and ℓ_i denotes its routing load. Unlike uniform load balancing, this term does not force all experts to be used equally. Instead, it discourages routing patterns that ignore the relative strength of semantic, geometric, and dynamic action cues.

Spatiotemporal routing consistency. The SRC objective penalizes frame-to-frame routing variation around tool regions:

$$\mathcal{L}_{\mathrm{SRC}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \sum_{h,w} M_{\mathrm{tool}}^{(t)}(h,w) \left\| R_t(h,w) - R_{t-1}(h,w) \right\|_2^2. \tag{20}$$

This term reflects the assumption that routing decisions around articulated tools should change smoothly unless the underlying motion changes. It reduces routing flicker but does not enforce temporal smoothness in static background regions.

Capacity prediction. The capacity-prediction objective trains a lightweight predictor to approximate routing decisions from detached action features:

$$\mathcal{L}_{\mathrm{CP}} = \mathrm{BCE}\left(\sigma(\hat{R}), \mathrm{sg}(R^{\star})\right). \tag{21}$$

This makes routing decisions more predictable from action-derived signals, which later supports the efficient execution policy. Its role is practical rather than theoretical: it helps expose a stable approximation of where full control computation is likely to matter.

Overall, these regularizers reduce the degrees of freedom of the router and make the learned routing easier to interpret.

B Related Work

B.1 Surgical Video Generation

Surgical video generation has recently attracted growing attention for simulation, training, and predictive modeling in robotic surgery [48, 25, 8, 34, 45, 39]. Early progress mainly focuses on scaling surgical video corpora and improving visual realism. MedGen [48] and Ophora [25] build large-scale captioned datasets for generalist and ophthalmic video synthesis, while Endora [23] studies endoscopic video generation as a simulator for minimally invasive scenes. Simulation-assisted efforts such as SimuScope [35] combine surgical simulation with diffusion-based image translation to improve synthetic endoscopic realism. Beyond visual synthesis, recent works explore controllable surgical video generation. SurGen [8] and Open-world [34] rely on language guidance for flexible scene generation, while SurgSora [6] introduces structured RGBD-flow decomposition for controllable surgical video synthesis. HieraSurg [3] further exploits hierarchy-aware segmentation to improve surgical video generation. More recent surgical world-modeling efforts move toward action-conditioned synthesis: SAW [39] conditions video diffusion on lightweight signals such as language, reference scenes, tissue affordance masks, and 2D tool-tip trajectories, while SurgWorld [16] uses generative world modeling and pseudo-kinematics to support surgical policy learning. Open-H-Embodiment [37] also highlights the importance of synchronized medical robotic videos and kinematics for embodied surgical AI. Despite these advances, existing methods still primarily rely on coarse text prompts, 2D trajectory cues, or dense visual priors such as masks, depth, or optical flow. These controls are either insufficient for fine-grained articulated manipulation or costly to obtain at inference time. In contrast, our work targets *action-conditioned* surgical video generation from sparse articulated controls. By lifting kinematics into a pixel-aligned action field and hierarchically routing conditioning across physical modalities and motion scales, KVLR aims to improve control precision, visual fidelity, and efficient inference under accessible robotic action inputs.

B.2 Controllable Generation with Structured Signals

Beyond surgery, controllable video generation has been actively studied in robotics, autonomous driving, and general scene modeling [61, 55, 11, 52, 19, 24, 26, 51, 22, 5, 54, 50, 14, 53, 15, 20, 18, 26]. In robotics, IRASim [63] studies action-to-video prediction for robotic manipulation, while Cosmos-Transfer [1] and RoboTransfer [31] condition synthesis on scene-level geometric cues such as depth or surface normals. In driving and embodied scene generation, UniScene [22] uses hierarchical occupancy priors, and MagicDrive [12] and Vista [13] exploit structured multi-view or latent control for controllable synthesis. Related efforts in controllable rendering and 3D-aware generation also rely on dense scene representations or explicit geometric guidance [28, 58, 27]. These approaches demonstrate the value of structured conditioning, but they typically depend on dense maps, occupancy fields, or other heavy intermediate representations that are difficult to access in robotic surgery, where sensing is limited and occlusion is frequent [52, 58, 22, 27]. Differently, our method focuses on sparse articulated kinematic actions as the primary control interface, and in transforming them into pixel-aligned visual control through hierarchical kinematic-to-visual routing rather than relying on dense geometric priors.

C Discussion on Limitations and Broader Impact

C.1 Limitations

Our framework is developed for action-conditioned surgical video generation under articulated robotic control, and is evaluated on representative suturing-related sub-actions with curated supervision. While the proposed structured kinematic-to-visual control improves faithfulness, fidelity, and efficiency in this setting, extending the framework to broader surgical procedures and more diverse instruments may require additional adaptation in data curation and model scaling. In addition, as with other generative models, quantitative improvements on benchmark metrics do not fully capture all aspects of clinical realism or downstream utility, and human-centered evaluation remains important for future study.

C.2 Broader Impact

This work has the potential to benefit surgical simulation, training, and data augmentation by enabling more controllable video generation from lightweight robotic action signals. At the same time, synthetic surgical videos should not be interpreted as a substitute for real clinical evidence or expert judgment. We do not advocate direct deployment of generated videos for autonomous clinical decision-making. More broadly, responsible use of such models requires appropriate attention to data governance, benchmarking transparency, and the intended educational or research context.

C.3 Asset License and Usage Note

This work builds on several existing datasets, models, and software components, including SAR-RARP for source video data, SAM2 for segmentation-assisted annotation, OpenSora and related pretrained video-generation components for model initialization, and publicly available baseline methods used for comparison. We credit the original creators of these assets in the main paper and references, and use them in accordance with their respective licenses, terms of use, and research-only restrictions where applicable. For all external assets, we follow the usage conditions specified by their official releases or project pages. Any future release of our code, models, or derived benchmark assets will also be conditioned on compatibility with the licenses and usage terms of these underlying resources. We refer to the official releases/pages of these assets for the exact license terms and usage conditions, and will include explicit license identifiers in any public release package.

D More Implementation Details

D.1 Detailed Experimental Setup

Training details. Unless otherwise specified, all models are trained on NVIDIA H20 GPUs with a batch size of 24 using AdamW [33], with a learning rate of 1×10^{-5} and weight decay of 0.01. We split the KASA benchmark into 85% training data and 15% validation data. Our model is trained with a progressive two-stage protocol for stable convergence. In the first stage, we finetune the generator using text-only conditioning to preserve the pretrained video prior and adapt the backbone to the surgical domain. In the second stage, we continue training with joint text-and-action conditioning, enabling the model to learn structured action control on top of the adapted generator. The overall training is implemented on 24 GPUs for approximately 7 days. Unless otherwise stated, generated videos use a spatial resolution of 576×1024, and the generated sequence length is aligned with the corresponding ground-truth clip. For inference, the full model uses 50 denoising steps. The efficient variant, denoted as *Ours-fast*, uses the distilled 4-step student described in Sec. 2.5.

Baseline setup. We compare with Wan-2.2 [47], HunyuanVideo [21], OpenSora-2.0 [61], and Cosmos-H-Surgical [17]. All evaluated baselines are finetuned on KASA under the same train and validation split and the same input protocol for fair comparison. For text-conditioned baselines, we use the original conditioning interface of each model. For action-conditioned comparison, models marked with '*' are augmented with ControlNet-style direct action injection, so that articulated actions are provided as explicit additional controls. Models with '†' leverage dense visual conditions as SurgSora [6], including depth, semantic, and flow maps. During inference, each baseline uses its default sampling setting when applicable. Our method uses the default settings described above unless otherwise specified in the corresponding experiment.

Evaluation metrics. We evaluate three aspects: *action faithfulness*, *generation fidelity*, and *computational efficiency*. For action faithfulness, each generated video is matched to its corresponding KASA annotation by video ID. Let $\Omega \subset \mathbb{Z}^2$ denote the image pixel domain. For an instrument instance m at frame t, we render the articulated annotation into a ground-truth action tube mask $T_t^m \subset \Omega$ by projecting the recovered 3D instrument skeleton into the image plane and drawing a fixed-width tube around the projected skeleton. We also render a thin centerline mask C_t^m and projected articulated keypoints for auxiliary validation. To extract the generated instrument region, SAM2 or Grounded SAM is initialized with the first valid ground-truth action tube mask and propagated through the generated video, producing a predicted binary mask $P_t^m \subset \Omega$ for each instrument instance and frame.

The reported Chamfer Distance (CD) measures the symmetric pixel distance between the generated instrument mask and the target action tube:

$$\mathrm{CD}_t^m = \frac{1}{2}\left(\frac{1}{|P_t^m|}\sum_{p\in P_t^m} d(p, T_t^m) + \frac{1}{|T_t^m|}\sum_{g\in T_t^m} d(g, P_t^m)\right), \tag{22}$$

where $|\cdot|$ denotes the number of foreground pixels and $d(x, A) = \min_{a\in A}\|x - a\|_2$ is the Euclidean distance from pixel x to the nearest foreground pixel in mask A, computed using distance transforms. Lower CD indicates that the generated instrument region is spatially closer to the projected target action.

Temporal IoU (TI) evaluates frame-to-frame consistency of the generated tool region. Let $\bar{P}_t = \bigcup_m P_t^m$ be the union of predicted masks over all instrument instances at frame t. We compute:

$$\mathrm{TI}_t = \frac{|\bar{P}_t \cap \bar{P}_{t-1}|}{|\bar{P}_t \cup \bar{P}_{t-1}|}. \tag{23}$$

Higher TI indicates more temporally stable generated tool masks. Area Flicker (AF) measures relative frame-to-frame area variation:

$$\mathrm{AF}_t = \frac{\big||\bar{P}_t| - |\bar{P}_{t-1}|\big|}{\max(|\bar{P}_{t-1}|, 1)}. \tag{24}$$

Lower AF indicates less abrupt temporal fluctuation in the generated action region. All action-faithfulness metrics are averaged over valid frames and instrument instances with non-empty projected action targets.

For generation fidelity, we report PSNR, SSIM, FID, and FVD. PSNR and SSIM are averaged over all generated frames. FID is computed by treating generated video frames as images. FVD is computed on 17-frame clips to evaluate temporal realism and video-level consistency.

For efficiency, we report inference latency and FLOPs, and additionally report peak memory in the efficient-generation ablations. Unless otherwise stated, all validation metrics are averaged over sequences. The computational efficiency results in the main paper are measured on NVIDIA H20 GPUs using 17-frame clips.

Main benchmark tables report averages over the three action subsets, while uncertainty estimates are provided for the representative quantitative analyses in Sec. F.

D.2 Backbone Instantiation and Training Context

Our generator G is instantiated on top of OpenSora, which consists of a pretrained text encoder, a 3D VAE, and a trainable DiT backbone. In the main paper, we keep Sec. 2.1 intentionally compact and mainly describe the two-stage control decomposition. Here, we provide the omitted implementation context.

Given a reference image I^{ref}, a text prompt y, and an articulated action sequence $\mathrm{a}_{1:T}$, we first encode the visual input into the latent space of the pretrained 3D VAE. The text condition is processed by the pretrained text encoder, while the articulated action sequence is transformed into the Kinematic-to-Visual Action Field described in Sec. D.3. The resulting structured action features are then processed by the hierarchical routing module before being injected into the DiT backbone through a lightweight conditional branch.

Unless otherwise specified, we preserve the original latent dimensionality and temporal tokenization of the OpenSora backbone. The action branch is designed to be additive rather than replacing the original visual-text conditioning pathway. This choice stabilizes optimization in the early stage of finetuning and allows the pretrained generator to progressively absorb structured action control without disrupting its original generative prior.

The full model is trained with the primary video generation objective together with the kinematic-prior routing objectives introduced in the main paper. Articulated supervision is obtained from our curated KASA pipeline, which provides lightweight action signals consisting of a temporally consistent articulated pose and gripper status. Additional details on the routing objectives and efficient generation strategy are provided in Sec. 2.3 and the later supplementary sections.

D.3 Construction of the Kinematic-to-Visual Action Field

Articulated action representation. For each frame t, we represent the articulated action as:

$$a_t = [p_t, r_t, q_t^{\mathrm{sw}}, q_t^{\mathrm{lg}}, q_t^{\mathrm{rg}}] \in \mathbb{R}^9, \tag{25}$$

where the wrist is connected to the shaft by a shaft-to-wrist joint, and the left and right grippers are connected to the wrist by separate wrist-to-gripper joints. $p_t \in \mathbb{R}^3$ and $r_t \in \mathbb{R}^3$ denote the wrist translation and wrist orientation, respectively, with r_t expressed in axis-angle form. The scalar variables q_t^{sw}, q_t^{lg}, and q_t^{rg} denote the shaft-to-wrist, wrist-to-left-gripper, and wrist-to-right-gripper joint states.

Dense field layout. We lift each sparse articulated state to a pixel-aligned tensor:

$$\mathrm{K}_t \in \mathbb{R}^{H \times W \times 9}, \tag{26}$$

whose channels encode part semantics, depth, rotation, velocity, and acceleration:

$$\mathrm{K}_t(h, w) = \big[\mathrm{s}_t(h, w),\ d_t(h, w),\ \rho_t(h, w),\ \mathrm{v}_t(h, w),\ \alpha_t(h, w)\big]. \tag{27}$$

Here, $\mathrm{s}_t \in \mathbb{R}^3$ denotes part-aware semantic channels, d_t is rendered depth, ρ_t is a local rotation descriptor, $\mathrm{v}_t \in \mathbb{R}^3$ is projected motion, and α_t is acceleration magnitude. We use α_t for acceleration to avoid overloading the symbol a_t for articulated action.

Semantic channels. The semantic channels are derived from the Human-in-the-loop Semantic Annotation (HSA) stage of KASA. Specifically, for each clip, we initialize part-aware annotation from a reference frame and obtain temporally propagated masks using a SAM2 model finetuned on SAR-RARP. Annotators then refine the propagated masks to correct leakage, part confusion, and ambiguous boundaries. In our default implementation, the three semantic channels correspond to the articulated instrument parts used throughout the paper: shaft, wrist, and gripper. We rasterize these refined part masks into three binary maps and stack them as:

$$\mathrm{s}_t(h, w) = \big[s_t^{\mathrm{shaft}}(h, w),\ s_t^{\mathrm{wrist}}(h, w),\ s_t^{\mathrm{gripper}}(h, w)\big]. \tag{28}$$

When multiple rendered parts overlap at a pixel, we resolve the assignment using the front-most visible part under the rendered depth order.

Depth and rotation channels. The geometric channels are obtained from the Differentiable Pose Tracking (DPT) stage. DPT estimates temporally consistent articulated poses directly from surgical videos by aligning rendered part-aware silhouettes from a controllable Gaussian representation, initialized from the instrument CAD prior, to the observed segmentation masks. Once the articulated pose M_t is estimated, we render a depth map in the camera view to obtain d_t.

For the rotation channel, we use a local descriptor $\rho_t(h, w)$ derived from the projected articulated orientation of the visible instrument part at pixel (h, w). In practice, this descriptor can be instantiated using the wrist/tool orientation most relevant to the visible part, *e.g.*, projected roll/yaw information or another normalized scalar orientation cue consistent across frames. In all cases, the role of ρ_t is to encode orientation-related geometry that is not captured by depth alone.

Temporal dynamic channels. We compute motion channels from consecutive articulated states:

$$\mathrm{v}_t = \frac{\Pi(\mathrm{M}_t) - \Pi(\mathrm{M}_{t-1})}{\Delta t}, \qquad \alpha_t = \frac{\|\mathrm{v}_t - \mathrm{v}_{t-1}\|_2}{\Delta t}, \tag{29}$$

where $\Pi(\cdot)$ denotes camera projection and M_t is the articulated instrument state at frame t. Intuitively, v_t measures projected motion in image space, while α_t captures the magnitude of temporal motion change. This yields a field that encodes not only where the instrument is, but also how it moves.

Rendering and normalization. All channels are constructed in the image plane and resized to the spatial resolution used by the conditional branch. Binary semantic masks are kept in $[0, 1]$. The depth, rotation, velocity, and acceleration channels are normalized per channel using statistics computed on the training split, which stabilizes optimization across heterogeneous motion magnitudes. Unless otherwise specified, the same normalization parameters are reused during inference.

Why this representation is sufficient. The Kinematic-to-Visual Action Field serves as the minimal structured control basis used by the subsequent routing module. It preserves the accessibility of sparse articulated control, establishes explicit spatial correspondence with the image plane, and separates semantic, geometric, and dynamic factors before routing. This separation is important because the downstream hierarchical router allocates conditional computation according to physical modalities and motion scales rather than learning such disentanglement implicitly from raw control tokens.

D.4 Detailed Hierarchical Routing

Tier-1 cross-modality routing. Here we provide the detailed formulation of the outer router. We use five modality-specific experts corresponding to semantics, depth, rotation, velocity, and acceleration. Each expert processes only its associated channels from the Kinematic-to-Visual Action Field. The outer gating network predicts token-wise routing weights conditioned on the action representation and diffusion timestep. We adopt sparse top-k routing during training, with a capacity schedule that gradually transitions from dense soft routing to sparse expert selection.

Tier-2 intra-modal motion-scale routing. Within each modality branch, we further introduce three sub-experts: a fine-manipulation branch for localized high-frequency motion, a transport-motion branch for larger displacements, and a skip branch for static or weakly changing regions. The inner router operates locally on modality features and selects the appropriate motion-scale operator for each token.

Sparse fusion and stable injection. The final routed action representation is obtained by sparse fusion over the selected Tier-1 experts using re-normalized routing weights. We inject the fused control features into the DiT backbone through zero-initialized additive layers, which preserve the pretrained generative prior at initialization and enable stable action-conditioned finetuning.

Additional architecture details. In practice, we apply the routing branch at selected DiT blocks and match its spatial resolution to the latent token grid used by the generator. We provide the exact layer placement, hidden dimensions, normalization layers, and activation functions in the provided supplementary code.

D.5 Detailed Kinematic-Prior Learning

This section expands the kinematic-prior learning objective introduced in Sec. 2.4. The purpose of these losses is not to force the router to use all experts uniformly, but to make expert usage agree with the physical content of the lifted action field and remain stable across time. In implementation, the auxiliary objective is attached to the action encoder and added to the flow-matching training loss whenever the structured MoE action pathway is enabled.

Notation and routing statistics. Let $K_t \in \mathbb{R}^{B\times T\times H\times W\times 9}$ be the Kinematic-to-Visual Action Field. The nine channels are ordered as semantic part indicators $K_t^{\mathrm{sem}} \in \mathbb{R}^3$, depth $K_t^{\mathrm{dep}} \in \mathbb{R}$, local rotation $K_t^{\mathrm{rot}} \in \mathbb{R}$, velocity $K_t^{\mathrm{vel}} \in \mathbb{R}^3$, and acceleration $K_t^{\mathrm{acc}} \in \mathbb{R}$. The outer router predicts token-wise logits:

$$G_{outer} \in \mathbb{R}^{BT\times|\mathcal{M}|\times H'\times W'}, \qquad \mathcal{M} = \{\mathrm{sem}, \mathrm{dep}, \mathrm{rot}, \mathrm{vel}, \mathrm{acc}\}, \tag{30}$$

where (H', W') is the spatial resolution of the routing grid. We denote the soft routing probability by $P = \mathrm{softmax}(G)$ and the binary top-k routing mask by $A \in \{0,1\}^{BT\times|\mathcal{M}|\times H'\times W'}$. In our default setting, the outer router uses top-$k = 2$ experts per token. For each modality expert i, the realized sparse routing load is measured by:

$$f_i = \mathbb{E}_{b,t,h,w}[A_{b,t,i,h,w}], \qquad p_i = \mathbb{E}_{b,t,h,w}[P_{b,t,i,h,w}], \qquad \ell_i = f_i p_i. \tag{31}$$

Here f_i captures the hard fraction of tokens assigned to expert i, while p_i captures the average soft confidence assigned to that expert. Their product follows sparse MoE load estimation and prevents a high-probability but rarely selected expert, or a frequently selected but low-confidence expert, from being treated as fully utilized.

Kinematic-prior adaptive load balancing. Uniform load balancing is a poor target for surgical manipulation, because the physical causes of appearance change are highly non-uniform across frames and actions. We therefore compute a sample-dependent target distribution directly from K_t. First, K_t is adaptively average-pooled to the router resolution (H', W'). At each routed token, we compute modality energies:

$$\begin{aligned} e_{\mathrm{sem}} &= \|K_t^{\mathrm{sem}}\|_2, \quad e_{\mathrm{dep}} = |K_t^{\mathrm{dep}}|, \quad e_{\mathrm{rot}} = |K_t^{\mathrm{rot}}|, \\ e_{\mathrm{vel}} &= \|K_t^{\mathrm{vel}}\|_2, \quad e_{\mathrm{acc}} = |K_t^{\mathrm{acc}}|. \end{aligned} \tag{32}$$

These terms respectively measure tool-part presence, rendered camera-depth magnitude, articulated rotation magnitude, projected motion magnitude, and acceleration magnitude. The local physical

prior is normalized over modalities:

$$\pi_i(K_t^{b,h,w}) = \frac{e_i(K_t^{b,h,w})}{\sum_{j\in\mathcal{M}} e_j(K_t^{b,h,w})}. \tag{33}$$

The batch-level target used by the loss is the mean physical prior:

$$\bar{\pi}_i(K_t) = \mathbb{E}_t^{b,h,w}\left[\pi_i(K_t^{b,h,w})\right]. \tag{34}$$

The KP-ALB loss then aligns the realized routing load with this physical target:

$$\mathcal{L}_{\text{KP-ALB}} = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \left(\ell_i - \text{sg}(\bar{\pi}_i(K_t))\right)^2, \tag{35}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. This loss differs from standard MoE balancing in two ways. First, the target is not uniform; it changes with the current action field. Second, the target is grounded in interpretable kinematic quantities, so semantic/depth experts are favored in static visible tool regions, rotation experts are favored during wrist articulation, and velocity/acceleration experts are favored during fast tool motion.

Spatiotemporal routing consistency. The SRC term reduces frame-to-frame routing flicker around articulated instruments. Instead of applying temporal smoothness to the entire image, we derive a binary tool mask from the semantic channels of K. Specifically, the semantic channels are max-pooled to (H', W'), and a token is marked as tool-present when any semantic channel is active. Let:

$$M^{\text{tool}} \in \{0,1\}^{B\times T\times 1\times H'\times W'} \tag{36}$$

denote this mask. The continuous routing probability used by SRC is the capacity-predictor probability:

$$R = \sigma(\widehat{A}) \in [0,1]^{B\times T\times|\mathcal{M}|\times H'\times W'}. \tag{37}$$

We penalize temporal differences only inside tool regions:

$$\mathcal{L}_{\text{SRC}} = \frac{\sum_{b,t=2}^{T}\sum_{i,h,w} M^{\text{tool}}_{b,t,1,h,w}\left(R_{b,t,i,h,w} - R_{b,t-1,i,h,w}\right)^2}{|\mathcal{M}|\sum_{b,t=2}^{T}\sum_{h,w} M^{\text{tool}}_{b,t,1,h,w}}. \tag{38}$$

This formulation stabilizes expert selection on moving tools and tool tips, while avoiding unnecessary constraints on static background regions. If a clip contains only one frame after temporal sampling, the SRC term is set to zero.

Tier-2 routing stabilizer. Within each modality expert, KVLR further routes features to three sub-experts: fine manipulation, transport motion, and skip. The inner router uses top-1 routing. Although the main paper groups this under hierarchical routing, the implementation includes a small auxiliary stabilizer to prevent the collapse of the Tier-2 router:

$$\mathcal{L}_{\text{sub}} = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} |\mathcal{S}|\sum_{s\in\mathcal{S}} f_{m,s}p_{m,s}, \qquad \mathcal{S} = \{\text{fine}, \text{transport}, \text{skip}\}. \tag{39}$$

Here $f_{m,s}$ and $p_{m,s}$ are the hard assignment fraction and mean soft probability of sub-expert s inside modality branch m. This term is lightweight and acts as an implementation-level stabilization term for the nested router.

Capacity schedule and final objective. The routing path is trained with a three-stage capacity schedule. Stage I uses dense softmax fusion so that all modality experts receive the gradient signal. Stage II linearly interpolates between dense fusion and sparse top-k fusion. Stage III uses fully sparse top-k fusion. In the default configuration, Stage I occupies the first 40% of training and Stage II ends at 75% of training. The auxiliary losses above are active during training whenever a routing mask is available.

The primary generation objective is the flow-matching loss. With latent endpoints x_0 and x_1, the training target is:

$$v_t = (1-\sigma_{\min})x_1 - x_0, \tag{40}$$

and the model prediction is trained with mean-squared error:

$$\mathcal{L}_{\text{Flow}} = \mathbb{E}\left[\|\hat{v}_t - v_t\|_2^2\right]. \tag{41}$$

For image-to-video conditioning, the implementation can exclude conditioned reference frames from this MSE so that the loss focuses on generated frames.

The full training objective for the structured action model is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Flow}} + \lambda_{\text{kp}}\mathcal{L}_{\text{KP-ALB}} + \lambda_{\text{src}}\mathcal{L}_{\text{SRC}} + \lambda_{\text{cp}}\mathcal{L}_{\text{CP}} + \lambda_{\text{sub}}\mathcal{L}_{\text{sub}}. \tag{42}$$

Unless otherwise specified, we use:

$$\lambda_{\text{kp}} = 0.01, \qquad \lambda_{\text{src}} = 0.005, \qquad \lambda_{\text{cp}} = 0.01, \qquad \lambda_{\text{sub}} = 0.005. \tag{43}$$

In the compact objective in Sec. 2.4, $\mathcal{L}_{\text{sub}}$ can be viewed as a small nested-router stabilization term associated with hierarchical routing, while the three primary kinematic-prior losses are KP-ALB, SRC, and CP.

Capacity-predictor supervision. The capacity predictor is a lightweight 1×1 convolutional network that predicts which outer experts will be active without recomputing full top-k routing at inference time. It receives the detached shared action feature c_{act} and outputs raw logits:

$$\widehat{A} = Q_\psi(\text{sg}(c_{\text{act}})) \in \mathbb{R}^{BT\times|\mathcal{M}|\times H'\times W'}. \tag{44}$$

The target is the router-induced binary routing mask A. We train the predictor with binary cross-entropy with logits:

$$\mathcal{L}_{\text{CP}} = \text{BCE}\Big(\widehat{A}, \text{sg}(A)\Big). \tag{45}$$

Detaching c_{act} and A makes this term supervise the predictor rather than changing the main router to become easier to predict. During dense warm-up, all experts contribute through softmax fusion, but the same top-k mask induced by the router probabilities is used as a synthetic target for this loss. During sparse training, A is the actual top-k routing decision.

At inference, the predictor probability $\sigma(\widehat{A})$ is compared with an expert-wise threshold. These thresholds are updated during training by an exponential moving average of expert-wise quantiles:

$$\tau_i \leftarrow \beta\tau_i + (1-\beta)\,\text{Quantile}_{1-\bar{a}_i}\Big(\sigma(\widehat{A}_i)\Big), \tag{46}$$

where $\bar{a}_i = \mathbb{E}[A_i]$ is the observed fraction of routed tokens for expert i and $\beta = 0.95$ in our implementation. The quantile level is clamped for numerical stability, but the threshold is updated even when an expert receives nearly zero or nearly all tokens. This avoids positive-feedback expert starvation: an unused expert still receives an updated threshold instead of being permanently suppressed.

D.6 Detailed Action-adaptive Efficient Generation

This section provides additional details for the efficient generation extension in Sec. 2.5. The goal is to reduce inference cost while preserving the action-conditioned behavior of the full structured-control model. The key principle is to reuse the same signals that drive action-conditioned generation of structured action lifting and hierarchical routing to determine where dense action-control computation remains necessary.

Overview. The efficient model combines two complementary mechanisms. First, a few-step student reduces the number of denoising evaluations. Second, an action-adaptive execution policy modulates the cost of action-control updates within each step. Thus, acceleration is achieved at both the sampling-depth level and the per-step control-computation level.

Let G_t denote the full teacher and G_s the efficient student. Both receive the same reference image I^{ref}, text prompt y, and articulated action sequence $\text{a}_{1:T}$. The teacher provides full structured-control supervision, while the student operates with a reduced denoising schedule and an execution policy induced by the control pathway.

Action-aware distillation. The student preserves the teacher's conditioning interface, including structured action lifting, hierarchical routing, and action-control injection. We distill the teacher at three levels.

Prediction-level distillation. We first match the teacher's prediction target:

$$\mathcal{L}_{\text{pred}} = \mathbb{E}\left[\left\|\hat{\mathrm{z}}^{\,s} - \mathrm{sg}\big(\hat{\mathrm{z}}^{\,t}\big)\right\|_2^2\right], \tag{47}$$

where $\hat{\mathrm{z}}^{\,s}$ and $\hat{\mathrm{z}}^{\,t}$ denote the student and teacher velocity/noise predictions at the same training state, and $\mathrm{sg}(\cdot)$ denotes stop-gradient.

Routing distillation. Since the routing pathway encodes which physical modalities and motion operators are active, we further align the outer routing distribution, the inner sub-expert routing distribution, and the skip probability:

$$\mathcal{L}_{\text{route}} = \mathrm{KL}\big(\mathrm{R}^t_{\text{out}} \,\|\, \mathrm{R}^s_{\text{out}}\big) + \mathrm{KL}\big(\mathrm{R}^t_{\text{in}} \,\|\, \mathrm{R}^s_{\text{in}}\big) + \mathrm{BCE}\big(p^s_{\text{skip}}, \mathrm{sg}(p^t_{\text{skip}})\big)\,. \tag{48}$$

Control-path distillation. We additionally preserve selected control-path signals:

$$\mathcal{L}_{\text{ctrl}} = \sum_{h \in \mathcal{H}} \left\|h^s - \mathrm{sg}\big(h^t\big)\right\|_2^2, \tag{49}$$

where $\mathcal{H}$ includes intermediate action-control features and structured action statistics, such as motion energy and tool masks. This term keeps the student aligned with the teacher's action-conditioning behavior even under a reduced denoising schedule.

The overall distillation objective is:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{pred}} + \lambda_r \mathcal{L}_{\text{route}} + \lambda_c \mathcal{L}_{\text{ctrl}}. \tag{50}$$

Action significance estimation. Not all spatio-temporal locations require the same amount of control computation. Surgical videos contain highly dynamic tool tips and contact regions, but also large static or slowly changing areas. We therefore estimate a token-level action significance score using the structured control pathway:

$$s(u) = \alpha e_{\text{motion}}(u) + \beta m_{\text{tool}}(u) + \gamma c_{\text{route}}(u) + \eta q_{\text{fine}}(u) - \delta p_{\text{skip}}(u), \tag{51}$$

where e_{motion} is computed from the velocity and acceleration channels of the lifted action field, m_{tool} is derived from the semantic tool map, c_{route} is the confidence of the outer routing distribution, q_{fine} is the probability of selecting the fine-manipulation sub-expert, and p_{skip} is the skip probability. The score is normalized per sample to obtain a token-level significance map. In our default implementation, we use the explicit weighted formulation above for interpretability and stable optimization.

Selective action-control execution. Given $s(u)$, tokens are assigned to three execution modes:

$$\pi(u) \in \{\text{full}, \text{light}, \text{reuse}\}. \tag{52}$$

We use budgeted top-ratio selection: the top fraction of tokens receives full updates, the next fraction receiveslight updates, and the remaining tokens use reuse. In the default setting, the full and light ratios are 0.2 and 0.3, respectively.

In the current implementation, the execution policy modulates the action-control residual injected into the student. Full tokens receive the complete action-control update, light tokens receive a scaled update, and reuse tokens use cached or suppressed residuals. This design is minimally invasive to the backbone architecture: the student generator remains unchanged, while the structured control branch induces non-uniform conditioning strength and computation. Unlike generic token pruning, the execution policy is not based solely on visual redundancy, but is induced by articulated motion, tool presence, routing confidence, and learned skip behavior.

Temporal refresh and feature caching. We further derive a frame-level refresh policy from the average significance of each frame. Let $\bar{s}(t)$ denote the mean significance over frame t. We define the refresh interval as:

$$r(t) = \begin{cases} 1, & \bar{s}(t) \geq \tau_h, \\ 2, & \tau_m \leq \bar{s}(t) < \tau_h, \\ K, & \bar{s}(t) < \tau_m, \end{cases} \tag{53}$$

where high-significance frames are refreshed every step and low-significance frames are refreshed less frequently. During inference, cached action-control residuals can therefore be reused for low-significance regions, while high-significance regions continue to receive dense updates. This temporal

reuse further reduces the cost of structured control on stable segments without changing the main sampling path.

Budget-aware training. To make the trade-off between speed and quality explicit, we introduce a budget regularizer. Let ρ_{full}, ρ_{light}, and ρ_{refresh} denote the observed full-update, light-update, and refresh ratios. We estimate the active compute ratio as:

$$\rho_{\text{compute}} = w_f \rho_{\text{full}} + w_l \rho_{\text{light}}, \tag{54}$$

and optimize the target-budget objective:

$$\mathcal{L}_{\text{budget}} = \left|\rho_{\text{compute}} - \rho_{\text{target}}\right| + w_r \left|\rho_{\text{refresh}} - \rho^{\star}_{\text{refresh}}\right|. \tag{55}$$

For learnable significance weights, we additionally use the mean normalized significance as a differentiable proxy for active computation.

To suppress flickering in lightly updated or reused regions, we further impose temporal feature consistency:

$$\mathcal{L}_{\text{temp}} = \sum_{(u,t)\in\Omega_{\text{light/reuse}}} \left\|\mathrm{f}^s(u,t) - \mathrm{sg}(\mathrm{f}^s(u,t-1))\right\|_2^2, \tag{56}$$

where $\Omega_{\text{light/reuse}}$ denotes the set of tokens assigned to light or reuse modes. The full efficient objective is:

$$\mathcal{L}_{\text{eff}} = \mathcal{L}_{\text{distill}} + \lambda_b \mathcal{L}_{\text{budget}} + \lambda_t \mathcal{L}_{\text{temp}}, \tag{57}$$

where the coefficients are set as $\lambda_b = 0.5$, $\lambda_t = 0.35$.

Training protocol and implementation notes. The teacher remains unchanged and provides prediction, routing, skip, and control-path supervision. The student uses a four-step generation setting by default. In the first stage, we optimize only the distillation objective in Eq. (50) to stabilize the few-step student. In the second stage, we enable $\mathcal{L}_{\text{budget}}$ and $\mathcal{L}_{\text{temp}}$ to train the adaptive execution policy. The resulting efficient extension stays close to the original structured-control model while exposing an interpretable trade-off between generation speed, visual quality, and action fidelity.

E Supplementary Details on KASA Benchmark Construction

This section provides additional details on the construction of the **Kinematic Action-centric Surgical (KASA)** benchmark introduced in the main paper. Our goal is not to build a generic surgical video corpus, but to curate a benchmark that explicitly supports *action-conditioned* video generation under sparse articulated control. Accordingly, the benchmark is designed to provide lightweight yet visually grounded supervision that can be transformed into the structured control representation used by our model. In particular, each sequence is associated with a temporally consistent action signal, consisting of articulated pose and gripper status, which later supports the construction of semantic, depth, rotation, velocity, and acceleration-aware control cues.

E.1 Design Objective and Benchmark Scope

A central motivation of this work is that existing controllable surgical video generation resources do not provide a suitable middle ground between two extremes. On one hand, text descriptions are lightweight but too coarse to specify the fine-grained tool motions and contact transitions that drive surgical scene evolution. On the other hand, dense visual priors such as optical flow, full segmentation maps, or manually designed intermediate annotations are expensive to obtain and often unavailable at inference time in realistic robotic settings. The KASA benchmark is constructed to address this gap.

The design principle of KASA is therefore to pair *realistic surgical video sequences* with *compact articulated action supervision* that remains physically meaningful and visually projectable. Rather than storing only categorical labels or global task descriptions, KASA provides an articulated representation that preserves the tool pose, orientation, and gripper status over time. This makes the benchmark particularly suitable for studying controllable generation under sparse action inputs, where the challenge is not only to synthesize plausible appearance, but also to maintain visual evolution consistent with the underlying surgical manipulation.

Another design consideration is alignment with the model structure introduced in the main paper. Our method decomposes action conditioning into multiple physical factors, including part semantics,

geometry, orientation, and motion. For this reason, the benchmark construction pipeline is organized around two complementary supervision sources:

- **Human-in-the-loop Semantic Annotation (HSA)**, which provides reliable part-aware semantic masks for articulated instrument components;
- **Differentiable Pose Tracking (DPT)**, which recovers temporally consistent articulated geometry directly from videos.

Together, these two branches provide the minimal supervision needed to bridge sparse articulated control and dense video synthesis.

E.2 Task-Driven Sequence Selection

We curate KASA from the SAR-RARP dataset, focusing on robotic surgical sequences performed with da Vinci Large Needle Drivers. The final benchmark contains 1,047 scenes and 105,175 frames. Rather than uniformly sampling all available surgical actions, we deliberately focus on three representative sub-actions: *needle grasping*, *needle puncture*, and *knotting*.

This task selection is driven by the type of action variability we want the benchmark to capture. These three sub-actions jointly span a broad range of articulated surgical dynamics. Needle grasping often contains localized pose adjustment and contact preparation. Needle puncture requires coordinated tool reorientation and short-horizon motion transitions around tissue interaction. Knotting further introduces more complex trajectories, including larger transport motion, interaction-induced deformation, and temporally extended manipulation. As a result, the benchmark contains both high-velocity transport motion and low-velocity fine manipulation, which is important for evaluating the motion-scale specialization behavior of our hierarchical routing framework.

To preserve temporal continuity for pose tracking while keeping the data manageable, all videos are downsampled to 30 FPS. After preprocessing, each sequence contains between 55 and 235 frames. This sequence length is sufficient to preserve meaningful action evolution while avoiding excessive redundancy. Importantly, the resulting clips remain challenging due to articulated motion, frequent self-occlusion, viewpoint variation, and complex instrument interaction of tissue. These properties make KASA a suitable benchmark not only for generation fidelity, but also for evaluating whether a model can remain faithful to sparse action control under heterogeneous surgical dynamics.

E.3 Human-in-the-loop Semantic Annotation

The first branch of the KASA pipeline is devoted to obtaining reliable part-aware semantic supervision. Because our action representation later separates physical modalities, it is important that the semantic component be clean and unambiguous. In robotic surgery, however, instrument appearance is challenging to segment consistently due to specular highlights, motion blur, partial occlusion, and visual similarity across articulated parts. Therefore, naively relying on automatic segmentation tends to produce part confusion, boundary leakage, or inconsistent labels over time.

To address this issue, we adopt a Human-in-the-loop Semantic Annotation (HSA) strategy. For each video, we begin from a reference frame and manually specify a small set of wrist keypoints used to initialize the geometric tracking stage described later. In parallel, we estimate part-level segmentation masks for the main articulated instrument components of the shaft, wrist, and gripper, using a SAM2 model finetuned on SAR-RARP. These predictions are then temporally propagated across frames to provide an initial dense annotation.

Automatic propagation alone is not sufficient for the level of semantic precision required by structured action conditioning. We therefore refine the propagated masks through interactive human correction. The purpose of this stage is not simply to improve visual mask quality in a generic sense, but specifically to eliminate errors that would contaminate the semantic branch of the action representation. Typical corrections include:

- resolving part confusion between adjacent articulated components;
- correcting mask leakage into background tissue or other instruments;
- refining ambiguous boundaries under self-occlusion or motion blur;

- enforcing temporally consistent labeling across consecutive frames.

The final output of HSA is a set of high-quality part-aware semantic masks for each sequence. These masks form the basis of the semantic channels in the Kinematic-to-Visual Action Field. In practice, the three semantic channels correspond to the shaft, wrist, and gripper regions, respectively. This explicit part decomposition is important because different articulated regions often play different functional roles during manipulation, and collapsing them into a single foreground mask would weaken the structured control signal.

E.4 Differentiable Pose Tracking

While HSA provides semantic supervision, action-conditioned generation also requires continuous geometric and motion information. For this reason, the second branch of the KASA pipeline recovers temporally consistent articulated pose trajectories directly from videos. Our goal is to avoid reliance on external robot logs or dedicated hardware tracking systems, since such signals are often unavailable or difficult to synchronize in retrospective video collections. We therefore design a video-based **Differentiable Pose Tracking (DPT)** procedure.

At a high level, DPT formulates articulated pose estimation as a render-and-compare problem. We start from a controllable geometric representation initialized from instrument CAD priors. Given a candidate articulated pose, we render part-aware observations in the camera view and compare them against the observed video evidence. This allows the optimization to remain tied to the image plane while still preserving a physically meaningful articulated parameterization.

The optimization objective is designed to capture both global alignment and local manipulation details. A standard part-aware silhouette alignment term encourages the rendered geometry to match the observed instrument masks. To better capture fine-grained distal motion, especially around the tool tip and gripper region, we additionally use a targeted matching constraint for the articulated end region. This is important because small pose errors near the distal components can lead to disproportionately large errors in the inferred manipulation dynamics.

Temporal coherence is introduced in two ways. First, the initial frame is anchored using Perspective-n-Point initialization from manually specified wrist keypoints. Second, for subsequent frames, we propagate pose estimates using temporally local correspondences. Concretely, wrist-related visual features are tracked across time and lifted to 3D using rendered depth, producing robust 2D-3D correspondences for coarse initialization before iterative optimization. This design stabilizes the trajectory and reduces the chance of drifting into implausible local minima.

In addition, long sequences may still contain severe occlusion or abrupt appearance changes that temporarily degrade tracking quality. To improve robustness, DPT maintains a keyframe memory pool of high-confidence articulated states. When the current tracking quality drops, the system can recover by matching the observation to this memory pool and reinitializing pose estimation through a new PnP stage. This reinitialization mechanism is particularly useful in clips with self-occlusion or rapid tool reorientation.

The output of DPT is a temporally consistent articulated pose sequence for each video. This output provides the geometric foundation in KASA. Once the articulated states are estimated, they can be rendered or projected into multiple visual control channels, including depth, local rotation, projected velocity, and acceleration-related cues.

E.5 Quality Validation of KASA Dataset

We validate the quality of KASA annotations through a render-and-compare consistency check. For each recovered articulated pose, we re-pose the controllable instrument Gaussian representation and render its part-aware silhouette into the imaging space. The rendered silhouette is then compared with the corresponding segmentation mask using the Dice score. This provides a direct geometric measure of whether the recovered pose explains the observed instrument region. We use this score as a quality indicator for DPT and remove low-confidence cases with poor silhouette-mask agreement before constructing the final action-supervised dataset.

Tab. 9 reports the Dice scores of KASA annotations across three surgical sub-actions: *NeedleGrasping*, *Knotting*, and *NeedlePuncture*. We evaluate the overall instrument region and three articulated parts, including the wrist, shaft, and gripper. We compare DPT with and without matching-based

Table 9: Quality validation of KASA annotations. Dice scores are computed between rendered silhouettes from recovered poses and segmentation masks.

Setting	Surgical Task	Dice Score ↑			
		Overall Instrument	Wrist Part	Shaft Part	Gripper Part
w/ Mem-pool Recovery	NeedleGrasping	0.967 ± 0.056	0.892 ± 0.031	0.978 ± 0.016	0.775 ± 0.067
	Knotting	0.966 ± 0.027	0.885 ± 0.047	0.979 ± 0.020	0.757 ± 0.082
	NeedlePuncture	0.947 ± 0.081	0.887 ± 0.038	0.982 ± 0.015	0.787 ± 0.070
w/o Mem-pool Recovery	NeedleGrasping	0.936 ± 0.063	0.864 ± 0.051	0.946 ± 0.049	0.758 ± 0.070
	Knotting	0.937 ± 0.029	0.852 ± 0.047	0.947 ± 0.022	0.726 ± 0.082
	NeedlePuncture	0.919 ± 0.085	0.841 ± 0.067	0.953 ± 0.035	0.749 ± 0.073

recovery using memory pool. Without memory-pool recovery, the tracker continues optimization from the current pose estimate when tracking quality drops, rather than retrieving reliable historical pose candidates for re-initialization. As a result, pose drift can accumulate and lead to complete tracking loss, producing missing or severely misaligned predictions with lower Dice scores. With memory-pool recovery, reliable historical pose candidates are used to re-initialize pose optimization once low-quality tracking is detected. As shown in Tab. 9, memory-pool recovery improves silhouette-mask alignment across most tasks and parts, suggesting more stable pose recovery under challenging surgical dynamics. Remaining low-Dice cases are mainly caused by unstable segmentation under complex illumination, instruments moving partially outside the camera field of view, or severe tissue occlusion that hides the instrument and disrupts pose tracking. In this way, the dice-based validation helps identify unreliable annotations and improves the consistency of the final KASA dataset.

E.6 From Curated Supervision to Structured Action Control

The KASA benchmark is constructed not merely as an annotation resource, but as the data foundation for the structured action representation used by our generation framework. The HSA and DPT branches serve different but complementary roles in this process.

HSA provides reliable *part-aware semantics*. These labels determine where each articulated component appears in image space and prevent semantic ambiguity between the shaft, wrist, and gripper regions. DPT provides temporally consistent *articulated geometry and motion*. These trajectories determine how the instrument is positioned, oriented, and displaced over time. By combining the outputs of both branches, we can derive the set of physically interpretable channels used in the Kinematic-to-Visual Action Field.

More specifically, the semantic masks obtained from HSA are rasterized into part-aware semantic maps. The articulated poses recovered by DPT are used to render depth and local orientation descriptors, and to compute projected velocity and acceleration cues over time. These quantities are then aligned in image space to form the structured control tensor introduced in the main paper. Therefore, the purpose of KASA is not only to provide a benchmark for evaluation, but also to make the sparse articulated control signal learnable by transforming it into a visually grounded supervision source.

E.7 Discussion

We emphasize that the contribution of KASA is not to propose a standalone tracking or annotation system optimized for surgical reconstruction as an end goal. Instead, the benchmark is constructed to support a specific research question: *can realistic surgical videos be generated faithfully from sparse articulated action control ?* From this perspective, the most important property of KASA is that it exposes a compact action interface that remains consistent with dense visual scene evolution.

This benchmark design offers two practical advantages. First, the action signal is lightweight and closer to the type of supervision or control interface that can plausibly be used in robotic settings. Second, because the supervision remains visually grounded, it can be directly transformed into structured intermediate control for generation. We believe this makes KASA a useful benchmark for studying controllable surgical video generation beyond text-only conditioning and beyond heavy dense priors that are difficult to access during inference.

Table 10: KASA usability validation through instrument segmentation. Standard perception baselines are trained on the KASA training split and evaluated on the held-out validation split.

Method	Part mIoU ↑	Tool IoU ↑	Boundary F-score ↑
U-Net [42]	68.4	86.7	71.5
DeepLabV3+ [4]	71.8	88.9	74.2
SegFormer-B1 [49]	75.6	91.3	77.8
Mask2Former-S [7]	**77.1**	**92.0**	**79.4**

Table 11: Visual-to-action consistency on KASA. Standard video encoders predict action trajectories from held-out validation clips.

Method	Trans. MAE (mm) ↓	Rot. MAE (deg) ↓	Gripper Acc. (%) ↑	DTW ↓
Mean trajectory	9.84	13.72	61.5	1.00
R3D-18 regressor [44]	5.62	8.41	82.3	0.58
Video Swin-T regressor [32]	4.91	7.36	86.7	0.51
TimeSformer-S regressor [9]	**4.58**	**6.92**	**88.1**	**0.47**

F Additional Experiments

F.1 Benchmark Usability Validation of KASA

The previous section focuses on evaluation independence. We further validate KASA as a standalone benchmark artifact using only the final KASA annotations, without relying on intermediate annotations or tracking outputs. Since intermediate correction states are not retained for all videos, we do not claim a full audit of every annotation step. Instead, we provide end-to-end usability checks showing that the final annotations are learnable, visually grounded, and useful for downstream evaluation.

Instrument part segmentation. We first train standard segmentation baselines on the KASA training split and evaluate them on the held-out validation split. The models predict four classes: background, shaft, wrist, and gripper. We report mean IoU over instrument parts, foreground IoU for the merged tool region, and boundary F-score. As shown in Tab. 10, standard perception models obtain consistent validation performance, suggesting that the final semantic annotations are sufficiently coherent to support downstream perception tasks.

Visual-to-action consistency. To assess whether the final action supervision is visually grounded, we train video-to-action regressors that predict articulated trajectories from the corresponding video clips. The models are trained only on the KASA training split and evaluated on held-out validation clips. We report translation MAE, rotation MAE, gripper-state accuracy, and trajectory DTW distance. As shown in Tab. 11, learned video encoders predict the action trajectories substantially better than a mean-trajectory baseline, suggesting that the final action supervision is correlated with observable tool motion.

Downstream perception with synthetic augmentation. We finally examine whether KASA can support downstream perception evaluation and whether action-conditioned synthetic videos provide useful training data. We train a SegFormer-B1 [49] segmentation model under different augmentation settings and evaluate all models on the same real KASA validation split. Synthetic videos are generated from the training split only, and validation videos are never used for generation or training. As shown in Tab. 12, adding synthetic data from KVLR improves real validation performance more consistently than adding synthetic data from the strongest baseline. This suggests that KASA provides a meaningful downstream benchmark and that action-conditioned generation yields useful synthetic supervision beyond improving generation metrics.

F.2 Evaluation Independence and Bias Control

A potential concern is that the KASA benchmark construction, the proposed Kinematic-to-Visual Action Field, and the action-faithfulness metrics all involve articulated surgical actions. We therefore provide additional analyses to clarify the separation between supervision, conditioning, and evaluation, and to verify that the reported gains are not merely due to optimizing a representation-specific metric.

Table 12: Downstream perception evaluation on KASA with synthetic data augmentation. SegFormer-B1 [49] is trained under different augmentation settings and evaluated on the real KASA validation split.

Training Data	Part mIoU ↑	Tool IoU ↑	Boundary F-score ↑	Temp. Consistency ↑
Real only	75.6	91.3	77.8	84.5
Real + Cosmos-H-Surgical* [17]	76.7	91.9	78.6	85.2
Real + KVLR-fast	78.4	92.8	80.1	86.8
Real + KVLR	**79.6**	**93.4**	**81.0**	**87.5**

Table 13: Cross-evaluator validation of action faithfulness. Grounded SAM [41] is independently trained and is not used for KASA construction or KVLR training. Results are averaged over the three KASA action subsets.

Evaluator	Method	CD ↓	TI ↑	AF ↓
Grounded SAM [41]	Cosmos-H-Surgical* [17]	94.65	0.78	8.52
Grounded SAM [41]	KVLR	86.90	0.89	4.48
SAM2 [40]	Cosmos-H-Surgical* [17]	101.34	0.75	9.17
SAM2 [40]	KVLR	90.82	0.86	5.03

Separation between action supervision, model conditioning, and evaluation. KASA provides temporally consistent articulated supervision obtained from human-in-the-loop semantic annotation and differentiable pose tracking. During training, KVLR uses this supervision to construct the Kinematic-to-Visual Action Field as a conditioning input. During evaluation, however, the generated videos are first processed by an external visual extractor to obtain instrument masks or trajectories, and these extracted visual measurements are then compared with the projected action references. Therefore, the evaluation does not directly access the internal routed action features or the model's conditioning branch. Nevertheless, because the default evaluator uses the same semantic categories as KASA, we further introduce representation-independent checks below.

Cross-evaluator action-faithfulness evaluation. To reduce dependence on the annotation model used during KASA construction, we evaluate generated videos with an independent instrument extractor that is not used for constructing KASA or training KVLR. Specifically, we use a separately trained surgical instrument segmentation model (Grounded SAM [41]), and recompute Chamfer Distance (CD), Temporal IoU (TI), and Area Flicker (AF) on the validation set. As shown in Tab. 13, KVLR remains consistently better than the strongest action-conditioned baseline under both evaluators. This suggests that the improvement is not specific to the evaluation pipeline.

Trajectory-level verification independent of mask overlap. Mask-overlap metrics may still favor methods that better match the projected action masks. We therefore evaluate action consistency using trajectory-level measurements extracted from generated videos. For each generated clip, we estimate the visible tool centerline and tip trajectory using Grounded SAM [41], and compare them with the projected reference trajectory. We report mean centerline error, endpoint error, and motion-direction accuracy. As shown in Tab. 14, KVLR also improves trajectory-level alignment, indicating that the generated motion better follows the specified articulated action beyond mask-overlap agreement.

Action perturbation sanity check. Finally, we verify that KVLR uses the provided action trajectory rather than only relying on the reference image or dataset prior. We evaluate three perturbed-control settings: *shuffled action*, where the action trajectory is randomly sampled from another clip of the same category; *wrong-category action*, where the action is sampled from a different sub-action category; and *reversed action*, where the temporal order of the action sequence is reversed. Tab. 15 shows that perturbing the action substantially degrades action-faithfulness metrics while only moderately affecting image-level fidelity. This confirms that the action input has a causal influence on the generated motion, and that the default improvements are not caused by simply replaying appearance priors from the reference image.

F.3 Quantitative Analysis of Hierarchical Expert Specialization

Fig. 7 further quantifies how the learned routing behavior evolves with motion intensity. We compute clip-level routing statistics over motion-magnitude quantiles and report the corresponding means and 95% confidence intervals. At Tier 1, the activation mass gradually shifts away from the semantic

Table 14: Trajectory-level action verification with an independent visual tracker. Lower centerline and endpoint errors are better; higher direction accuracy is better.

Method	Centerline Error (px) ↓	Endpoint Error (px) ↓	Direction Acc. (%) ↑
Cosmos-H-Surgical* [17]	11.83	15.62	71.4
KVLR-fast	8.05	10.24	82.1
KVLR	**7.21**	**9.13**	**84.8**

Table 15: Action perturbation sanity check for KVLR. Perturbing the input action reduces action faithfulness, showing that the model depends on the specified action trajectory. Results are averaged over the validation subset.

Control Input	CD ↓	TI ↑	AF ↓	FID ↓
Matched action	**86.90**	**0.89**	**4.48**	**14.38**
Shuffled action	126.75	0.64	10.82	18.94
Wrong-category action	138.42	0.58	12.36	21.15
Reversed action	119.63	0.67	9.74	17.88

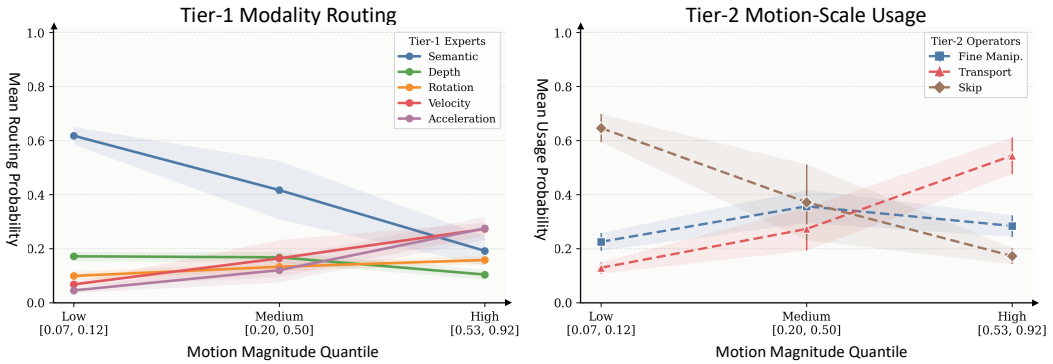

Figure 7: Motion-specialized routing statistics. We group tokens by motion-magnitude quantiles and report the mean routing mass with 95% confidence intervals. At Tier 1, expert usage shifts from semantic-dominant routing in low-motion regions to velocity- and acceleration-dominant routing as motion increases, indicating that the model allocates more computation to dynamic cues when articulated changes become larger. At Tier 2, the routing pattern transitions from skip-dominant behavior in low-motion bins to transport-dominant behavior in high-motion bins, while fine-manipulation operators are most active in the intermediate regime. These results quantitatively confirm that the proposed hierarchical routing learns a motion-dependent decomposition of surgical control rather than a static or uniform expert allocation.

branch and toward the velocity and acceleration branches as motion magnitude increases, showing that the router does not amplify all experts uniformly, but selectively emphasizes dynamics-sensitive pathways when stronger articulated motion is present. At Tier 2, the operator distribution changes from skip-dominant behavior in low-motion regions to transport-dominant behavior in high-motion regions, while fine-manipulation experts remain most competitive in the intermediate range. This intermediate regime also exhibits wider confidence intervals, suggesting that it contains the most heterogeneous micro-dynamics across clips, where subtle local adjustments and larger motion transitions co-exist. Overall, these statistics provide quantitative evidence that hierarchical routing captures a meaningful motion-specialized control decomposition, aligning low-motion states with semantic/static processing and high-motion states with transport-oriented dynamic computation.

F.4 Detailed Efficient-Generation Ablation Breakdown

We expand the main efficiency ablation to isolate what each component contributes. For every row, we report the latency and FLOP speedup relative to the 50-step teacher, the marginal compute saving relative to the immediately preceding row, the CD-based control retention, and the FID increase relative to the teacher. The control-retention ratio is computed as $\mathrm{CD}_{\mathrm{teacher}}/\mathrm{CD}_{\mathrm{method}}$, since lower CD indicates better action alignment.

Table 16: Expanded Efficient-Generation Ablation. All metrics are evaluated on the KASA validation split. Latency and FLOPs are measured per 17-frame clip.

Group	Setting	Latency (s) ↓	FLOPs (PF/clip) ↓	CD ↓	FID ↓
Teacher	Full 50-step structured generator	4.64	6.94	86.89	14.37
Distillation Signal *(Base: 4-step)*	Prediction-only (Generic)	0.52	0.78	98.45	19.82
	Prediction + Route	0.52	0.78	92.10	17.50
	Prediction + Control-path	0.52	0.78	90.50	16.80
	Pred + Route + Ctrl (Action-aware)	0.52	0.78	87.50	15.60
Spatial Execution *(Base: Action-aware)*	Full update for all tokens	0.52	0.78	87.50	15.60
	Full/Reuse only $(0.35, 0.00, 0.65)$	0.38	0.57	91.20	17.90
	Full/Light only $(0.20, 0.80, 0.00)$	0.46	0.69	88.00	16.10
	Full/Light/Reuse $(0.20, 0.30, 0.50)$	0.42	0.63	88.65	16.55
Temporal Cache *(Base: Spatial)*	Cache disabled $(1, 1, 1)$	0.42	0.63	88.65	16.55
	Conservative $(1, 2, 2)$	0.39	0.59	89.10	17.05
	Default intervals $(1, 2, 4)$	0.37	0.55	89.87	17.67
	Aggressive reuse $(1, 4, 8)$	0.33	0.50	92.40	19.30

Overall Interpretation. The baseline table separates three effects that are conflated in the final KVLR-fast number: few-step acceleration, action-aware preservation of the teacher's control policy, and routing-derived adaptive execution. Generic few-step distillation (DMD2) [56] is computationally effective but action-destructive: it gives an $8.92\times$ latency speedup, yet retains only 88.3% of the teacher's CD-based control accuracy and increases FID by 5.45. The action-aware student keeps exactly the same measured cost as DMD2 but recovers the control retention to 99.3% and reduces the FID increase to $+1.23$. Spatial adaptive execution and temporal caching then provide additional compute reductions, yielding the final $12.54\times$ latency speedup while retaining 96.7% of teacher control accuracy.

Expanded Component Validations. To further isolate the internal dynamics of the efficient student and the adaptive execution policy, we introduce an expanded ablation matrix. Tab. 16 summarizes these diagnostic settings, evaluating the independent contributions of the distillation signals, spatial execution policies, and temporal refresh rates.

Analysis of Distillation Signals. The distillation ablation isolates the value of the structured control pathway. While prediction-only distillation (DMD2) [56] severely degrades action faithfulness (CD 98.45), adding routing alignment explicitly informs the student which physical modalities are active, dropping CD to 92.10. However, preserving the intermediate action-control features (Control-path) provides an even stronger independent signal (CD 90.50). The combination of both yields the full action-aware student (CD 87.50), proving that route and control-path supervision are synergistic and transfer the teacher's action policy at zero additional inference cost.

Analysis of Spatial Execution. The spatial execution ablation tests whether the scaled "light" branch is more effective than hard token suppression. A strict Full/Reuse policy (allocating 35% of tokens to full updates and aggressively reusing the rest to match the target FLOPs) drops latency to 0.38 s but penalizes quality heavily (CD 91.20), as critical edge cases are suppressed. Introducing the "light" branch (Full/Light/Reuse at $20/30/50$) allows the model to maintain broad contextual awareness with cheaper cached-routing updates, yielding a strictly better Pareto frontier (Latency 0.42 s, CD 88.65).

Analysis of Temporal Cache. The temporal cache ablation exposes the deployment trade-offs of feature caching. Disabling the cache entirely relies strictly on spatial token sparsification (0.42 s). Enabling conservative caching $(1, 2, 2)$ yields "free" latency improvements (0.39 s) with negligible CD degradation ($+0.45$). Our default $(1, 2, 4)$ setting pushes this efficiency further. However, aggressive cache reuse $(1, 4, 8)$ introduces noticeable temporal drift; as the instrument moves out of the cached bounding regions, the student fails to update the active kinematics, resulting in a sharp penalty to both CD (92.40) and visual fidelity (FID 19.30).

F.5 More Qualitative Results.

Qualitative results on architecture design. Fig. 8 provides additional qualitative comparisons for the architecture ablation. The text-only baseline fails to maintain a clear instrument appearance and exhibits inaccurate tool motion, especially in the later frames. Directly injecting raw action vectors yields only limited improvement: although the generated motion is partially guided by the input

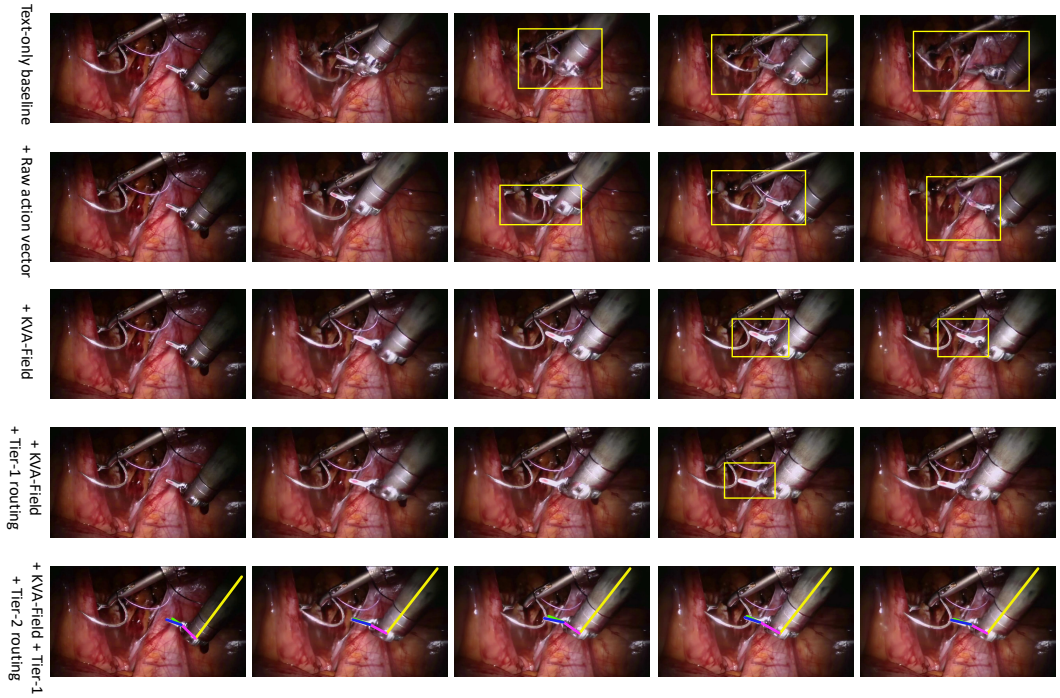

Figure 8: Additional qualitative comparisons on architecture design. We compare representative variants from the architecture ablation, including text-only generation, direct raw-action conditioning, KVA-Field conditioning, Tier-1 routing, and the full hierarchical routing model. The full model produces clearer tool appearance and more action-consistent temporal evolution.

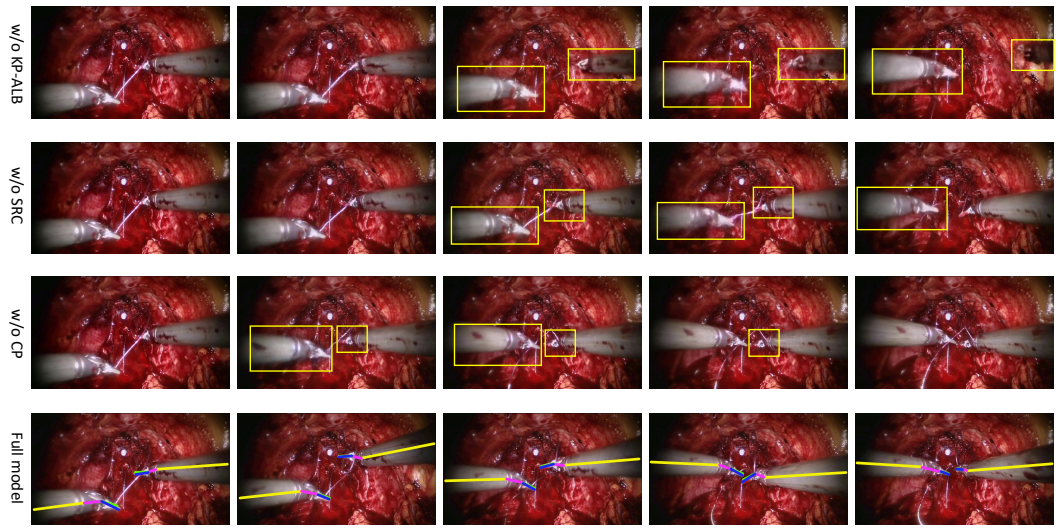

Figure 9: Additional qualitative comparisons on kinematic-prior losses. We compare the full model with variants that remove KP-ALB, SRC, or CP. The proposed kinematic-prior losses improve spatial alignment and temporal consistency, leading to more stable action-conditioned generation.

actions, the instrument trajectory remains imprecise, and the visual quality degrades over time. This supports our observation that low-dimensional articulated kinematics are difficult to use directly for dense image-space video evolution. Replacing raw actions with the proposed KVA-Field substantially improves the spatial alignment between action inputs and visual generation, leading to clearer tool structures and more consistent motion. Introducing Tier-1 routing further improves the use of different physical control modalities, while the full model with both Tier-1 and Tier-2 routing achieves the best qualitative results by adapting control to both modality types and motion scales. These qualitative

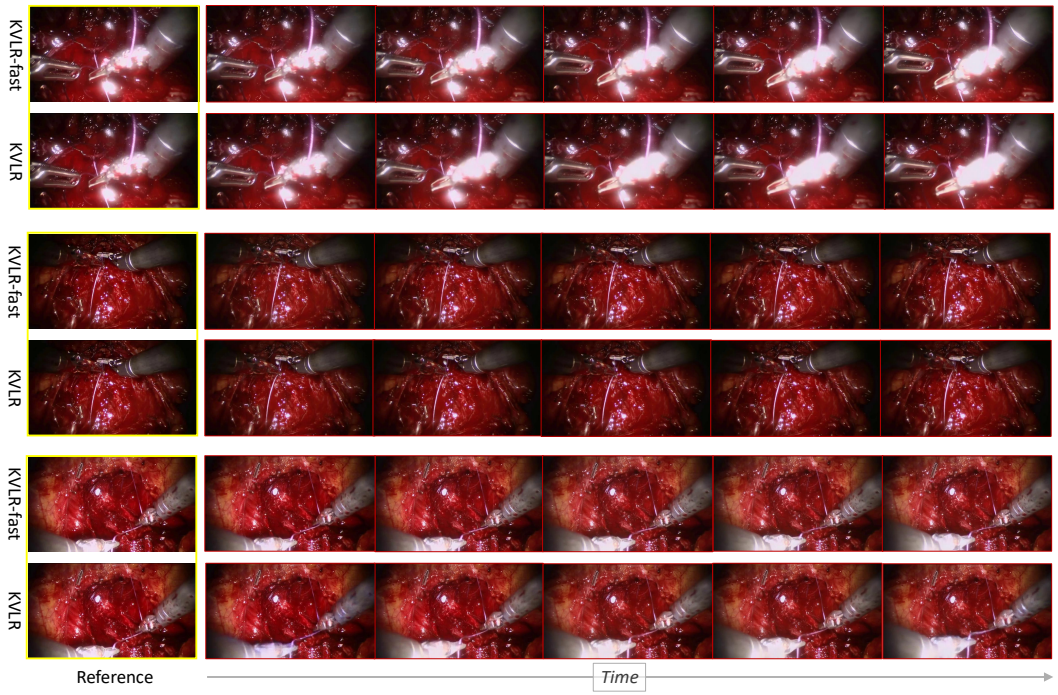

Figure 10: Additional generation results of KVLR and KVLR-fast. We show more examples across diverse surgical scenes and actions. Both the full model and the efficient variant preserve action-consistent visual evolution, while KVLR generally yields slightly sharper appearance details and stronger temporal stability under complex motion. These examples further show that the action-adaptive efficient model retains most of the control fidelity of the full generator despite using substantially reduced computation.

trends are consistent with the quantitative architecture ablation in Tab. 6, showing that action lifting and hierarchical routing are jointly important for action-controlled surgical video generation.

Qualitative results on kinematic-prior losses. Fig. 9 shows additional qualitative comparisons for the kinematic-prior loss ablation. Replacing KP-ALB with uniform load balancing [43, 10, 60] weakens the correspondence between routing decisions and physical action cues, resulting in degraded spatial alignment and blurrier tool appearance. Removing SRC leads to visibly less stable temporal evolution, with tool structures and interaction regions becoming inconsistent in the later frames. This suggests that temporal regularization of routing decisions is important for maintaining coherent action-conditioned generation across video frames. Removing CP also degrades the results, indicating that predictable routing helps stabilize expert selection and supports reliable control during generation. Overall, the full model produces the most stable and visually faithful results, confirming that the three kinematic-prior losses are complementary and collectively improve the quality of routed visual control.

Additional qualitative results. We provide additional qualitative results in Figs. 10 and 11. These visualizations cover a wider range of scenes, action trajectories, and procedural stages across *Knotting*, *NeedleGrasping*, and *NeedlePuncture*, and further illustrate two key properties of the proposed framework. As shown in Fig. 10, both KVLR and KVLR-fast preserve action-consistent scene evolution under diverse articulated controls, including large tool transport, fine local adjustment, and contact-heavy interaction. Moreover, as illustrated in Figs. 11 , the generated videos remain visually coherent across challenging conditions such as appearance variation, illumination change, cluttered backgrounds, and different instrument poses. Together with the main qualitative comparisons, these results further support that the proposed structured kinematic-to-visual control improves both controllability and visual stability across heterogeneous surgical dynamics.

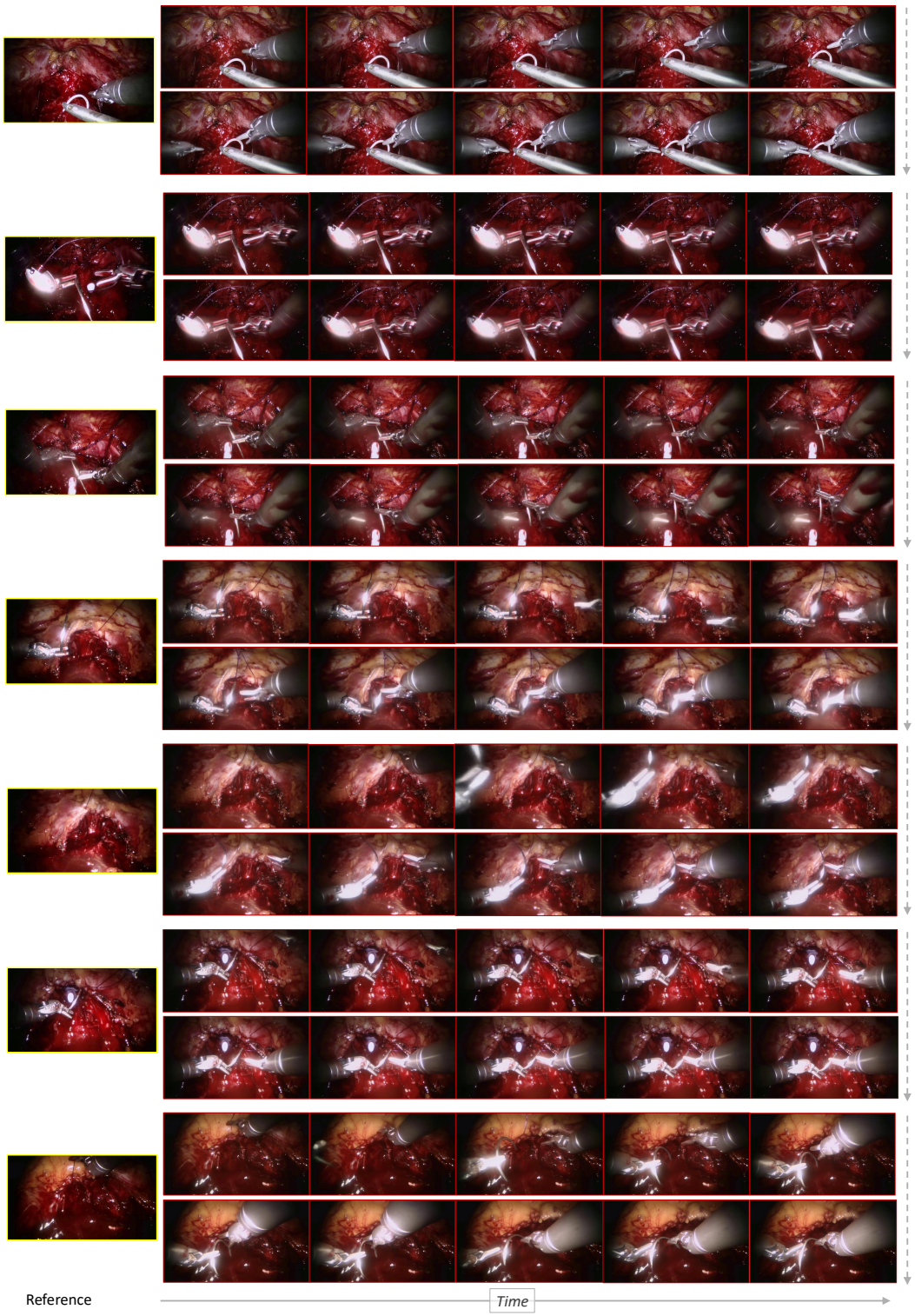

Figure 11: Additional diverse generation results of KVLR on different surgical actions. The generated videos remain consistent with the prescribed articulated controls across different motion patterns, including fine manipulation, larger transport motion, and varying tool–tissue interaction stages. These results highlight the robustness of the proposed structured control interface across multiple action categories and scene configurations.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main claims—structured kinematic-to-visual control, hierarchical routing, efficient generation, and the KASA benchmark.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: [Yes]

   Guidelines: The Appendix C includes a dedicated discussion of limitations and broader impact. It explicitly scopes the method to articulated robotic surgical video generation with curated 7-DoF supervision and notes that broader procedures and instruments may require additional adaptation.

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The paper's theoretical results are provided in Appendix A, and for these results, the assumptions and proofs are explicitly stated.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper specifies the problem setup, model components, training/evaluation protocol, metrics, dataset construction pipeline, and ablations in Secs. 2–4 and the appendix, which together provide the information needed to reproduce the reported main results.

Guidelines:

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

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We provide anonymized access to the code and supporting artifacts used in this work for peer review. The core codebase is included in the supplementary material, while larger checkpoints and data access instructions are shared through anonymized external links that preserve double-blind reviewing. Any public de-anonymized release will follow the licenses and usage restrictions of the underlying assets.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the main training and evaluation details needed to understand the results, including the data split, optimizer, learning rate, weight decay, batch size, default resolution, number of denoising steps, baseline setup, and evaluation metrics in Sec. 4.1, with additional implementation details deferred to the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: The paper reports additional statistical information in the experimental section and Appendix F, including uncertainty estimates, definitions of the averaging protocol, and clarification of how the reported confidence intervals or summary statistics are computed for the supporting analyses.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: The paper describes the compute resources in the experimental section and Appendix D, including the hardware used for training and evaluation, as well as additional resource details such as the relevant worker configuration and runtime information needed for reproduction.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research described in this submission conforms with the NeurIPS Code of Ethics. The paper preserves anonymity and uses standard research and evaluation practices for academic study.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The Appendix C includes a broader-impact discussion covering both potential positive impacts.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [No]

    Justification: The paper does not describe concrete safeguards for releasing data or models, such as gated access, usage restrictions, or release-time filtering.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The Appendix C includes an asset-license note that credits the external datasets, models, and software components used in this work and explicitly states their license or usage conditions, together with the commitment to respect the corresponding terms of use and release constraints where applicable.

Guidelines:

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

13. **New assets**

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

Answer: [No]

Justification: Although this work introduces new derived benchmark annotations and model artifacts for evaluation, we do not claim a formal public release of new assets in the current submission. Materials shared for peer review are anonymized and provided only for evaluation purposes. Any future release will be subject to the licenses and usage constraints of the underlying resources.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not report crowdsourcing experiments or human-subject studies involving participants who were instructed, compensated, or evaluated. The human-in-the-loop annotation used for dataset curation is not presented as a crowdsourcing study in the submission.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not report crowdsourcing studies or human-subject experiments aimed at collecting responses from participants. The human-in-the-loop annotation used for dataset curation is part of internal data annotation rather than a crowdsourcing or behavioral study.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core methodology in this research does not use LLMs as an important, original, or non-standard component, so declaration is not required under the checklist policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.