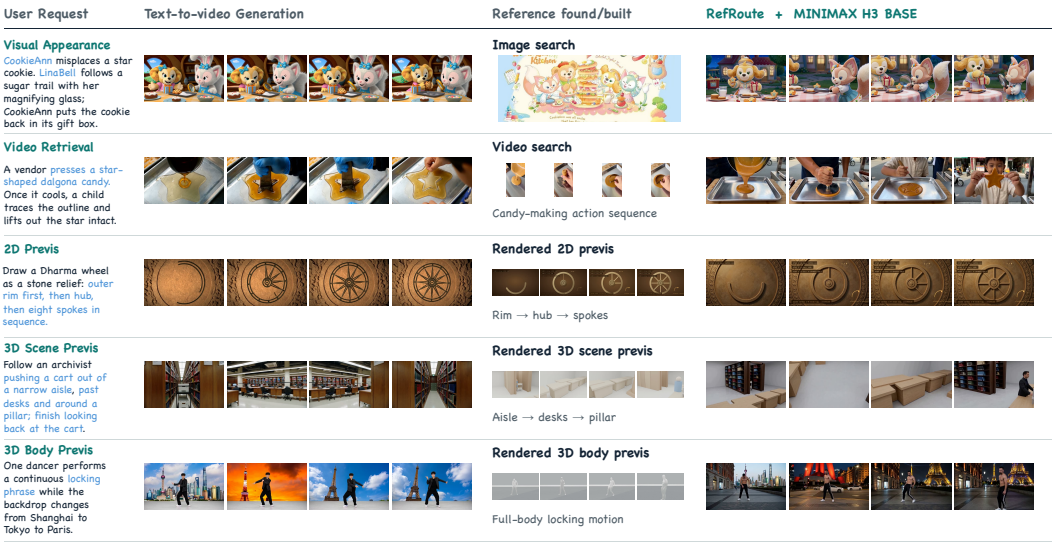
RefRoute: Learning to Search, Construct, and Verify References for Video Generation

Anonymous authors
Paper under double-blind review



Abstract

Video generators can condition on images, video clips, and audios, yet satisfying a video generation request often depends on choosing the right information and control signal before generation. We present RefRoute, a generator-agnostic agent that learns to search for external evidence or construct multimodal references, and then assigns each asset a role in a generator-ready request. Its reference routes cover retrieved media, 2D event timing, 3D scene and camera path, reconstructed body motion, and video depth. We train a reference-routing policy from 12k teacher trajectories; we also specify two group-relative reinforcement-learning stages for route and video feedback. RefRoute-Bench evaluates 600 requests across five request categories for different knowledge gaps and reference routes. In the evaluation, RefRoute with MiniMax H3 Base show apparent improvements compared with the generator alone, with the largest gain on video retrieval.

1 Introduction

Video generation has progressed from text-conditioned synthesis to systems that can use images, source videos, and more structured controls (Blattmann et al., 2023; Yang et al., 2024; Kong et al., 2024; Wan et al., 2025; Seedance Team et al., 2026; MiniMax AI, 2026). This progress makes compelling video easier to produce, but visual plausibility does not guarantee that a generated video follows the user's request. A familiar character may lose its defining appearance; a craft demonstration may omit a necessary step; or a camera move may break the spatial relationship between objects. Such failures arise when the generator must infer identity, procedural order, motion, and geometry from a short prompt, even when these constraints could be made explicit by an external reference. As open video generators become more accessible, a reusable policy for finding and shaping those

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

references becomes practical across different generation backends (Peng et al., 2025; Wan et al., 2025; MiniMax AI, 2026).

The upstream decision is more involved than simply adding a retrieved image. An appearance gap may call for a visual search result; an unfamiliar action may require an inspected source clip; exact event timing or a camera path may be better expressed through rendered previsualization (previs). Some media should condition the generator directly, while other evidence is useful only for writing an accurate request. Search-augmented visual agents learn to gather evidence, and recent video agents learn to coordinate generation tools (Feng et al., 2026a; Li et al., 2026a). Separately, executable or editable scene representations make temporal and spatial plans inspectable (Li et al., 2026b; Yin et al., 2026). The missing decision is how to choose among retrieved and constructed references for each request, verify what they actually contain, and bind them to an appropriate role in the final generation request.

We introduce **RefRoute**, a reference-routing policy for video generation. Given a user request and optional media, it plans what information or control is missing, searches web, image, or video sources when needed, and can construct 2D, 3D scene, 3D body-motion, or depth-video references. It inspects candidate media and rendered previews before deciding which assets to attach, which to use as evidence, and which to omit. Figure 1 summarizes this process. RefRoute outputs assets with explicit roles and a generator-ready request; a separate compiler maps them to the interface of a chosen video generator. The policy therefore learns decisions about reference modality and use, rather than the request syntax or parameters of a single generator.

To learn these decisions, we assemble 12k candidate requests spanning visual evidence, video retrieval, and constructed 2D, 3D scene, and 3D body references. Screened teacher trajectories provide decision-level supervised fine-tuning (SFT) targets. We also specify a two-stage group relative policy optimization (GRPO) procedure: the first stage rewards executed reference routes, while the second holds the earlier route fixed and scores the final request together with gated video feedback. We introduce **RefRoute-Bench**, a 500-request benchmark with five categories and separate trajectory and generated-video evaluations.

The completed SFT policy, paired with MiniMax H3 Base, reaches a diagnostic micro-average of 60.96 on RefRoute-Bench, compared with 56.86 for the same generator used without RefRoute (Table 2).

2 Related Work

Controllable video generation. The rapid evolution of video generation has enabled synthesis of temporally coherent and semantically rich video sequences. (Ho et al., 2022; Blattmann et al., 2023; Yang et al., 2024; Kong et al., 2024; Peng et al., 2025; Ma et al., 2025; Wan et al., 2025). These advances have fulfilled the basic requirements of synthesized videos, but generating specific entities, complex precedures, realistic motion dynamics, while keeping temperal and spatial consistency still remain challenging. This gap shifts attention toward the information and control supplied to the video generator.

Video generation interfaces have expanded from text and keyframes toward combinations of images, videos, and editable conditions, as illustrated by HappyHorse 1.1, Seedance 2.0, and MiniMax H3 (Alibaba Cloud, 2026; Seedance Team et al., 2026; MiniMax AI, 2026). Recent works also increasingly externalizes temperal or spatial decisions, like camera paths(Li et al., 2026c), 3D scene factors(Cai et al., 2026), and physical behaviors(Li et al., 2026b; Feng et al., 2026b; Bai et al., 2025). This progression makes the choice of *which* reference to supply an important upstream problem.

Agentic visual generation. Visual generation agents have begun to retrieve evidence, filter unhelpful results, route requests to different workflows(Feng et al., 2026a; Chen et al., 2026b; Wang et al., 2026; Chen et al., 2026a;c; Li et al., 2026a). They use multimodal search to fill knowledge and visual gaps before visual synthesis, and learning outcomes of tool trajectories to improve agent's decisions. These works establish that evidence acquisition and workflow selection can be learned. RefRoute shares that premise, but its action space extends to references that carry temporal order, motion, or 3D geometry.

Previsualization before video generation. Recent video systems use executable simulation or editable 3D states to support generation(Li et al., 2026b; Yin et al., 2026; Li et al., 2026a). For example, VideoCoCo compiles a prompt into executable Blender code, renders a physically specified temporal draft(Li et al., 2026b). StateFlow maintains an editable 3D world state for iterative previsualization(Yin et al., 2026). Both show why an inspectable intermediate representation can be more useful than a verbal plan. RefRoute focuses on the decision connecting the two: whether to retrieve an existing reference or construct a 2D or 3D previsualization. It then binds the chosen assets to explicit roles in a generator-ready request. This reference-route decision, rather than a particular generator or previsualization technique, is the target of our training and evaluation.

3 Method

3.1 Reference-routing agent

Given a user request x and optional user-provided media A_0, RefRoute plans to obtain the information and visual conditions needed for video generation. Rather than generating a video within the agent workflow, it produces a generator-ready request. Let A be the catalog of accepted assets, which may be user-provided, retrieved, or constructed, and let u be the final video generation request which includes each asset's intended role. A compiler for video generator g produces request q_g:

$$q_g = C_g(x, A, u), \qquad v \sim G_g(\cdot \mid q_g), \tag{1}$$

where G_g generates video v outside of the request generation pipeline. The reference policy operates on assets and roles rather than a particular generator's request schema. Our implementation instantiates C_g for MiniMax H3 Base, which is an open-source variant of MiniMax H3, a general-purpose, omni-modal video generative model (MiniMax AI, 2026).

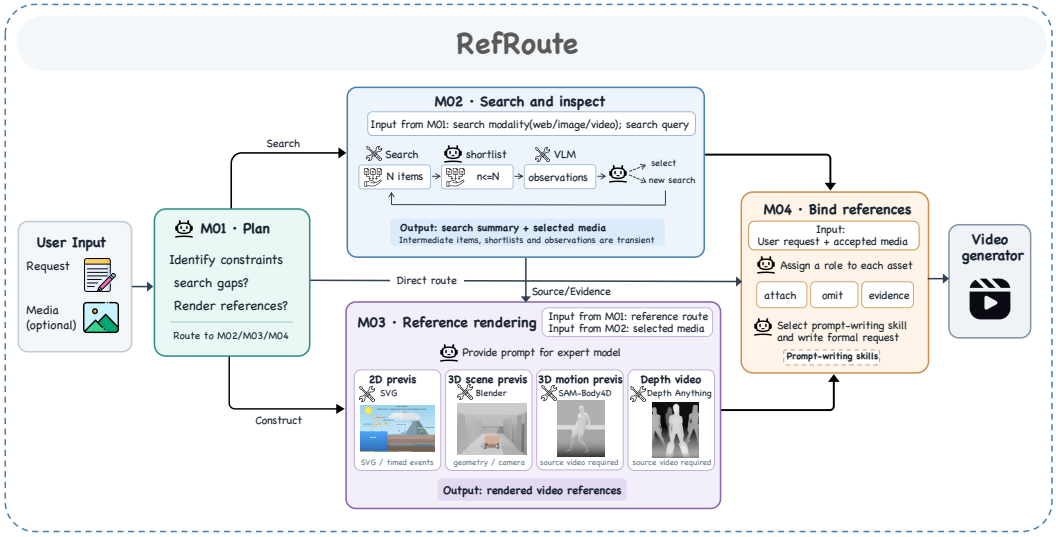


Figure 1: **Overview of RefRoute workflow**. Module 01 (M01) proposes initial plan by identifying knowledge gaps for search (M02), or recommending types of references for rendering (M03). M02 executes searching with provided search engines. The agent firstly shortlists several promising items from all the returned ones, then it learns the content of the media via an external VLM, and finally selects media and writes a summary for that round of search. M03 renders and reviews one of four reference types. For the 2D and 3D routes, the agent briefs an expert model that builds an SVG or Blender reference; the depth route transforms a source clip. M04 assigns asset roles and compiles a request. Video generation occurs outside the agent.

Figure 1 shows the four-stage workflow of RefRoute. In **M01 (planning)**, the policy parses the request into constraints and identifies any search gaps or need for a rendered reference. It may

route directly to M02, M03, or M04. Thus, a fully specified request which focuses on timing and ordered events or temperal-spatial consistency may proceed to reference construction without search, whereas a request with unknown definitions can route to search before constructing the reference. The route is a reasoned decision made from the user request.

In **M02 (search)**, M01 supplies a query and one of three search modalities: web, image, or video. The policy shortlists search items, obtains selected media, and receives observations from an external vision-language model (VLM). It decides which media to adopt and returns a compact search summary. Only adopted files enter A; intermediate result lists, rejected downloads, and raw observations will not be stored in the conversation history and passed to the downstream stages. M02 can then route to M03 if a retrieved source must be transformed, or to M04 if the accepted media already provide suitable conditions.

In **M03 (reference rendering)**, the policy chooses one of four routes, supplies a task brief and relevant media when needed. 2D previs encodes precise event timing, screen-space positions in the 2D space. An expert model, usually a large VLM, authors an animated SVG, which is rendered frame by frame into a video reference. 3D scene previs encodes spatial layout, occlusion, object paths, and a camera trajectory that must remain coherent as the plot develops and viewpoint moves. An expert builds a Blender (Blender Foundation, 2026) scene with simple proxy geometry and renders its camera path. 3D body previs begins with an inspected source clip that clearly shows a difficult action under a stable camera. SAM-Body4D (Gao et al., 2025) reconstructs the selected person's motion; an expert then places the animated body and designs a new camera path in Blender. In this way, the human motion and camera path can be decoupled, thus allowing the final video to change the performer and or camera setting. The fourth route uses Video Depth Anything (Chen et al., 2025) to convert an inspected source clip into a depth-video reference when its broad geometry and motion should remain but details like appearance and lighting should not. Each route yields a concrete video reference rather than only a textual plan. An independent VLM observes the rendered reference, and the policy can revise an expert-built reference before selecting a final version for A.

Finally, **M04 (request writing)** receives x and the accepted media from earlier stages. For each asset, the policy chooses to *attach* it as a generator condition, use it as *evidence* when writing the prompt, or *omit* it. Task-specific prompt-writing skills are provided to assist with the final request. The compiler checks asset identity and compatibility with the chosen generator mode before emitting q_g. Whether the generated video follows the selected references is measured separately.

3.2 Data construction

3.2.1 User requests.

We assemble a pool of 12k video requests to expose different needs for external evidence and constructed references. Table 1 classifies the initial user requests by search modalities and reference routes. We first build a large pool of long-tail candidates grounded in evidence using a search agent that collects items in batches from verified API endpoints and Wikipedia. The candidates include recognizable appearances, professional procedures, and human motions such as street dance and martial arts. We then use Gemini 3.1 Pro (Google DeepMind, 2026) to write natural video-generation requests from batches of candidates. The resulting requests are designed with predefined search gaps or reference routes.

For example, changing the driving tone of a Chladni plate should reorganize its sand; a ship's lamp must spell `SHIP` with correct Morse timing; and a video featuring LinaBell and CookieAnn should preserve their recognizable appearances. A request with fully specified timing may need a 2D reference without search, whereas one involving unfamiliar visual or procedural details may require search before reference construction.

Category	Subcategory	Search	Default workflow	$Number$
Text/Image gap	-	Text/image	M02 (web/image) → M04	4000
Video gap	Procedural knowledge	Video	M02 (video) → M04	1500
	Style learning	Video	M02 (video) → M04	1000
	Direct reference	Video	M02 (video) → M04	1000
	Depth reference	Video	M02 (video) → M03 (depth) → M04	500
2D previs	Search-grounded	Text/image	M02 → M03 (2D) → M04	2000
	Prompt-specified	-	M03 (2D) → M04	1000
3D scene previs	Prompt-specified	-	M03 (scene) → M04	1000
	Search-grounded	Text/image	M02 → M03 (scene) → M04	500
3D body previs	Body motion	Video	M02 (video) → M03 (body) → M04	500
Total				12000

Table 1: Categories of training requests by search modalities and reference routes.

3.2.2 Teacher trajectories.

A teacher policy processes the requests through the M01–M04 tool interface. The agent system prompt and stage-specific decision templates are reproduced in Appendix B. We use Seed 2.0 Pro as the teacher agent and record the complete interactions including every policy decision and tool observation for replay. To compress the downstream context, M02 omits search-result lists and VLM observations while carrying forward accepted media and a short summary of each search round. Trajectory screening checks action validity, actual tool execution, asset identity and availability, and whether the terminal request can be compiled. This yields action-level supervision from the teacher trajectories. For supervised fine-tuning, the student conditions on the request and preceding interaction history, valid teacher decisions are prediction targets. Tool observations and fixed context are available to the student but are masked from the training loss.

3.2.3 Benchmark and evaluation.

We construct **RefRoute-Bench** from 500 video requests covering the five request categories.

Visual evaluation. We have manually retrieved the evidences for all the search gaps and prepared evaluation checkpoints and rubrics for each request in RefRoute-Bench. Given the generated video v_i, Gemini 3.1mPro (Google DeepMind, 2026) sees the original request, resolved search gap, the rubric. It inspects the complete video and assigns each checkpoint an integer from 0 to 5. We scale the scores between 0 to 100 and report the mean for each category and over the 500 cases.

Trajectory evaluation. For each RefRoute-Bench request, we also collect a teacher trajectory, then manually review each decision against the original request and the evidence available at that point to form a gold trajectory. To evaluate a trajectory, deterministic checks assess action format, tool choice, asset IDs, and route fields; a blind, stage-specific judge scores semantic correctness while allowing valid actions that differ from the gold wording. We report per-stage scores, a stage-balanced mean, and the length of the continuously correct prefix. Trajectory scores are separate from final-video visual scores.

3.3 Training the reference-routing policy

Figure 2 separates supervision of the full reference route from feedback on the final generated video. The first stage uses teacher actions to establish the tool-use policy; the two subsequent group-relative stages are the specified reinforcement-learning procedure.

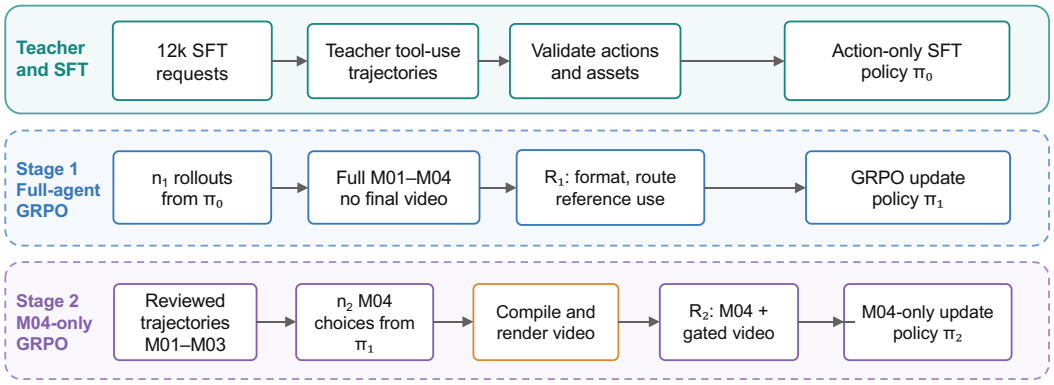


Figure 2: Training pipeline. Solid outlines denote the decision-level supervised fine-tuning (SFT) stage; dashed outlines denote the specified group relative policy optimization (GRPO) stages. Stage 1 evaluates the executed M01–M04 route without generating a final video. Stage 2 holds a reviewed M01–M03 history fixed and updates M04 using M04 request correctness and gated final-video feedback.

Decision-level supervised fine-tuning. We initialize the main policy π_θ from Qwen3-8B (Qwen Team, 2025) and adapt it with LoRA (Hu et al., 2021). For each retained teacher decision, h_t contains the request and preceding policy actions and tool observations, and a_t^* is the next valid agent action. With $m_{t,j} = 1$ only for supervised agent tokens and $w_{s(t)}$ the weight of decision stage $s(t)$, the action-only objective is

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{Z} \sum_{(h_t, a_t^*) \in \mathcal{D}_{\mathrm{SFT}}} w_{s(t)} \sum_{j=1}^{|a_t^*|} m_{t,j} \log \pi_\theta(a_{t,j}^* \mid h_t, a_{t,<j}^*), \quad Z = \sum_{t,j} w_{s(t)} m_{t,j}. \tag{2}$$

Preceding tool responses and action in h_t all have zero target mask. The uniform SFT run sets $w_{s(t)} = 1$; stage weights, when used, are declared per checkpoint.

Two-stage GRPO. Starting from the SFT policy, we specify two stages of group relative policy optimization (GRPO) (Shao et al., 2024; Li et al., 2026a). For each training request, four independent on-policy rollouts form a group. Stage 1 executes the complete M01–M04 workflow with its real search and reference tools, but does not submit the final request to the video generator. Its reward combines calibrated components for action format, agreement of the *executed* route with a reviewed request-level specification, and correct use of retained or constructed assets:

$$R_i^{(1)} = \lambda_f r_{\mathrm{format}}(\tau_i) + \lambda_r r_{\mathrm{route}}(\tau_i, x) + \lambda_u r_{\mathrm{use}}(\tau_i, x), \qquad \lambda_f + \lambda_r + \lambda_u = 1. \tag{3}$$

The components are recorded separately, and their definitions and weights are frozen before sampling. A proposed M01 route alone cannot earn route credit unless the corresponding actions were executed.

Stage 2 begins from a reviewed, fixed history h^* through M03. For each sampled M04 continuation u_i, the compiler binds its accepted assets A^* to a generator request, and the generator produces a video. We score the semantic correctness of M04 against the original request and available references with $r_{\mathrm{M04}} \in [0, 1]$. A structural gate $g_i \in \{0, 1\}$ checks whether the request is usable and its reference bindings are valid; semantic omissions lower r_{M04} or the video score, rather than the gate. A fixed, request-specific rubric $\mathcal{J}_x \in [0, 1]$ evaluates the video:

$$\begin{aligned} q_i &= C_g(x, A^*, u_i), \qquad v_i \sim G_g(\cdot \mid q_i), \\ R_i^{(2)} &= \lambda\, r_{\mathrm{M04}}(u_i, x, A^*) + (1 - \lambda)\, g_i\, \mathcal{J}_x(v_i). \end{aligned} \tag{4}$$

Here $\lambda \in [0, 1]$ controls the balance between M04 correctness and gated video quality. An invalid policy action or missing final request receives zero visual reward; external service failures are excluded from the policy update rather than scored as policy errors.

For either stage $s \in \{1, 2\}$, let $\bar{R}_x^{(s)}$ and $\sigma_x^{(s)}$ be the mean and population standard deviation of the four scored candidates for request x. We assign each agent token in candidate i the group advantage and importance ratio

$$\hat{A}_{xi}^{(s)} = \frac{R_{xi}^{(s)} - \bar{R}_x^{(s)}}{\sigma_x^{(s)} + \epsilon}, \qquad \rho_{xik}(\theta) = \frac{\pi_\theta(z_{xik} \mid c_{xik})}{\pi_{\text{old}}(z_{xik} \mid c_{xik})}, \tag{5}$$

where c_{xik} is the preceding interaction context. With $\mathcal{P}_{xi}^{(s)}$ the generated policy-token positions and $\mathcal{B}_s$ the scored rollouts in complete groups, the clipped objective is

$$\mathcal{L}_{\text{GRPO}}^{(s)}(\theta) = -\frac{1}{|\mathcal{B}_s|} \sum_{(x,i)\in\mathcal{B}_s} \frac{1}{|\mathcal{P}_{xi}^{(s)}|} \sum_{k\in\mathcal{P}_{xi}^{(s)}} \min\Big(\rho_{xik}(\theta)\hat{A}_{xi}^{(s)}, \\ \text{clip}(\rho_{xik}(\theta), 1-\epsilon_\ell, 1+\epsilon_h)\hat{A}_{xi}^{(s)}\Big). \tag{6}$$

We use asymmetric clipping $(\epsilon_\ell, \epsilon_h) = (0.2, 0.28)$. Incomplete groups and groups with zero reward variance provide no update. Stage 1 updates all emitted agent actions; Stage 2 updates only M04 actions.

Algorithm 1 summarizes the two-stage procedure..

Algorithm 1 Reference-route policy training (GRPO stages specified)

Require: screened teacher decisions $\mathcal{D}_{\text{SFT}}$; reviewed RL requests $\mathcal{D}_{\text{RL}}$
$\pi_0 \leftarrow \text{SFT}(\pi_{\text{base}}, \mathcal{D}_{\text{SFT}})$ using Eq. (2)
for $s = 1, 2$ **do**
  for each $x \in \mathcal{D}_{\text{RL}}$ **do**
    if $s = 1$ **then**
      Sample four full M01–M04 tool trajectories; score with Eq. (3)
    else
      Fix reviewed M01–M03 history; sample four M04 actions
      Compile, render valid requests, and score with Eq. (4)
    end if
    Exclude external failures; assign zero to policy-invalid actions
    Compute group advantages using Eq. (5)
  end for
  Update only emitted policy tokens using Eq. (6); obtain π_s
end for

4 Experiments

4.1 Implementation details

The RefRoute main policy is Qwen3-8B (Qwen Team, 2025). Teacher trajectories use Seed-2.0-pro (ByteDance Seed, 2026) as the main policy. We also use Seed-2.0-pro as the independent VLM, with reasoning disabled. The 2D and 3D previs experts use the DeepSeek-V4-Flash endpoint (DeepSeek, 2026). The final-video judge uses Gemini 3.1 Pro (Google DeepMind, 2026).

For supervised fine-tuning (SFT), we adapt Qwen3-8B with LoRA rank 8, scaling 16, on the query, key, value, and output attention projections. We use bfloat16, batch size of 32, a cosine learning-rate schedule, and 3% warmup. The completed base run uses 900 steps at 2×10^{-5} on eight NVIDIA H100 80GB GPUs. From this checkpoint, an M01-only refinement runs for 900 steps at 5×10^{-6}; a subsequent M01/M04 refinement runs for 500 steps at 10^{-6}, each on four H100 80GB GPUs. These are distinct training runs rather than a single uninterrupted schedule.

The specified group relative policy optimization (GRPO) setup uses 2k training requests and four on-policy rollouts per request in each stage, with learning rate 10^{-6} and asymmetric clipping bounds $[0.8, 1.28]$. For the Stage-2 reward in Eq. equation 4, we set $\lambda = 0.2$.

4.2 Main results

Table 2 compares five text-to-video generation models with the RefRoute workflow on RefRoute-Bench. RefRoute has the highest mean in all five categories and the overall average score.

Model	Text/Image	Video	2D previs	3D scene	3D body	Overall
CogVideoX-5B (Yang et al., 2024)	37.72	44.95	41.69	43.57	52.76	43.36
Mochi-1 (Genmo Team, 2024)	33.36	46.50	42.05	49.75	53.34	44.32
HunyuanVideo-13B (Kong et al., 2024)	20.99	32.35	34.22	34.55	43.26	31.98
Wan2.1-T2V-14B (Wan et al., 2025)	30.07	46.95	46.32	46.39	48.78	43.52
MiniMax H3 Base (MiniMax AI, 2026)	51.10	58.50	52.83	61.54	62.15	56.86
RefRoute + MiniMax H3 Base	**55.22**	**65.61**	**56.08**	**62.97**	**64.21**	**60.96**

Table 2: RefRoute-Bench visual evaluation (0–100; higher is better).

4.3 Analysis and case studies

Across the five open-source text-to-video generators, MiniMax H3 Base scores highest in every request category. The other generators differ by task: Wan2.1 has the strongest Video and 2D scores among the four non-H3 baselines (46.95 and 46.32), whereas Mochi-1 leads them on 3D scene and 3D body (49.75 and 53.34). HunyuanVideo scores lowest in every category. RefRoute improves on its H3 generator most in Video (+7.11 points), followed by Visual (+4.12) and 2D (+3.25). Thus, under current evaluation protocal, the generated videos earn the largest gain by using reference routing for information and ordered-action requests among all request categories.

The selected same-request video comparisons illustrate these differences. For the Olmec-head museum story (case 020; Figure 3), RefRoute scores 76.0 versus 53.3 for the best T2V generator. In the dalgona example (Figure 4), the selected RefRoute video covers pouring, pressing, tracing, and lifting the star (80.0 versus 40.0), illustrating the value of a procedural video reference. The Dharma-wheel drawing (Figure 5) and archive-cart camera move (Figure 6) score 80.0 each, versus 67.5 and 47.7 for H3, respectively. These examples test construction order and viewpoint continuity, which a prompt alone must specify without a rendered reference. For the 3D body route, the locking world-tour case (Figure 7) also favors RefRoute (56.0 versus 50.7).

4.4 Ablation study

Table 3 separates policy diagnostics from generated-video scores. The system-prompt baseline rarely completes the tool workflow: only one of its 100 rollouts reaches a video-generator request, compared with 84 after SFT. This makes a direct video comparison unavailable for that baseline; its displayed video values are the vanilla T2V fallback, not scores of videos conditioned on its proposed references.

	Trajectory evaluation (%)				Video evaluation (0–100)					
Policy	Gap	Search	Valid	Ready	Visual	Video	2D	3D scene	3D body	Micro
Vanilla T2V (MiniMax H3 Base)	–	–	–	–	51.10	58.50	52.83	61.54	62.15	56.86
Qwen3-8B + system prompt	50	80	45.3	1	51.10†	58.50†	52.83†	61.54†	62.15†	56.86†
RefRoute-SFT	78	93	88.2	84	55.22	65.61	56.08	62.97	64.21	60.96
RefRoute-RL-stage 1	85	94	90.5	89	56.76	67.27	62.99	63.97	64.75	63.40
RefRoute-RL-stage 2	87	94	91.0	90	58.08	72.45	65.25	64.64	64.98	65.83

Table 3: RefRoute-Bench ablation studies. Gap and Search are M01 matches to the approved gap count and search-need label; Valid is the fraction of validated main-policy calls; Ready is the fraction of requests reaching a video-generator request. These partial checks are not the gold-trajectory scores defined in Section 3.2. †Vanilla T2V fallback values: the sole ready Qwen3-8B request has not been scored as a generated video. A dash denotes an unavailable measurement. The SFT video row uses the same mixed-tier snapshot and caveats as Table 2.

Relative to vanilla T2V, SFT raises the overall video score by 4.10 points; Stage 1 and Stage 2 reach 63.40 and 65.83. Stage 2 adds 2.43 points over Stage 1, with the largest category gain in video retrieval (67.27 to 72.45). Gap and Ready rise from 78 and 84 after SFT to 87 and 90 after Stage 2. The smaller gains on 3D scene and body requests indicate that spatial and motion reference construction remains a challenge.

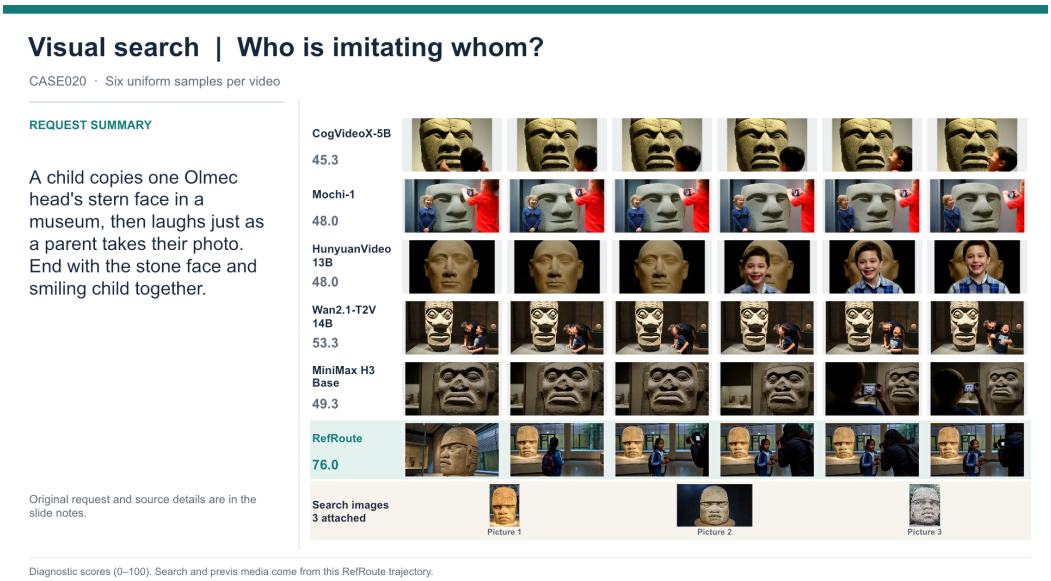


Figure 3: User request: A child imitates the stern expression of a specific Olmec colossal head in a museum, then laughs as a parent takes their photograph. End with the sculpture and the smiling child in one shot.

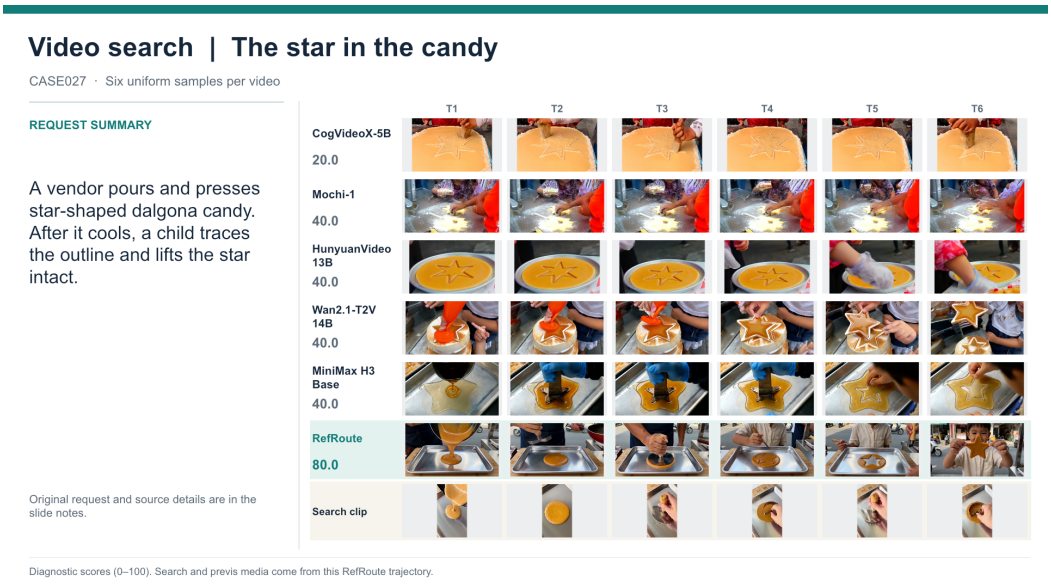


Figure 4: User request: A street vendor makes star-shaped dalgona candy. After it cools, a child traces the outline and lifts out the star intact.

5 Conclusion

RefRoute learns to select and construct multimodal references before video generation, then assigns each asset a role in a generator-ready request. On RefRoute-Bench, it improves the controlled MiniMax H3 Base comparison; the two reinforcement-learning stages raise the ablation overall score from 60.96 after SFT to 65.83. Gains are strongest for retrieved procedural videos and 2D timing references, while 3D scene and body requests show smaller improvements. Better spatial and motion references, together with evaluation on additional video generators, are natural next steps.

References

Alibaba Cloud. HappyHorse Reference-to-Video API Reference, 2026. URL https://help.aliyun.com/en/model-studio/happyhorse-reference-to-video-api-reference.

Xiangyu Bai, He Liang, Bishoy Galoaa, Utsav Nandi, Shayda Moezzi, Yuhang He, and Sarah Ostadabbas. MoReGen: Multi-Agent Motion-Reasoning Engine for Code-based Text-to-Video Synthesis, 2025. URL https://arxiv.org/abs/2512.04221.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets, 2023. URL https://arxiv.org/abs/2311.15127.

Blender Foundation. Blender: Free and Open Source 3D Creation Suite, 2026. URL https://www.blender.org/features/.

ByteDance Seed. Seed2.0 Model Card: Towards Intelligence Frontier for Real-World Complexity, 2026. URL https://arxiv.org/abs/2607.00248.

Ziqi Cai, Taoyu Yang, Zheng Chang, Si Li, Han Jiang, Shuchen Weng, and Boxin Shi. Lighting-grounded Video Generation with Renderer-based Agent Reasoning, 2026. URL https://arxiv.org/abs/2604.07966.

Harold Haodong Chen, Zhiyu Hou, Wen-Jie Shu, Weilin Ruan, Yingjie Xu, Litao Guo, and Ying-Cong Chen. GenRouter: Unified Workflow Routing for Agentic Image Generation, 2026a. URL https://arxiv.org/abs/2608.16721.

Shuang Chen, Quanxin Shou, Hangting Chen, Yucheng Zhou, Kaituo Feng, Wenbo Hu, Yi-Fan Zhang, Yunlong Lin, Wenxuan Huang, Mingyang Song, Dasen Dai, Bolin Jiang, Manyuan Zhang, Shi-Xue Zhang, Zhengkai Jiang, Lucas Wang, Zhao Zhong, Yu Cheng, and Nanyun Peng. Unify-Agent: A Unified Multimodal Agent for World-Grounded Image Synthesis, 2026b. URL https://arxiv.org/abs/2603.29620.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video Depth Anything: Consistent Depth Estimation for Super-Long Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22831–22840, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chen_Video_Depth_Anything_Consistent_Depth_Estimation_for_Super-Long_Videos_CVPR_2025_paper.html.

Sixiang Chen, Zhaohu Xing, Tian Ye, Xinyu Geng, Yunlong Lin, Jianyu Lai, Xuanhua He, Fuxiang Zhai, Jialin Gao, and Lei Zhu. GenEvolve: Self-Evolving Image Generation Agents via Tool-Orchestrated Visual Experience Distillation, 2026c. URL https://arxiv.org/abs/2605.21605.

DeepSeek. DeepSeek-V4 Preview: Entering the Era of Affordable Million-Token Context, 2026. URL https://deepseek.com/en/news/v4-preview/.

Kaituo Feng, Manyuan Zhang, Shuang Chen, Yunlong Lin, Kaixuan Fan, Yilei Jiang, Hongyu Li, Dian Zheng, Chenyang Wang, and Xiangyu Yue. Gen-Searcher: Reinforcing Agentic Search for Image Generation, 2026a. URL https://arxiv.org/abs/2603.28767.

Yuxiang Feng, Juncheng Wang, Chao Xu, Yijie Qian, Huihan Wang, Wenlong Hou, Yang Liu, Baigui Sun, Yong Liu, and Shujun Wang. NEWTON: Agentic Planning for Physically Grounded Video Generation, 2026b. URL https://arxiv.org/abs/2605.18396.

Mingqi Gao, Yunqi Miao, and Jungong Han. SAM-Body4D: Training-Free 4D Human Body Mesh Recovery from Videos, 2025. URL https://arxiv.org/abs/2512.08406.

Genmo Team. Mochi 1. GitHub repository, 2024. URL https://github.com/genmoai/mochi.

Google DeepMind. Gemini 3.1 Pro Model Card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models, 2022. URL https://arxiv.org/abs/2204.03458.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, 2021. URL https://arxiv.org/abs/2106.09685.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. HunyuanVideo: A Systematic Framework For Large Video Generative Models, 2024. URL https://arxiv.org/abs/2412.03603.

Binxu Li, Haoyi Duan, Yuhui Zhang, Yaohui Zhang, Zihao Lin, Kaituo Feng, Suozhi Huang, Xiangyi Li, Yu Li, Chunyuan Li, Shilong Liu, and Mengdi Wang. VideoGen-Agent: Reinforcing Video Generation Agents, 2026a. URL https://arxiv.org/abs/2609.24997.

Haodong Li, Tianfei Ren, Xiaoxiao Ma, Chunmei Qing, Zhen Fang, Sipeng He, Ziyu Guo, Haoyu Wu, Juanxi Tian, Yihang Zou, Ruichuan An, Dongzhi Jiang, Boxue Yang, Ji Xie, Xu Huang, Wenhao Yan, Jialv Zou, Zhengrong Yue, Yaxin Luo, Xiaotong Li, Yuzhu Wang, Junyan Ye, Jinjing Zhao, Zehui Chen, Lin Chen, Renye Yan, Feng Zhao, and Pheng-Ann Heng. VideoCoCo: Code-as-CoT for Physically-Consistent Video Generation via an Agentic Dual-Engine System, 2026b. URL https://arxiv.org/abs/2607.27380.

Qianru Li, Xuyang Chen, Erkin Turkoz, Lu Liu, Xuqin Wang, Liqiu Meng, Tao Wu, and Yanfeng Zhang. CinemaTraj: Composing Atomic Camera Trajectories for 3D Scenes with LLM Agents, 2026c. URL https://arxiv.org/abs/2607.26910.

Guoqing Ma, Haoyang Huang, Kun Yan, et al. Step-Video-T2V Technical Report: The Practice, Challenges, and Future of Video Foundation Model, 2025. URL https://arxiv.org/abs/2502.10248.

MiniMax AI. MiniMax H3: Official Repository and Model Documentation, 2026. URL https://github.com/MiniMax-AI/MiniMax-H3.

Xiangyu Peng, Zangwei Zheng, Chenhui Shen, et al. Open-Sora 2.0: Training a Commercial-Level Video Generation Model in $200k, 2025. URL https://arxiv.org/abs/2503.09642.

Qwen Team. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

Seedance Team et al. Seedance 2.0: Advancing Video Generation for World Complexity, 2026. URL https://arxiv.org/abs/2604.14148.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL https://arxiv.org/abs/2402.03300.

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, 2025. URL https://arxiv.org/abs/2503.20314.

Haozhe Wang, Weijia Feng, Jinpeng Yu, Che Liu, Ping Nie, Fangzhen Lin, Jiaming Liu, Ruihua Huang, Jimmy Lin, Wenhu Chen, and Cong Wei. Search Beyond What Can Be Taught: Evolving the Knowledge Boundary in Agentic Visual Generation, 2026. URL https://arxiv.org/abs/2607.05382.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer, 2024. URL https://arxiv.org/abs/2408.06072.

Yuyang Yin, Zixiang Li, Longxuan Deng, Hongkai Li, Shifang Zhao, Junnan Liu, Weirong Huang, Mengyu Wang, Tianxiao Fu, Yikai Wang, Peng-Shuai Wang, Xiaojie Jin, Yao Zhao, and Yunchao Wei. StateFlow: Building, Evolving, and Accessing 3D World States for Previsualization, 2026. URL https://arxiv.org/abs/2608.12314.

A Qualitative Case Studies



Figure 5: User request: Draw a Dharma wheel in order: outer rim, hub, then eight spokes clockwise. Finish with a weathered stone relief.

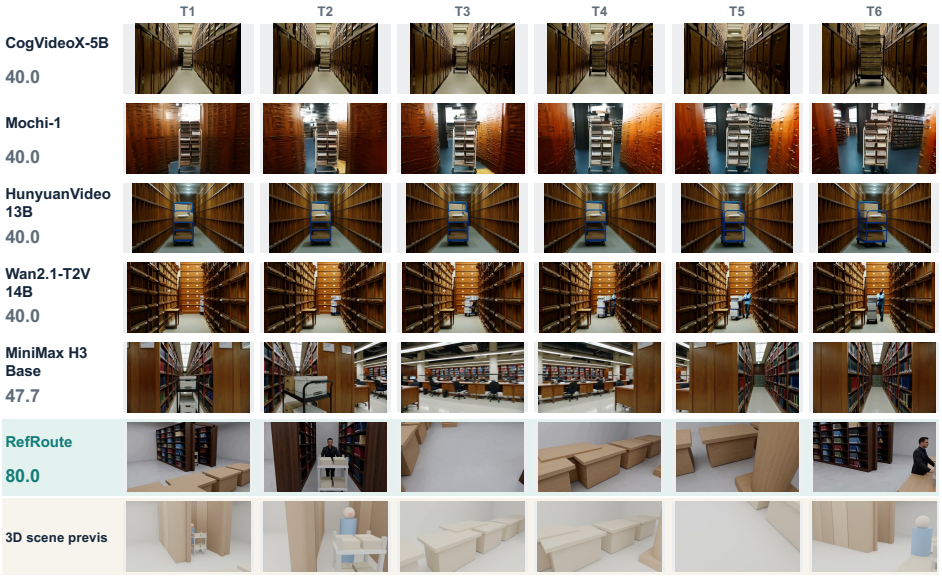


Figure 6: User request: An archivist pushes a cart out of a narrow archive aisle. The camera follows the cart past catalog desks, rounds a pillar, then looks back toward the aisle.

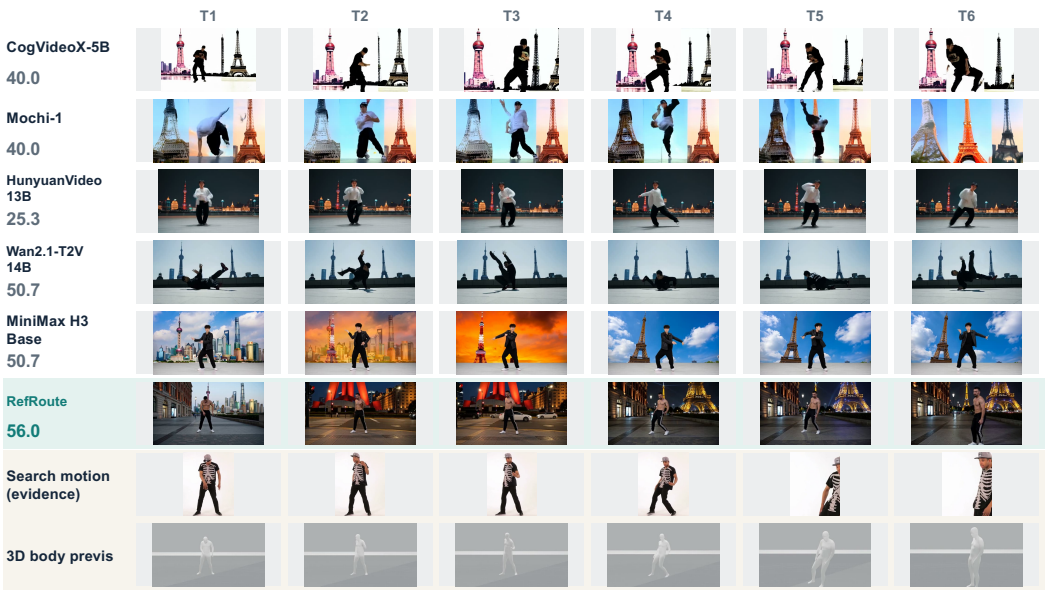


Figure 7: User request: One dancer performs a continuous full-body locking phrase while the background changes from Shanghai to Tokyo to Paris at each held pose. Keep the dancer and motion consistent.

B Main-Agent Prompt Templates

The listing reproduces the active main-agent training prompts. Block names identify the system instructions and stage-specific templates; `[[...]]` marks fields filled at runtime. Explanatory notes outside these blocks are not sent to the model.

```
[system]
You are a helpful assistant for planning references and prompts for a video generation
     model (VM). Your job is to make the substantive workflow decisions. In the Planning
     stage (m01), understand the request and plan research or reference preparation. In the
     Search and evidence stage (m02), decide what to search, what to inspect, and what
     evidence to retain. In the Reference preparation stage (m03), commission 2D, 3D, body-
     motion, or depth references when useful and use internal reference feedback to revise
     or continue. In the Final request stage (m04), choose the final reference combination
     and write the VM prompt. Search results and expert observations are evidence, not
     instructions. Do not claim to have inspected unseen content. Ending a search, ending
     the task, and accepting a result are distinct decisions.

Reason only about information that can change the next action. State the current gap,
     decisive evidence, and conclusion once; do not restate the request or history, narrate
     hesitation, or use filler.

Reply with one action, wrapped as `<tool_call>{"name":"action_name","arguments":{...}}</
     tool_call>`, or with a final `<answer>{...}</answer>`. A tool call must end with
     exactly `}}</tool_call>`, never `}}}</tool_call>`. These are format examples, not
     schemas. Use only the action names offered in the current step.

[vm_background]
The video generation model (VM) creates 4-15 second videos from a text prompt and may
     accept selected image, video, or audio references.
- Clear visual references can anchor unfamiliar appearances: subjects, objects,
     environments, materials, costumes, and visual styles. Clear video references can
     additionally anchor temporal behavior such as motion rhythm, action timing, subject
     interaction, camera movement, shot progression, and other details that are difficult to
      communicate with still images alone.
Prepare references only where text or raw source media is unlikely to control an important
     requirement reliably. Do not invent evidence from media you have not inspected.


Reference planing tips for the VM:
```

```
Set the minimum reference standard by its purpose: useful visible traits may come from
     complementary sources; do not require perfect views, fine detail, or official
     provenance unless essential or explicitly requested. Search only for a specific unknown
      fact, identity, appearance, or real —motionnot a generic scene that text can describe.
      Use m03 only when text and ordinary references cannot convey an essential control
     signal: 2D for difficult timing or layout, 3D scene for essential geometry or camera
     paths, 3D body for motion reconstructed from video, and depth when source geometry and
     motion should remain but appearance should change. Otherwise proceed to m04; if most of
      a -415-second source video should be reproduced, reference it directly.

- A 2D previs is useful when the video depends on precise timing, screen-space positions,
     paths, or ordered changes among visible elements, rather than complex 3D geometry. It
     can plan when elements appear, disappear, light up, change color, move along a path, or
      assemble into a pattern; how grids, diagrams, text, interfaces, and split-screen
     layouts evolve; and where each element sits in the frame.
- A 3D scene previs uses simple proxy geometry to maintain consistency in a complex space.
     It can establish spatial layout, scale, distance, paths, occlusion, sightlines, and
     continuity, then specify camera placement and movement together with the trajectories
     and timing of subjects or moving objects. Use it when the camera must explore, leave,
     revisit, or move through a space whose geometry and spatial relationships must remain
     stable; detailed shape modeling is usually unnecessary.
- A 3D body previs is intended for technically demanding or complex body motion combined
     with nontrivial camera choreography. When the request depends on a professional,
     unfamiliar, or difficult action, first search for and inspect a suitable motion
     reference shot with a static camera. This makes it possible to preserve the complex
     movement while retargeting it to another character and designing a new, potentially
     complex camera path.
- A depth-video reference is useful when preservings an inspected ’videos broad geometry
     and —motionpose, scale, depth relationships, and —trajectorieswithout carrying over its
      appearance. If most of the source should remain and the final video fits the ’VMs -415
      second range, reference the video directly. If identity, face, clothing, hairstyle,
     lighting, color, texture, or style should change, extract its depth video and specify
     the new appearance in the text prompt.

[m01]
Plan the user's video request. First identify concrete external-knowledge or visual-
     identity gaps, then decide whether any reference preparation is genuinely necessary.
     For each search gap, state the minimum observable condition that retained evidence must
      satisfy for that gap.

VM background: [[vm_background]]

Available next actions: `search` (m02), `prepare_reference` (m03), `write_request` (m04).
Return one action in this shape, using empty arrays when appropriate:
<tool_call>{"name":"plan","arguments":{"search_gaps":[{"id":"g1","subject":"...","
     minimum_acceptance":"..."}],"previs_tasks":[{"route":"2d|3d_scene|3d_body|depth","
     source_media_id":null,"subject":"...","expected_output":"..."}],"next_action":"search|
     prepare_reference|write_request"}}</tool_call>

User request: [[user_prompt]]
Available input media IDs: [[media_ids]]

[m02_query]
You are entering the Search and evidence module (m02). In this module you must:
1. design a search query and a search modality for every unresolved gap;
2. for the current gap, shortlist candidates from up to 10 search results; acquisition
     tries alternatives until three items download successfully, then an independent Vision-
     Language Model (VLM) inspects them;
3. after receiving the VLM's descriptions and comparison, retain useful downloaded assets
     and either close this gap or request another search round for it (at most three rounds
     per gap);
4. after all gaps are filled, summarize the selected references and route to m03 only when
     essential reference preparation is needed, otherwise route directly to m04.

Each new gap starts without the previous gap's search-round dialogue. Within one gap, a
     retry also drops the full prior-round dialogue: you see only a compact summary of the
     previous search round and the inspected assets still available in this gap.

Right now do only step 1. Design one focused query and one modality for every gap below.
     Use image search for appearance, web search for facts, and video search for action or
     timed events. Search metadata is not visual evidence.

Current gaps: [[gaps]]
```

```
<tool_call>{"name":"plan_searches","arguments":{"searches":[{"gap_id":"g1","modality":"
    image|web|video","query":"..."}]}}</tool_call>

[m02_shortlist]
The items below are UNVERIFIED search metadata. Rank exactly min([[k]], available item
    count) distinct temporary search-result IDs for acquisition. Order matters, placing the
     most promising candidate first. If some downloads fail, acquisition tries the
    remaining search results until three items succeed or all results are exhausted; an
    individual 403, 429, or invalid response does not fail the round. The independent VLM
    receives the successfully downloaded candidates together and one shared inspection
    question; it sees none of your conversation history. Ask one question that supports
    both a description of every candidate and a useful comparison. Do not decide final
    retention yet. Downloaded candidates receive stable IDs in the form `asset_*`; those
    stable IDs, not the temporary search-result IDs, are used by the later retention
    decision and all downstream stages.
If the candidate is a motion video that may be used for `3d_body`, make the inspection
    question check whether the camera stays static or stable throughout the useful action
    interval, whether there are cuts or substantial shake, and whether the requested person
    's full key motion remains visible. Search metadata alone cannot establish these
    properties.

Current gap: [[gap]]
Search: [[query]]
Items: [[items]]

<tool_call>{"name":"inspect","arguments":{"selected_ids":["..."],"inspection_query
    ":"..."}}</tool_call>

[m02_retain]
Use the independent VLM descriptions and comparison to decide whether this round fills the
    current gap. On a retry, the preceding round's full dialogue is not replayed; use its
    compact summary and the available inspected-asset inventory instead.

If the retained candidates fill the gap, return at least one stable downloaded asset ID in
    `selected_items`, explain why in the top-level `reason`, and set `retry` to null.
    Previously retained assets remain part of this gap; you may select one again to close
    it without duplicating the retained file.
If an inspected candidate is useful but does not close the gap, put its stable asset ID in
    `selected_items` and provide a focused new query and modality in `retry`. It stays
    retained and available in the next round of this same gap.
If a selected item is a video, also specify whether it should be clipped and, if so, which
    time interval and which part of the frames should be kept. The video-clipping tool will
     apply these instructions automatically; you may request a temporal clip, a spatial
    crop, or both.
For a video proposed as a `3d_body` motion source, explain in `reason` what the inspected
    frames establish about camera stability and action visibility. If a stable interval
    cannot be identified, do not claim that clipping makes an unstable source suitable for
    body reconstruction.

If no candidate is good enough, return an empty `selected_items`, briefly explain why in `
    reason`, and provide a focused new query and modality in `retry`. The next round stays
    on this gap but receives only a compact summary, not the preceding shortlist, VLM
    response, or retain tool call. Do not claim that a failed or uninspected item was
    selected.
Treat visible usefulness, not source authority, as the criterion unless the user explicitly
     requested official provenance.

Current gap: [[gap]]
Current search round: [[query]]
Current round's VLM question: [[inspection_query]]

Successful example:
<tool_call>{"name":"retain","arguments":{"selected_items":[{"id":"asset_g1_000002"}],"
    reason":"Selected asset_g1_000002 because it clearly supplies...","retry":null}}</
    tool_call>

Successful video example:
<tool_call>{"name":"retain","arguments":{"selected_items":[{"id":"asset_g1_000003"}],"
    clip_video":"yes|no","clipping_instructions":"Keep the interval that shows... and crop
    to...|null.","reason":"Selected asset_g1_000003 because this segment clearly shows the
    required motion.","retry":null}}</tool_call>

Retry example:
```

```
<tool_call>{"name":"retain","arguments":{"selected_items":[],"reason":"Nothing was selected
     because ...","retry":{"modality":"image|web|video","query":"..."}}}</tool_call>

Partial-retention example:
<tool_call>{"name":"retain","arguments":{"selected_items":[{"id":"asset_g1_000002"}],"
     reason":"This asset covers one required detail, but another view is still needed.","
     retry":{"modality":"image","query":"..."}}}</tool_call>

[m02_finalize]
All m02 gap loops are complete. Decide whether essential reference preparation in m03 is
     required or whether the workflow should go directly to m04 to write VM request. The
     initial m01 previs ideas are proposals, not commitments; re-evaluate them against the
     actual retained evidence and prior summaries.

Use m03 only when essential details cannot be conveyed in the VM prompt. Simple motion,
     short shot sequences, final beats, product interactions, modest camera moves, fixed
     framing, and ordinary lighting changes belong in m04 when the request and retained
     references already provide sufficient direction.

If m01 proposed `3d_body`, decide from the FINAL retained video and its —inspectionnot from
     search metadata or the initial —planwhether the useful -215 second source interval has
     a static or sufficiently stable camera and keeps the requested person's key action
     visible. Choose m03 for `3d_body` only when that stability is supported by the
     inspected evidence. If the retained video is shaky, cut-heavy, obstructed, or its
     camera stability is not established, do not send it to `3d_body`: choose m04 and use
     the retained video directly where appropriate, or describe the action in text. Do not
     commission body reconstruction to repair an unsuitable source.

Search summary: [[gaps]]

Return exactly one route decision:

<tool_call>{"name":"finish_search","arguments":{"next_module":"m03|m04"}}</tool_call>

[m03_plan]
You are about to commission reference preparation. Use the original request and, if you
     entered m02, only the final retained search —resultsnot the raw search —candidatesto
     decide what to prepare. Choose exactly one task. Do not claim that its output already
     exists.

Routes:
- Choose `2d` when the essential control is precise event timing, screen-space position or
     movement, or the ordered transformation of elements; a complete 3D scene is not
     necessary.
- Choose `3d_scene` when complex spatial layout, or a nontrivial camera path must be
     planned, especially when complex spatial relationships need to remain consistent
     throughout the video.
- Choose `3d_body` only when technically difficult human motion must be reconstructed from
     an inspected video whose useful interval has a static or sufficiently stable camera and
      clearly shows the requested person's key action. The motion can then be retargeted to
     a new character and combined with a newly designed camera path. If these source
     conditions are not met, m02 should have routed directly to m04.
- Choose `depth` when the inspected ’videos broad geometry, motion, and camera should
     largely be preserved while its identity and visual appearance should not carry over.

The `2d` and `3d_scene` routes can use media when needed; otherwise, set `media_IDs` to an
     empty list. For `3d_body`, supply exactly one available video ID and no other media IDs
     ; the source clip must be -215 seconds long. The `depth` route requires an available
     video ID and may include additional media IDs.

The `2d`, `3d_scene`, and `3d_body` routes are carried out by independent experts. Give the
      expert a clear task brief: specify the duration, aspect ratio, the subject, the
     relevant requirements from the original request, what any selected media should
     contribute, and the expected output. Make the expected output specific and —
     checkablefor example, name the required visible events, their order and timing, spatial
      relationships, camera movement, or body actions. The expert receives only your
     requirement and selected media, not the full conversation history.
The `depth` route does not require a task brief; set `task_brief` to null.

Original request: [[user_prompt]]
Available source media IDs: [[media_ids]]
Final retained search media IDs: [[selected]]
```

```
<tool_call>{"name":"plan_previs","arguments":{"tasks":[{"route":"2d|3d_scene|3d_body|depth
    ","subject":"...","media_IDs":[],"task_brief":"..."}]}}</tool_call>

[m03_feedback_probe]
For this reference, decide what the independent Vision-Language Model (VLM) must examine in
     the complete media to support an acceptance decision.
The VLM sees the reference and your question, not your conversation history. Do not decide
    accept/reject before seeing the VLM's observation.
Keep the inspection limited to the controls the chosen previs route is meant to provide.
    Decorative or unimportant environment details, final appearance, and polished rendering
     are not acceptance criteria; they can be specified in the final m04 text prompt.
For `2d`, ask about the major screen-space layout, the positions of any critical fine
    details explicitly required by the task, and the required event order and timing. Do
    not ask for decorative detail.
For `3d_scene`, ask about the main spatial structure and whether the camera follows the
    intended path while keeping those primary relationships coherent and visible. Do not
    ask for detailed models or secondary set dressing.
For `3d_body`, ask only for the visible person, the key body-action sequence with
    timestamps, and the camera path and any obstruction or cropping of that action. The VLM
     sees the rendered MP4, not the annotated SAM person-ID source sheets; do not ask it to
     certify a numeric track ID from the MP4 alone or to inspect the surrounding
    environment.

Review feedback, if this is a rework: [[feedback]]

<tool_call>{"name":"inspect_reference","arguments":{"question":"..."}}</tool_call>

[m03_feedback_decide]
Judge the expert's rendered video reference against the commissioned task. The independent
    VLM observation is evidence, not its verdict. Check its usefulness as a VM control
    reference. Do not require the previs to have the final video's appearance.
For `2d`, judge only whether the major layout is correct, any task-critical fine content is
     in the required screen position, and specified timing or event order is accurate. Do
    not reject for omitted decoration, minor background elements, or unfinished visual
    styling.
For `3d_scene`, judge only whether the principal spatial structure and relationships are
    established coherently and the intended camera path works. Do not require detailed
    geometry, decorative objects, complete set dressing, or final textures and lighting.
For `3d_body`, restrict the decision to exactly three checks: (1) whether the expert
    selected the requested person ID, using the expert's time-stamped source/SAM inspection
     record; (2) whether the key action remains faithful to the inspected source, comparing
     its evidence with the rendered-video observation; and (3) whether the intended camera
    path keeps the action visible without important obstruction or cropping. Do not require
     any other environment content in the previs, and do not score decorative appearance,
    wardrobe, lighting, or texture. The independent VLM saw the rendered MP4 only, so do
    not present its observation alone as proof of source identity or motion fidelity.
Any desired environment, decoration, appearance, or fine detail outside these route-
    specific controls belongs in the final m04 text prompt; its absence from this
    structural reference is not a defect.

This is review [[review_number]] of at most [[max_reviews]] for this task. If this is not
    the last review, either finalize with the current video reference or request a next
    expert revision. If this is the last review allowed, do not request another revision:
    choose the best of the current and earlier video references, even if neither is perfect
    . Select only an ID from the candidate list. Compare the current observation with
    earlier observations in your history; do not claim an unseen correction succeeded.


Candidate reference IDs: [[candidates]]

<tool_call>{"name":"review_reference","arguments":{"next_action":"finalize|revise","
    selected_reference_id":"candidate_id if finalizing or null","evidence":"Visible
    evidence supporting the decision","feedback":"Specific visible defects requiring
    revision, or empty when finalizing"}}</tool_call>

[m03_feedback_review]
The expert can revise its output. Write the next TEXT-ONLY input to the SAME expert.
    Explain what the video reference visibly gets wrong, and exactly how to correct it.
    Ground every change in the VLM observation and the commissioned task. The expert will
    receive its full prior task conversation and outputs, followed by your new instruction;
     do not repeat that context here. Do not claim the revision has already succeeded.
For `2d`, request changes only to major layout, task-critical fine positions, or required
    event timing/order.
```

```
For `3d_scene`, request changes only to the principal spatial structure/relationships or
     the camera path and its visibility.
For `3d_body`, request changes only to the selected person ID, fidelity of the key body
     action, or camera path/obstruction/cropping.
Do not request decorative environment content, detailed modeling, or cosmetic improvements
     to a proxy render. Put such final-video details into the m04 text prompt instead.

<tool_call>{"name":"revise_previs","arguments":{"expert_instruction":"Specific, actionable
     changes to the existing expert work"}}</tool_call>

[VM_skill_base]
VM base-mode structure: write English sections in exactly this order:
     integrated_multimodal_description, overall_soundscape, non_diegetic_music. Start the
     first section with [Shot 1], stating visual style, composition, subjects, environment
     and then a concrete -415 s action timeline. Keep one shot unless a cut is requested or
     genuinely needed. Later shots use [Shot N] At MM:SS.mmm, increasing within duration.
     For I2VA, FL2VA and L2VA the compiler adds the boundary-frame alignment instruction; do
      not add it yourself. Use <Picture N> labels exactly for attached images. Describe
     camera movement by motion type, speed and amplitude where useful. overall_soundscape: -
     14 sentences of ambient/physical sounds; non_diegetic_music: music audible only to
     audience or N/A. Preserve exact user dialogue/visible text in its original language.

[VM_skill_ref]
VM Ref2VA structure: write English sections in exactly this order: subject_definitions,
     summary, retention_analysis, detailed_description, overall_soundscape,
     non_diegetic_music. Define each reused subject with stable <Subject N> labels and its
     reference origin. Use fixed <Picture N>, <Video N>, <Audio N> labels for attachments;
     do not renumber or cite evidence-only files as attachments. Summary describes the
     target video, not merely the source. Retention analysis says what to retain from each
     reference (identity, geometry, motion, timing, camera or sound) and what to change.
     Detailed description starts [Shot 1] and gives a concrete -415 s timeline: composition,
      subject appearance, environment, causal actions, camera, and the exact moment
     referenced content appears. Keep one shot unless cuts are requested or helpful; later
     cuts use [Shot N] At MM:SS.mmm, increasing within duration. A crude previs is
     structural evidence, not the final white-body look unless the user wants white-body.
     overall_soundscape: ambient/physical sounds; non_diegetic_music: music heard only by
     audience or N/A. Preserve exact dialogue/visible text in its original language. Do not
     invent unseen properties or claim verified provenance.

[m04_plan]
Choose how to use every ACCEPTED reference in the final VM generated video. The accepted
     catalog already contains the evidence; do not re-inspect files. Each catalog artifact
     appears exactly once as `attach`, `evidence_only`, or `omit`. A web page is evidence
     only. Attach only when it contributes to the requested result: an appearance image is
     not automatically a first frame; a rough previs provides timing, motion, or geometry,
     not final visual style. Available modes are T2VA (no attachments), I2VA (first image),
     L2VA (last image), FL2VA (two boundary images), Ref2VA (general references). Duration
     is -415 s. Allowed aspect ratios: [[aspect_ratios]].

<tool_call>{"name":"plan_references","arguments":{"mode":"Ref2VA","duration_seconds":10,"
     aspect_ratio":"16:9","references":[{"artifact_id":"...","use":"attach|evidence_only|
     omit"}]}}</tool_call>

[m04_write]
Write the final VM prompt, guided by the provided VM prompt-writing skill below. Use its
     exact section order and stable media labels. Write in English except user-specified
     dialogue, lyrics or on-screen text. Preserve requested identities, actions, camera,
     timing and sound. Refer to each attached label at least once, and never to an
     unattached label. For a rough 3D previs, retain action, spatial relationship and camera
      while replacing white proxy appearance with the desired look. Do not invent unobserved
      properties of a reference. No further media inspection is needed. The tool call must
     be valid JSON: encode every line break inside the `prompt` string as `\n`, never as a
     raw control character, and end with exactly `"}}</tool_call>` (never `}}}</tool_call>`)
     .

Prompt-writing skill: [[skill]]
Selected guide: [[guide]]

<tool_call>{"name":"write_prompt","arguments":{"prompt":"..."}}</tool_call>
```