UNIFYING VIDEO TASKS VIA SPATIOTEMPORAL ANALOGY

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting video models to new tasks typically requires dedicated data curation and fine-tuning. While visual analogy provides a training-free alternative by specifying tasks in-context, it remains restricted to the image domain. To explore whether analogy-based methods can unify diverse video tasks and generalize to out-of-distribution scenarios, we introduce **VIGEO**, a framework that extends visual in-context learning to the video domain via spatiotemporal canvas completion. Evaluated on a diverse task taxonomy with a strict train-test split, **VIGEO** generalizes to unseen video manipulations and unseen modalities (e.g., event cameras). Finally, we identify *task internalization*, where a query format associated with a pretrained task overrides the demonstration, and show that this shortcut can be removed with a small amount of task-unrelated data, highlighting the need to decorrelate prompt format from task identity.

1 INTRODUCTION

Traditionally, cultivating a new capability in visual generative models demands dedicated data curation, architectural modifications, and task-specific fine-tuning. Large language models have demonstrated a compelling alternative: models can learn to execute new tasks simply by observing a few demonstrations in context (Brown et al., 2020; Gao et al., 2021; Zhao et al., 2021; Schick & Schütze, 2021; Liu et al., 2022a; Dai et al., 2023). In vision, this principle takes the form of visual analogy (Hertzmann et al., 2001; Bar et al., 2022), where a model infers a mapping from a source-target image pair and applies it to a novel query. Yet, this capability has remained largely confined to still images, leaving the domain of video largely unexplored.

Extending visual analogy to video is challenging because video tasks demand strict spatiotemporal consistency and complex motion understanding, requirements that currently drive researchers to build specialized datasets and architectures for every new capability. This paper investigates whether a single, analogy-based formulation can absorb this diversity. We ask: *Can a single analogy-based framework effectively unify diverse video tasks and generalize to tasks it has never seen?*

To explore these questions, we must first design a model capable of video analogy without relying on task-specific pipelines. We introduce **VIGEO**, which formulates the problem strictly as a spatiotemporal canvas completion task. Rather than relying on auxiliary task heads or textual descriptions, **VIGEO** arranges the demonstration input, demonstration target, and query input into a single composite video grid. The model is then trained to generate the missing quadrant, i.e., the query target, based purely on the visual context. This framework reduces diverse generative operations to a single, unified visual-completion objective, allowing the model to learn complex mapping relationships entirely from pixels.

While the canvas formulation is task-agnostic in principle, the central question is whether it can support the spatiotemporal complexity of real video operations. We therefore evaluate **VIGEO** through a structured task taxonomy spanning three categories that demand spatiotemporal understanding: camera control, modal-guided generation (from signals such as optical flow or point tracks), and video manipulation. This taxonomy is designed to test two complementary properties of the framework: whether a single canvas can *unify* diverse in-distribution video tasks, and whether the learned analogy *transfers* to tasks that are never seen during training. Concretely, camera control serves as an in-distribution benchmark against dedicated baselines, while modality-guided generation and video

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

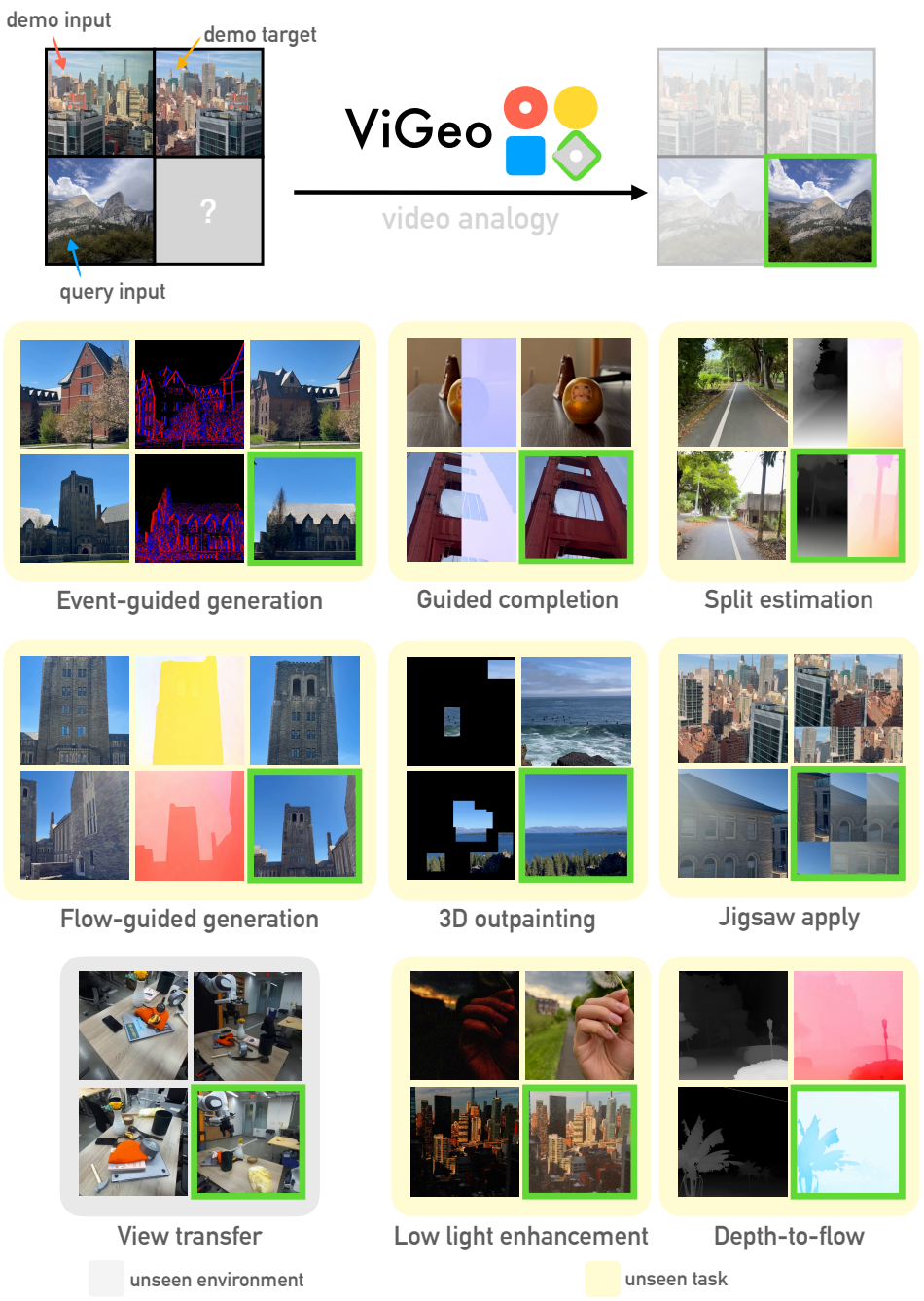


Figure 1: **VIGEO unifies and generalizes diverse video tasks via video analogy. VIGEO** achieves *video analogy* by inferring a transformation from a demonstration pair and applying it to a novel query. We show that this formulation unifies a highly diverse taxonomy of tasks, including camera motion and view transfer. Additionally, it exhibits strong generalization to unseen guided generation and video manipulation tasks. **VIGEO** predictions are highlighted in green. *Video results are provided in the* ***supplementary material****.*

manipulation are evaluated under a strict task-level train-test split. This design allows us to measure genuine in-context learning from the demonstration, rather than memorization of trained task formats.

Empirically, **VIGEO** succeeds on both in-distribution tasks and held-out transfer tasks. On camera control, it remains competitive with task-specific baselines, showing that a single canvas can unify

structured in-distribution video operations without task-specific modules. More importantly, it also generalizes beyond its training taxonomy. For modal-guided generation, **VIGEO** matches or surpasses purpose-built methods on several reconstruction metrics and transfers to unseen guidance modalities such as event camera data. For video manipulation, it correctly infers and executes the demonstrated transformation on 7 of 12 strictly held-out tasks, indicating that the model is not merely copying inputs but performing in-context learning over video transformations.

Beyond benchmark results, we examine when the model actually relies on its demonstrations. We identify ***task internalization***, a failure in which the task is resolved from the query's *format* rather than from the demonstration. On held-out depth-to-flow task, the same demonstration yields either the desired optical flow or a copy of the RGB video, depending only on whether the RGB hint appears once or is repeated across frames. We trace this to a format–target shortcut in the pretraining corpus, where every 2×3 canvas with a frozen RGB video in query has an RGB target, and show that finetuning on a small amount of data containing neither depth nor optical flow removes the shortcut and restores demonstration-following in both held-out directions. This suggests that in-context generalization depends not only on which tasks a model is trained on, but on whether the prompt format alone reveals the task.

In summary, our contributions are as follows:

- **Formulation:** We propose **VIGEO**, extending visual analogy to the video domain via a pure spatiotemporal canvas completion framework.
- **Rigorous Task-Level Evaluation:** We construct a comprehensive video task taxonomy with a strict task-level train-test split, enabling the explicit measurement of emergent generalization rather than memorization.
- **One-Shot Capabilities:** We demonstrate that **VIGEO** achieves competitive or even superior performance against task-specific baselines, successfully generalizing to unseen modalities (e.g., event cameras) and novel video manipulation tasks.
- **Analysis:** We identify *task internalization*, in which during evaluation a specific query input format overrides the demonstration. We trace it to a format–target shortcut in the training data and remove it with a small amount of task-unrelated data, showing that in-context generalization requires prompt formats that do not themselves reveal the task.

2 Related Work

Visual (image) analogies. The idea of transduction with visual demonstration is first explored in Image Analogies (Hertzmann et al., 2001). In this earliest conception, it defines it as "synthesis of a new filtered target image" by first inferring the filters from a pair of images in the *design* phase and applying the transformation on a target image in the *application* phase, opening up many earlier works (Reed et al., 2015; Liao et al., 2017).

Recently, Visual Prompting (Bar et al., 2022) fits this idea into a general self-supervised learning scheme by reformulating it as an inpainting/completion pretext task. Since then, numerous works have been proposed, such as (Xu et al., 2023; Šubrtová et al., 2023; Wang et al., 2023a;b; Nguyen et al., 2023; Zhang et al., 2024; Gu et al., 2024; Lai et al., 2025; Srivastava et al., 2025; Kim et al., 2025; Li et al., 2025). However, they come with one of the following shortcomings: (1) Focus on object-centric editing, manipulation, or style transfer tasks (Šubrtová et al., 2023; Nguyen et al., 2023; Srivastava et al., 2025; Kim et al., 2025; Li et al., 2025; Lai et al., 2025), limiting their scope to localized or low-level transformation; (2) Use auxiliary text prompt (Wang et al., 2023b; Gu et al., 2024; Lai et al., 2025), making it hard to disentangle the effect of the text prompt and image demonstration; or (3) Evaluate on training tasks, with a few exceptions like (Li et al., 2025). In comparison, our **VIGEO** operates in the video regime, focus on scene-level, video-prompt setup, and only evaluates on testing tasks or scenarios.

Concept transfer in video generation. In the video domain, one-shot learning has been focusing on concept transfer in specific domains, such as camera pose (Luo et al., 2025; Mitchel et al., 2025), motion (Sun et al., 2024), and editing (Zhang et al., 2025b). In these works, architectural changes and specific objective functions are designed, but are not yet ready to extend towards diverse video tasks. Our **VIGEO** demonstrates a generic framework by drawing visual analogies.

Table 1: **Overview of task taxonomy and train/eval protocol.** We train on a diverse set of task families that all share a single video analogy canvas $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$: a demonstration pair ($A$, B) shows the transformation, and the model must generate D given the query input C. Evaluation tasks are strictly held out to ensure the model never sees them during training. For 2×3 layouts, the input spans two grid cells (separated by "|"). See Appendix C for full per-task schemas.

Category	Task Family	A/C (input)	B/D (target)	Layout	#	Split
CAMERA	Motion transfer	still frame	video @ traj.	2×2	1	Train + Eval
	View transfer	video @ pose 1	video @ pose 2	2×2	2	Train + Eval
MODAL	Modality estimation	rgb video	modality video	2×2	5	Train
	Multi-modal quad	rgb video	4-modal grid	2×2	1	Train
	Segmentation	rgb video	mask / matted	2×{2, 3}	2	Train
	Guided generation	frozen rgb vid. | modal	rgb video	2×3	4	Train
	Flow-guided gen	frozen rgb vid. | flow viz	rgb video	2×3	1	Eval
	Point-guided gen	frozen rgb vid. | point map	rgb video	2×3	1	Eval
	Event-guided gen	2 anchor frames | event viz	rgb video	2×3	1	Eval
MANIP.	Zoom (in/out)	wide view	close-up	2×2	4	Train
	Photometric editing	original	edited	2×2	1	Train
	Spatiotemporal perm.	permuted	original	2×2	4	Train
	Inpainting / outpainting	masked video	original	2×2	2	Eval
	Deblurring	blurred	sharp	2×2	3	Eval
	Temporal interp.	sparse frames	full video	2×2	2	Eval
	Jigsaw (2D)	shuffled tiles	original	2×2	2	Eval
	Symmetry / low-light	flipped or dark	original	2×2	3	Eval

3 METHOD

3.1 CANVAS SETUP

Consider a demo input video ($A \in \mathbb{R}^{T\times H\times W\times 3}$), a demo target ($B$), a query input ($C$), and a query target ($D$). A video analogy requires the mapping from A to B is the same as the mapping from C to D, i.e., $A : B \sim C : D$. We define a canvas construction operator $\mathcal{C}$. This operator spatially arranges the videos into a large composite video canvas $\mathcal{C}(A, B, C, D) \in \mathbb{R}^{T\times(2H+m)\times(2W+m)\times 3}$, forming a 2×2 grid separated by a margin of size m. The demonstrations are placed in the top row and queries in the bottom, yielding $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$ (the margin is omitted for clarity).

3.2 TASK DESIGN

As the canvas treats each block as a pixel video, a wide range of video tasks fit the same interface: RGB scenes, depth maps, flow visualizations, segmentation overlays, and point trajectories all occupy the same pixel space, enabling a single model to absorb diverse tasks. We exploit this flexibility to unify a diverse set of video tasks under a single framework, spanning three categories: camera control, modal-guided generation, and video manipulation. Table 1 summarizes the full taxonomy.

Training tasks. We train on over 20 task families drawn from complementary data sources, including SpatialVID (Wang et al., 2025a), DL3DV (Ling et al., 2024), SynCamVideo (Bai et al., 2024), CameraClone (Luo et al., 2025) BridgeData-v2 (Walke et al., 2023), DROID (Khazatsky et al., 2024), SA-V (Kirillov et al., 2023), and Hypersim (Roberts et al., 2021). These tasks collectively expose the model to modality estimation and generation, geometric camera transformations, and spatiotemporal permutations, all presented through the same canvas interface without any task-specific text.

Evaluation protocol. To rigorously test generalization, we enforce a strict task-level train/test split: evaluation tasks are never seen during training. The held-out tasks probe three transfer axes, including (1) *unseen conditioning modality*: flow, point, and event-camera guidances are never used

as conditioning signals during training, and event-camera is the genuine unseen-modality case as it is absent from every training task; (2) *unseen operation*: 2D jigsaw, deblurring, and flipping are structurally distinct from the 1D strip shuffles, warps, and photometric edits seen during training; (3) *unseen environment*: camera view transfer tasks are evaluated on physically held-out buildings and rooms. This design ensures that strong evaluation performance reflects genuine in-context performance from the demonstration. Furthermore, we split the source videos to train set and test set, so that any clip used in the test time, whether as a source or target and whether in the demo or query row, is not seen during training.

3.3 Learning Objective

We train the model to complete the target quadrant D using a Rectified Flows objective (Liu et al., 2022b). Let $\mathcal{E}$ denote the video encoder and f be our generative model. During training, our target data point is the encoded query target, $x_1 = \mathcal{E}(D)$, and we sample standard Gaussian noise $x_0 \sim \mathcal{N}(0, I)$. For a given time step $t \in [0, 1]$, the noisy latent x_t is defined as a linear interpolation between the noise and the target: $x_t = tx_1 + (1 - t)x_0$, whereas the ground-truth velocity v_t is defined as $v_t = \frac{dx_t}{dt} = x_1 - x_0$. The model is trained to predict this velocity, and the loss function is formulated as the mean squared error between the model's prediction and the true velocity.

To condition the prediction on the canvas context, we make a few adjustments. Let M denote a zero-mask video of the same dimensions as D. We first encode the masked canvas, obtaining the contextual latents $\mathcal{E}(\mathcal{C}(A, B, C, M))$. We then construct the full noisy canvas latent, denoted as z_t, by replacing the quadrant corresponding to the masked query target M with our noisy state x_t. The generative model f takes this composite latent z_t and the timestep t as inputs to predict the velocity of the target region. The overall training objective is defined as:

$$\mathcal{L} = \mathbb{E}_{x_0, x_1, t}\left[\|v_t - f(z_t, t)\|_2^2\right] \tag{1}$$

3.4 Training Setup

All the videos (i.e., A, B, C, D) are cropped and resized to 480×480, with a length of 81. With a margin m of 32, the canvas spatial resolution is 992×992 or 992×1504, corresponding to 2×2 or 2×3 layout. We perform full finetuning on Wan2.2-TI2V-5B (Wan et al., 2025) with a learning rate of 0.00005 and a batch size of 512 for 10K steps. The video encoder has a pixel patch size of $16 \times 16 \times 4$ and a latent patch size of $2 \times 2 \times 1$, and thus we select the margin size m to be 32. We use FastVideo (Zhang et al., 2025a) and train the model across 32 NVIDIA A100 nodes.

4 Experiments

4.1 Modal Guided Generation

Setup. We evaluate three guided generation tasks, all strictly held out during training. Each task uses a 2×3 canvas layout. Specifically, for flow- and point-guided generation, the input (A/C) consists of the first frame alongside the modality visualization, whereas the target is the full RGB video (B/D). For event-based interpolation, two anchor RGB frames are provided and the target is the full RGB video. The three evaluation modalities are optical flow, point trajectories (16 uniformly sampled points), and event camera data. We compare against task-specific baselines: FloVD (Jin et al., 2025) and Go-with-the-flow (Burgert et al., 2025) for flow; Tora (Zhang et al., 2025c) and Wan-Move (Chu et al., 2026) for points; CBMNet (Kim et al., 2023) and RE-VDM (Chen et al., 2025a) for event-based interpolation, where our model digests rendered event data visualization. Since these tasks consider dense control signals, we adopt metrics including optical flow end-point error (EPE) and reconstruction quality (PSNR, SSIM, LPIPS).

Table 2 presents the quantitative results. For flow-guided generation, **VIGEO** achieves exceptional performance across all metrics. Despite lacking any prior exposure to optical flow as conditioning signals, it yields the lowest end-point error (EPE) and the highest reconstruction scores (PSNR, SSIM, and LPIPS), outperforming even dedicated flow-conditioned architectures.

Table 2: Quantitative comparison of our method against strong task-specific baselines on flow-, point-, and event-guided generation. Notably, all three tasks are unseen to our model. The **best** and second best results are bolded and underlined, respectively.

Task	Method	Optical Flow	Reconstruction Metrics		
		EPE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Flow-Guided Generation	FloVD (CVPR 2025)	3.771	12.363	0.240	0.539
	Go-with-the-flow (CVPR 2025)	0.356	19.214	0.491	0.181
	VIGEO	**0.185**	**22.365**	**0.671**	**0.106**
Point-Guided Generation	Tora (CVPR 2025)	0.612	6.923	0.366	0.230
	Wan-Move (NeurIPS 2025)	0.289	17.048	0.365	**0.157**
	VIGEO	**0.275**	**19.515**	**0.543**	0.226
Event-Based Interpolation	CBMNet (CVPR 2023)	8.020	**23.587**	**0.759**	0.284
	RE-VDM (CVPR 2025)	**3.384**	20.892	0.665	0.239
	VIGEO	4.539	21.035	0.652	**0.195**

In the point-guided setting, conditioned on sixteen, uniform points, **VIGEO** remains highly competitive against purpose-built baselines such as Tora (Zhang et al., 2025c) and Wan-Move (Chu et al., 2026). It secures the lowest optical flow EPE and the highest PSNR and SSIM scores, demonstrating its robust ability to parse and extrapolate from spatial prompts.

Finally, we push the model's out-of-distribution generalization further by evaluating video frame interpolation guided by event camera data, a visual representation entirely absent from the training distribution. **VIGEO** adapts to this novel sensory input as rendered images and achieves the best LPIPS score among all methods. The remaining metrics reveal a trade-off relative to the specialized baselines: **VIGEO** achieves lower optical flow EPE than CBMNet (Kim et al., 2023) and higher PSNR than RE-VDM (Chen et al., 2025a), but it trails RE-VDM in EPE and CBMNet in both PSNR and SSIM. We attribute this gap to the fact that both baselines are designed and trained specifically for event-based interpolation, whereas **VIGEO** handles event inputs without task-specific design. These results suggest that **VIGEO** produces perceptually faithful interpolations on an unseen modality, while specialized methods retain an advantage in pixel-level fidelity and motion accuracy.

4.2 CAMERA CONTROL

Table 3: Quantitative comparison of our method on the `camera_motion_transfer` task. The **best** and second best results are bolded and underlined, respectively.

Task	Dataset	Method	Camera Metrics		
			RotErr ↓	TransErr ↓	CamErr ↓
Camera Motion Transfer	DL3DV	CamCloneMaster (SIGGRAPH Asia 2025)	3.367	13.941	15.408
		XFactor (ICLR 2026)	**1.066**	**5.378**	**5.761**
		VIGEO	2.173	14.612	15.203

Table 4: Quantitative result on the `camera_view_transfer` task. The **best** results are bolded.

Task	Dataset	Method	Camera Metrics		
			RotErr ↓	TransErr ↓	CamErr ↓
Camera View Transfer	BridgeData-v2	XFactor (ICLR 2026)	36.131	0.857	1.235
		VIGEO	**2.382**	**0.093**	**0.117**
	DROID	XFactor (ICLR 2026)	64.824	1.200	1.959
		VIGEO	**9.149**	**0.116**	**0.261**

Setup. We evaluate two camera control tasks using a 2×2 canvas. For `camera_motion_transfer`, the demonstration shows a static frame (A) animated along a

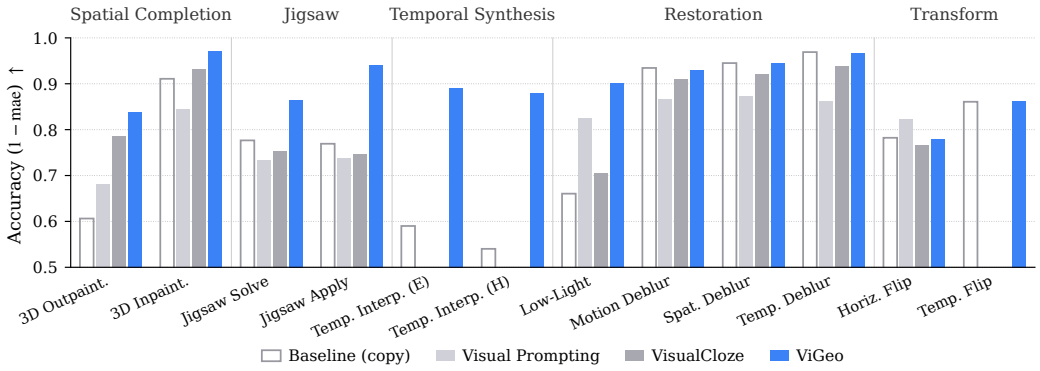


Figure 2: **Generalization to unseen video processing tasks.** We evaluate on 12 held-out video processing tasks and report the accuracy between the model output and the target video. Our model generalizes to 7 of 12 unseen tasks, while deblurring and geometric flipping remain challenging, and the model merely copies the input query video.

trajectory (B); the model must apply the same camera motion to the query static frame (C). We evaluate on DL3DV (Ling et al., 2024). For `camera_view_transfer`, the demonstration shows a scene from one viewpoint (A) captured at a different pose (B, with synchronized cameras); the model generates D at the target pose given query input C. We evaluate on held-out environments from BridgeData-v2 (Walke et al., 2023) and DROID (Khazatsky et al., 2024). Baselines include XFactor (Mitchel et al., 2025) (frame-level) and CamCloneMaster (Luo et al., 2025) (video-level). Metrics are rotation error (RotErr, in degrees), translation direction error (TransErr), and combined camera error (CamErr); see Appendix A.2 for reconstruction and covisible metrics.

In Table 3, we benchmark **ViGeo** against both frame-level and video-level baselines for the `camera_motion_transfer` task. While XFactor achieves the highest overall scores on this specific task, **ViGeo** achieves highly competitive performance and outperforms CamCloneMaster on both rotation and overall camera error.

For `camera_view_transfer`, **ViGeo** achieves lower camera errors than XFactor on held-out environments from BridgeData-v2 and DROID (Table 4). We note that this comparison may be unfair due to a difference in training domains: **ViGeo**'s training data includes such dual-camera setup from both datasets, whereas XFactor is trained on human-captured videos. As shown in our analysis in Appendix A.4, XFactor's degradation may stem from the large roll and pitch angles of the dual-camera setup, which are rare in its training data. Taken together, our results therefore suggest that, when trained on diverse camera distributions, the video analogy formulation allows **ViGeo** to infer large multi-axis rotations from a single demonstration pair and transfer them to a new query.

4.3 Video Manipulation

Setup. We evaluate **ViGeo** on 12 unseen tasks using a 2×2 canvas. These tasks span spatiotemporal completion (2), jigsaw (2), temporal synthesis (2), restoration (4), and spatiotemporal transforms (2). We compare against a naive copy-input baseline, Visual Prompting (Bar et al., 2022), and VisualCloze (Li et al., 2025), a recent model demonstrating emergent out-of-domain capabilities. We calculate the pixel-wise mean absolute error (MAE) and report accuracy as $1 - \text{MAE}$.

As shown in Figure 2, on 7 of the 12 tasks, the output of **ViGeo** is substantially better than all three baselines, confirming that it has successfully inferred and applied the demonstrated transformation rather than defaulting to a copy. The model handles spatial permutations well, effectively applying and inverting jigsaw shuffles, and performs reliable temporal interpolation. For strongly generative tasks such as 3D outpainting, while the absolute accuracy is comparatively lower than inpainting, it is likely because it requires hallucinating novel content rather than rearranging existing pixels, **ViGeo** still demonstrates a significant improvement over the copy baseline.

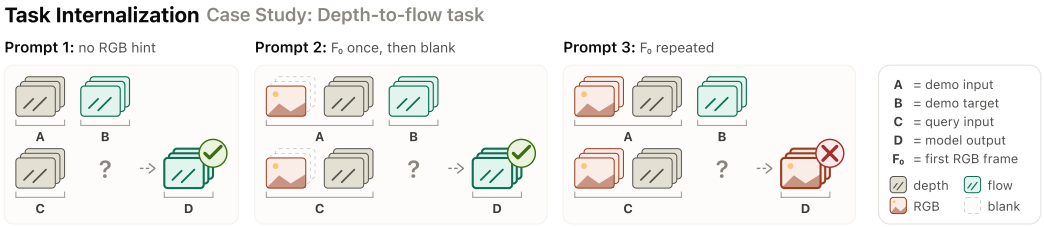


Figure 3: **Illustration of *task internalization*.** We test an unseen depth-to-flow translation task under three different demo prompts. In prompt 1 (no RGB image provided, 2×2 format) and prompt 2 (RGB hint provided, where the first RGB frame appears once), the model succeeds in outputting optical flow visualization video. In prompt 3, where the RGB hint is the first frame repeated across all frames, the model's outputs collapse to RGB videos.

Conversely, the model struggles with the remaining 5 tasks (deblurring and flipping), where it fails to generalize and defaults to the copy baseline. This divergence in performance suggests a consistent pattern: operations that preserve local pixel structure transfer more successfully, while those requiring global geometric reorganization (flipping) or the inversion of a destructive process (deblurring) do not. We hypothesize that test-time generalization to unseen tasks is not unconditional; rather, it requires the underlying transformation to share structural primitives with those seen during training. Because deblurring and flipping involve spatial convolution and global permutation operations that are absent from our training taxonomy, the model struggles to infer them. This implies that for future work, increasing the structural diversity of the training taxonomy may be more critical than simply expanding the dataset size for existing tasks.

5 A Deep Dive to Video Analogy

The same analogy can be expressed through different query formulations. For example, an unseen depth-to-flow translation can be prompted with a [depth; optical flow] demo alone. But what happens if we add an RGB hint in the query like [first-frame RGB, depth; optical flow]? In this section, we study two sensitivities: (1) *task internalization* (Section 5.1), where a familiar query format overrides the demonstration and the model executes the trained task instead of the demonstrated analogy. We then break this format–task coupling by finetuning with an additional normal-to-edge task to mitigate the problem. (2) *perceptual invariance* (Appendix B.2), where the model abstractly interprets modal guidance across different color palettes.

5.1 Task Internalization: The Interplay Between Familiar Queries and Novel Demonstrations

Probing Setup. As shown in Figure 3, we compare three query formats: (1) depth only, in the 2×2 format; (2) the first RGB frame followed by black frames, in the 2×3 format; and (3) the first RGB frame repeated across all frames, i.e. a frozen video. The task is out-of-distribution, so a well-behaved model should follow the input–target relation given in the demonstration and output an optical-flow visualization in all three cases. To detect the failure mode, we measure the Pearson correlation between the generated block and the query clip's RGB video; a high value means the model reproduced the RGB video instead of the demanded modality. We run both depth-to-flow and flow-to-depth, with n=55 samples each task.

Findings. The left panel of Figure 4 shows the result. Under prompt 3 the model outputs RGB videos, reaching correlations of 0.85 (depth-to-flow) and 0.81 (flow-to-depth) with the RGB input. Under prompts 1 and 2 the same model produces the correct modality, with correlations near zero (e.g., -0.02 and -0.01 respectively for the depth-to-flow task). The results have two implications. First, the failure is *format-dependent rather than task-dependent*: the outcome flips according to how the RGB hint is given. Second, *a single RGB frame does not trigger it* — prompt 2 also supplies RGB, yet behaves like the no-hint condition.

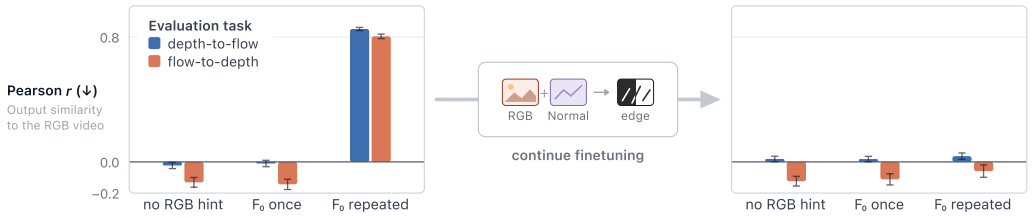


Figure 4: **Task internalization, and its mitigation.** Correlation between the generated output and the query's RGB video is computed; a higher value means the model outputs RGB instead of the demanded modality. **Left:** the pretrained model spikes to 0.85 and 0.81 under the repeated RGB hint, in both held-out tasks, and sits near zero in every other condition. **Right:** after finetuning on a normal-to-edge task that contain neither depth nor optical flow, the spike disappears.

Hypothesis: a format–target shortcut. Upon inspecting the training task composition, we find a structural pattern that potentially accounts for this phenomenon. In the training tasks, every 2×3 canvas containing a frozen RGB video query input has an RGB video as its target. The implication of *a* 2×3 *layout with a frozen RGB column* $\Rightarrow$ *RGB target* therefore holds. We term the behavior ***task internalization***: once the query matches a familiar task template, the model collapses to its pretrained prior. The pattern mirrors *cue competition* (Rescorla, 1972), in which two informative cues are presented together and the more salient one absorbs the prediction error. This also echoes the in-context learning behaviors in large language models: when the query format suffices to recognize a pretrained task, the demonstration is ignored (Min et al., 2022), and genuine task learning occurs only when recognition fails (Pan et al., 2023).

5.2 Mitigating Task Internalization

Insight. Since the shortcut is structural rather than task-specific, simply training on the tasks of interest would invalidate the held-out split. Yet, this same property itself suggests a solution: because the shortcut is not tied to any task, we may be able to break the format–target correlation using unrelated content. We can construct a 2×3-layout task where the target are not RGB videos anymore, forcing the model to check demo targets, and break the format–target shortcut.

Experiment. We build a small-scale normal-to-edge transformation dataset (1.1% of the original training corpus) and mix it into the original pool with a $10\times$ oversample and continue finetuning the model for 750 steps. The right panel of Figure 4 shows the effect. Under the repeated RGB hint prompt, correlation with the RGB video falls from 0.85 to 0.04 for depth-to-flow and from 0.81 to -0.06 for flow-to-depth. See Section B.1 for more finetuned model results.

6 Conclusion

We introduced **VIGEO**, a framework that extends visual analogy to the video domain by reducing diverse video tasks to a single spatiotemporal canvas completion objective. By arranging demonstration and query videos into a unified grid and training the model to generate the missing target quadrant, **VIGEO** absorbs a broad taxonomy of video operations, including camera control, modal-guided generation, and video manipulation, without task-specific text, architectures, or modules. Evaluated under a strict task-level train-test split, **VIGEO** matches or is competitive with dedicated baselines on entirely unseen modalities (event cameras) and successfully generalizes to 7 of 12 held-out video manipulation tasks. Finally, our discovery of task internalization, where familiar query formats can override in-context demonstrations, provides critical insights into the design principles of spatiotemporal prompting for future generalist video models.

AI USE STATEMENT

In this work, we used generative AI tools (Claude) to implement data processing and reformating pipeline and support qualitative data analysis especially in checking the re-implementation of previous baselines.

We did not use generative AI tools for translation, interpreting results, designing research methodology or experiments. Generating synthetic datasets, developing theoretical models, or proposing hypotheses are not applicable to this work. All the conceptual work, like problem formulation, scoping, experimental design, and discussion are done by the authors.

Additionally, we used generative AI tools for suggest experimental parameters, create or edit code, edit a research paper to improve readability, and grammar check.

We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure our results can be reproduced.

- Model and training: The canvas construction and rectified-flow objective are described in Section 3.1 and Section 3.3. The full training configuration is in Section 3.4, including the base model (Wan2.2-TI2V-5B), resolution, canvas margin, learning rate, batch size, number of steps, and compute.
- Tasks and data: The task taxonomy is summarized in Table 1. Appendix C gives per-task schemas and construction procedures, covering data sources, the off-the-shelf estimators used for each modality, sampling and filtering criteria, and degradation parameters (Tables 8 and 9).
- Evaluation: Evaluation protocols and baselines are described in Section 4. Camera pose metrics are formally defined in Appendix C.4.

REFERENCES

Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9535–9545, 2024.

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. *arXiv preprint arXiv:2412.07760*, 2024.

Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *NIPS*, 35:25005–25017, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13–23, 2025.

Jingxi Chen, Brandon Y Feng, Haoming Cai, Tianfu Wang, Levi Burner, Dehao Yuan, Cornelia Fermuller, Christopher A Metzler, and Yiannis Aloimonos. Repurposing pre-trained video diffusion models for event-based video interpolation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12456–12466, 2025a.

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 Computer Vision and Pattern Recognition Conference*, pp. 22831–22840, 2025b.

Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Dingdong WANG, Hongwei Yi, Xihui Liu, Hengshuang Zhao, Yu Liu, et al. Wan-move: Motion-controllable video generation via latent trajectory guidance. *Advances in Neural Information Processing Systems*, 38:404–432, 2026.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3816–3830, 2021.

Zheng Gu, Shiyuan Yang, Jing Liao, Jing Huo, and Yang Gao. Analogist: Out-of-the-box visual in-context learning with image diffusion model. *ACM Transactions on Graphics (TOG)*, 43(4): 1–15, 2024.

Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, et al. Alltracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5253–5262, 2025.

Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 327–340, 2001.

Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2989–2998, 2023.

Wonjoon Jin, Qi Dai, Chong Luo, Seung-Hwan Baek, and Sunghyun Cho. Flovd: Optical flow meets video diffusion model for enhanced camera-controlled video synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2040–2049, 2025.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Hyunsoo Kim, Donghyun Kim, and Suhyun Kim. Difference inversion: Interpolate and isolate the difference with token consistency for image analogy generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18250–18259, 2025.

Taewoo Kim, Yujeong Chae, Hyun-Kurl Jang, and Kuk-Jin Yoon. Event-based video frame interpolation with cross-modal asymmetric bidirectional motion fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18032–18042, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Bolin Lai, Felix Juefei-Xu, Miao Liu, Xiaoliang Dai, Nikhil Mehta, Chenguang Zhu, Zeyi Huang, James M Rehg, Sangmin Lee, Ning Zhang, et al. Unleashing in-context learning of autoregressive models for few-shot image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18346–18357, 2025.

Zhong-Yu Li, Ruoyi Du, Juncheng Yan, Le Zhuo, Zhen Li, Peng Gao, Zhanyu Ma, and Ming-Ming Cheng. Visualcloze: A universal image generation framework via visual in-context learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18969–18979, 2025.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pp. 100–114, 2022a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b.

Yawen Luo, Xiaoyu Shi, Jianhong Bai, Menghan Xia, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Camclonemaster: Enabling reference-based camera control for video generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–10, 2025.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 11048–11064, 2022.

Thomas W Mitchel, Hyunwoo Ryu, and Vincent Sitzmann. True self-supervised novel view synthesis is transferable. *arXiv preprint arXiv:2510.13063*, 2025.

Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: Image editing via image prompting. *Advances in Neural Information Processing Systems*, 36:9598–9613, 2023.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8298–8319, 2023.

Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. *Advances in neural information processing systems*, 28, 2015.

Robert A Rescorla. A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. *Classical conditioning, Current research and theory*, 2:64–69, 1972.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10912–10922, 2021.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 2339–2352, 2021.

Ashutosh Srivastava, Tarun Ram Menta, Abhinav Java, Avadhoot Gorakh Jadhav, Silky Singh, Surgan Jandial, and Balaji Krishnamurthy. Reedit: Multimodal exemplar-based image editing. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 929–939, 2025.

Adéla Šubrtová, Michal Lukáč, Jan Čech, David Futschik, Eli Shechtman, and Daniel Sỳkora. Diffusion image analogies. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–10, 2023.

Yihong Sun, Hao Zhou, Liangzhe Yuan, Jennifer J Sun, Yandong Li, Xuhui Jia, Hartwig Adam, Bharath Hariharan, Long Zhao, and Ting Liu. Video creation by demonstration. *arXiv preprint arXiv:2412.09551*, 2024.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023.

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.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025b.

Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, pp. 6830–6839, 2023a.

Yihan Wang and Jia Deng. Waft: Warping-alone field transforms for optical flow. *arXiv preprint arXiv:2506.21526*, 2025.

Zhendong Wang, Yifan Jiang, Yadong Lu, Pengcheng He, Weizhu Chen, Zhangyang Wang, Mingyuan Zhou, et al. In-context learning unlocked for diffusion models. *Advances in Neural Information Processing Systems*, 36:8542–8562, 2023b.

Jiarui Xu, Yossi Gandelsman, Amir Bar, Jianwei Yang, Jianfeng Gao, Trevor Darrell, and Xiaolong Wang. Improv: Inpainting-based multimodal prompting for computer vision tasks. *arXiv preprint arXiv:2312.01771*, 2023.

Jiahao Zhang, Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Instruct me more! random prompting for visual in-context learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2597–2606, 2024.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025a.

Zhengbo Zhang, Yuxi Zhou, Duo Peng, Joo-Hwee Lim, Zhigang Tu, De Wen Soh, and Lin Geng Foo. Visual prompting for one-shot controllable video editing without inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7784–7794, 2025b.

Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2063–2073, 2025c.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. Pmlr, 2021.

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. Cami2v: Camera-controlled image-to-video diffusion model. *arXiv preprint arXiv:2410.15957*, 2024.

Appendices

A More Experiments

A.1 Visualization

We provide qualitative rollouts generated by **ViGeo** across several challenging scenarios: depth-flow dual estimation, flow-guided video generation, and camera view transfer in Figure 5. These visualizations demonstrate the model's ability to maintain spatiotemporal consistency across diverse modalities on a single unified canvas.

A.2 Camera Control

In addition to Table 4, we provide a more detailed comparison between XFactor (Mitchel et al., 2025) and **ViGeo** in Table 5. Here, we consider Reconstruction Metrics along with the Covisible Metrics, which calculate the metrics on pixels that are co-visible (estimated using VGGT (Wang et al., 2025b)) from both poses of the query input and query target. In both cases, **ViGeo** obtains superior performance.

Table 5: Quantitative comparison of our method on camera view transfer. The **best** results are bolded.

			Full Frame			Covisible	
Task	**Dataset**	**Method**	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR ↑**	**SSIM ↑**
Camera View Transfer	BRIDGE	XFactor (ICLR 2026)	13.268	0.502	0.604	14.279	0.524
		ViGeo	**16.773**	**0.622**	**0.299**	**16.908**	**0.608**
	DROID	XFactor (ICLR 2026)	12.047	0.480	0.666	12.186	0.498
		ViGeo	**14.470**	**0.564**	**0.377**	**14.416**	**0.569**

Table 6: Quantitative comparison of `camera_view_transfer` where the demo and query videos are from different hold-out environments.

	DROID ($N = 99$)		**BRIDGE** ($N = 100$)	
Method	**RotErr ↓**	**TransDir ↓**	**RotErr ↓**	**TransDir ↓**
XFactor (ICLR 2026)	68.92	63.64	39.48	49.12
ViGeo	**31.58**	**18.77**	**27.32**	**22.79**

During training, **ViGeo** leverages the environment context from both the demo and query, and this yields high reconstruction performance (Table 4). A natural next question is to challenge *what would happen if the demo and query videos are from different environments?* This means that the model can no longer leverage the information from the demo environment and has to extract exclusively the camera pose from the demo pair and the environment context from the query input. We construct this experiment by cross-pairing the demos and queries from different environments. As shown in Table 6, our performance worsens compared to the more aligned scenario.

However, we argue that this failure mode is expected and comes naturally with the current task setup. During training time, we do not provide training scenarios where the environment of the demo videos and query videos is not the same, causing the model to build a shortcut towards using the context from both query and demo rows. A future next step is to construct synthetic demo videos that share the same relative pose difference as the query videos to encourage better pose and context disentanglement.

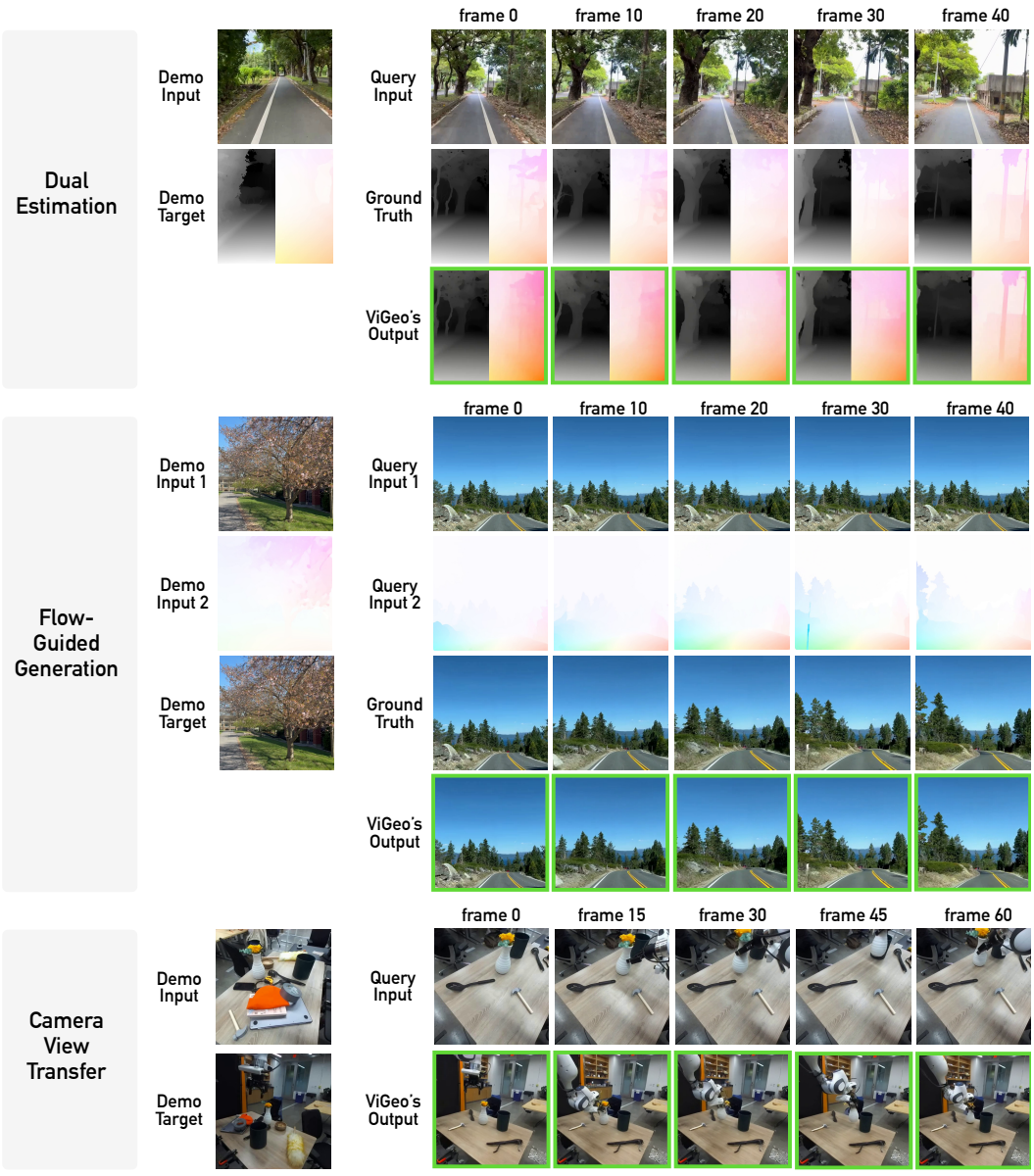


Figure 5: **Qualitative results across diverse video tasks.** We visualize the model's predictions for depth-flow dual estimation, flow-guided generation, and camera view transfer. For the demonstration videos in the context, we display only the initial frame ($t = 0$) for brevity, while the query and generated target quadrants are shown as temporal rollouts. The results highlight **ViGeo**'s ability to synthesize high-fidelity, motion-consistent video while adhering to the geometric and semantic constraints provided by the in-context examples.

A.3 Supplementary Experiment on Task Internalization: Flow → Depth Video Generation

Similar to the depth-to-flow experiment in Section 5.1, we show the result on the held-out flow-to-depth cross-modal video translation task, where a similar task internalization pattern is observed.

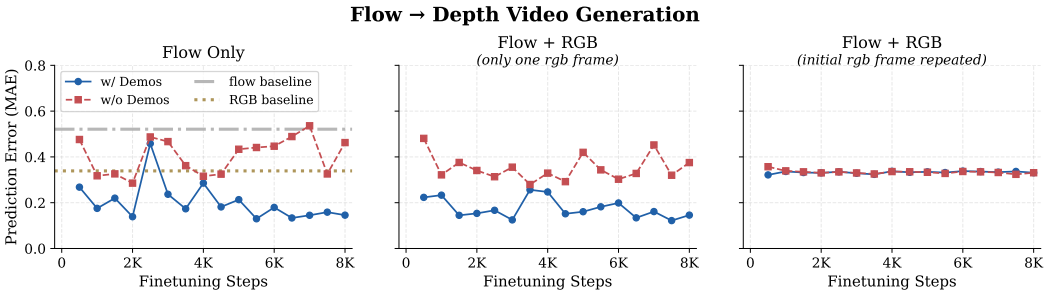


Figure 6: **Query formats that resemble training inputs trigger task internalization.**

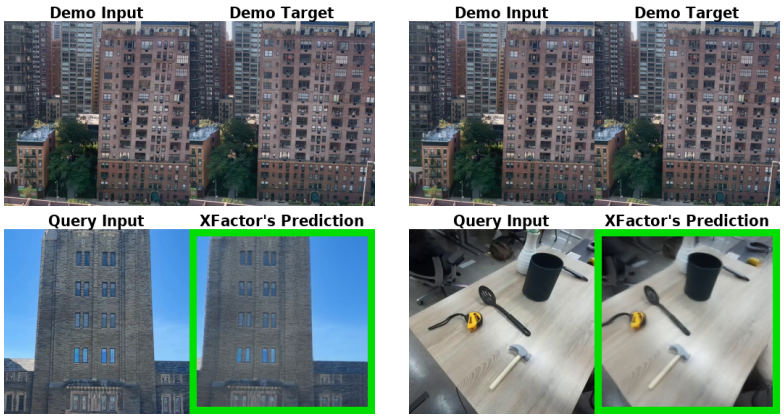
A.4 Failure Mode Analysis of XFactor

The severe performance drop of XFactor (Mitchel et al., 2025) on the `camera_view_transfer` task (Section 4.2) warrants further investigation. To understand why a model trained on extensive camera trajectory data fails in robotic environments, we conduct a qualitative failure mode analysis.

We hypothesize that the extreme camera rotations inherent to robotic manipulation represent a critical out-of-distribution (OOD) shift for XFactor. To test this, we evaluate the model across four demo-query pair combinations using two distinct domains: a general outdoor scene (representative of standard trajectory datasets like DL3DV) and an indoor robotic manipulation setup (captured via left and right side cameras observing a Franka robot, a common configuration in the DROID dataset (Khazatsky et al., 2024)). The task requires XFactor to observe the relative camera transformation in the demo pair (demo input to demo target) and apply that exact transformation to the query input.

The results, illustrated in Figure 7, are consistent with this hypothesis. When the demo transformation is sourced from the standard outdoor dataset (Figure 7a, b), XFactor successfully applies the motion to both outdoor and indoor query images. However, when the demo transformation is sourced from the indoor robotic cameras (Figure 7c, d), the failure appears to be driven by the camera motion rather than by the visual appearance of robotic scenes.

We attribute this failure to a gap between training and test rotation distributions. XFactor is trained on large scene- and object-level video datasets in which human camera operators rarely induce extreme roll or steep pitch. In contrast, the robotic demo used in Figures 7c and d involves a relative rotation of roughly 80° yaw, 75° pitch, and 30° roll, a combination that is likely underrepresented in XFactor's training data.



(a) Setup: Demo: Standard → Query: Standard

(b) Setup: Demo: Standard → Query: Robotic

(c) Setup: Demo: Robotic → Query: Standard

(d) Setup: Demo: Robotic → Query: Robotic

Figure 7: **Qualitative failure mode analysis of XFactor.** We construct a 2x2 grid of cross-domain demo and query pairs. While XFactor easily extracts and applies standard, human-captured camera motions (a, b), it completely collapses when forced to process the extreme roll, pitch, and yaw typical of robotic side-camera setups (c, d).

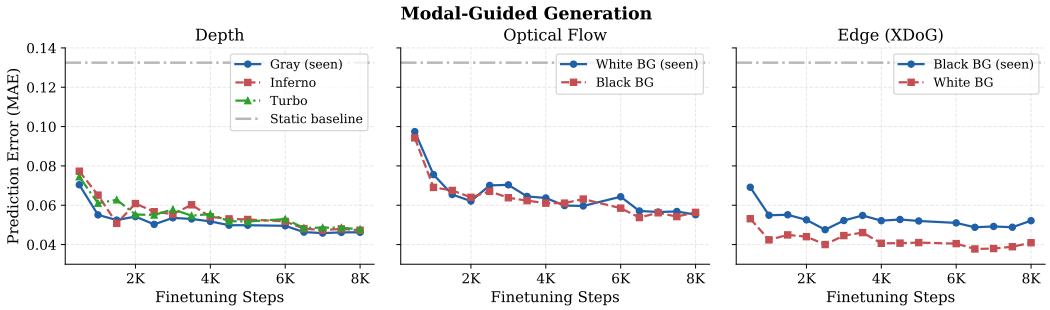


Figure 8: **Robust perception across modality shifts: in-distribution vs. held-out guidance.** Per-checkpoint MAE for modal-guided RGB generation under varying prompt styles. **VIGEO** successfully consumes out-of-distribution renderings as structural guidance. Left: Edge-guided generation works comparably well with seen (Black BG) and unseen (White BG) edge maps. Middle: Depth-guided generation maintains low error when prompted with unseen Inferno and Turbo colormaps, closely tracking the seen grayscale baseline. Right: Optical flow-guided generation adapts seamlessly to unseen black backgrounds. For reference, the 'Static baseline' represents the MAE obtained by trivially repeating the first frame of the ground-truth video.

B A DEEP DIVE TO VIDEO ANALOGY

B.1 ADDITIONAL RESULT FOR TASK INTERNALIZATION

Table 7: Quantitative effect of task internalization under the repeated first-frame hint (F_0 repeated), and its mitigation. Both directions are held out from training; the finetuning data contains neither depth nor optical flow. **Best** results are bolded. n=55 sequences per cell.

	Depth-to-Flow		Flow-to-Depth	
Method	**MAE** ↓	**SSIM** ↑	**MAE** ↓	**SSIM** ↑
VIGEO (base)	0.375	0.335	0.333	0.257
VIGEO (+ finetuning)	**0.142**	**0.818**	**0.108**	**0.793**

Under the repeated first-frame hint, the base model produces an RGB copy rather than the demanded modality, giving an MAE between their outputs and targets of 0.375 and SSIM of 0.335 on depth-to-flow and 0.333/0.257 on flow-to-depth. After finetuning, error (against their target) falls by 62% and 68% respectively, to 0.142 and 0.108, while SSIM rises to 0.818 and 0.793.

B.2 SEEING THROUGH THE STYLE: *Perceptual Invariance*

We ask: *is the model's ability to comprehend input prompts brittle to visual styling shifts?*

Setup. To evaluate this, we test the model on modal-guided video generation (predicting an RGB video from its first frame and a modal guidance video) but deliberately perturb the rendering style of the guidance. Specifically, we prompt the model with modalities in visual conventions it has never seen: depth maps rendered in Inferno and Turbo colormaps (instead of the training-standard grayscale), optical flow on a black background (instead of the training-standard Middlebury white), and edge maps on a white background (instead of the training-standard black). We then measure the prediction error (MAE) of the resulting generated RGB videos with the groundtruth video.

Findings. The empirical results demonstrate perceptual robustness (Figure 8). Despite the drastic pixel-level domain shifts in the guidance videos, the model consistently executes the correct structural generation. For depth-guided generation, the prediction error for unseen Inferno and Turbo prompts closely tracks the baseline grayscale performance. Similarly, optical flow guidance remains robust when inverted to a black background, and edge-guided generation succeeds regardless of the background polarity. We characterize this capability as perceptual invariance.

Implications. The model's perceptual robustness provides deeper insight into how video analogies are processed within the canvas. We hypothesize that during the encoding phase, the model decouples high-frequency spatiotemporal structure (e.g., motion boundaries, geometric gradients) from low-frequency semantic styling (the absolute RGB colormap). It can essentially "see through" the color palette to extract the underlying geometric intent. For in-context learning frameworks, this implies that input guidance can afford to be highly diverse and out-of-distribution.

C TASK TAXONOMY

C.1 CAMERA POSE

C.1.1 TRAINING TASKS.

We organise camera-pose training into two task families.

Camera motion transfer (1): In `camera_motion_transfer`, the model is asked to generate a continuous video sequence along a target camera trajectory given (a) a single static frame to anchor appearance and (b) a demonstration of the same trajectory on a different scene. We curate this from two complementary sources. Layout is 2×2. (i)CamCloneMaster (Luo et al., 2025): each demo and query row pairs a frozen first frame (left cell) with the corresponding full video (right cell); the demo and query rows are drawn from different scenes that share the same Unreal Engine 5-rendered camera trajectory, so the model must transfer the demonstrated motion to the query's appearance. (ii) SpatialVID (Wang et al., 2025a): we precompute a per-window descriptor of the camera path from the dataset's COLMAP poses and match windows that follow approximately the same trajectory across different real-world scenes. The matched pair is then assembled into the same layout.

Camera reangling (2): The model is asked to synthesise the same scene as observed from a target camera placement B given a video at an initial placement A and a demonstration of the $A \to B$ correspondence. We use synchronised multi-camera real-world recordings from BridgeData-v2 (Walke et al., 2023), DROID (Khazatsky et al., 2024), and synthetic synchronised recordings from SynCamMaster (Bai et al., 2024). (i) `camera_view_transfer`: the demo row shows the same scene from two camera placements side-by-side as `[demo@A | demo@B]`; the query row shows another scene from the same pair of placements as `[query@A | query@B]`, where the right cell is the target the model must generate. Layout: 2×2. (ii) `repose_hinted`: we expand the pair task to a 2×3 grid by inserting a frozen first-frame at placement B as a static "hint" between the two views, giving the layout `[A | hint(`$B[0]$`) | B]` per row. The hint anchors appearance at the target placement so the model only has to recover the temporal motion, and provides a graded curriculum from the easier hinted variant to the harder unhinted `camera_view_transfer`.

C.1.2 EVALUATION TASKS.

For evaluation, we consider `camera_view_transfer` and `camera_motion_transfer`. For `camera_view_transfer`, we measure generalisation to unseen environments, we hold out a small set of physical environments at the source level and exclude every scene, session, and trajectory belonging to them from training. From Bridge we hold out two environments, `toysink2_bww` and `tabletop_dark_wood`; from DROID we hold out five buildings, `Bayes - Kitchen 3`, `AHG 1st floor kitchen`, `Bayes - ICMS`, `Bayes - Kitchen Level 1`, and `Smith Hall 121`. The held-out evaluation set therefore measures whether the model can reangle a query scene captured in a physical environment it has never seen during training. For `camera_motion_transfer`, we use the 55 evaluation subset data and 100 drone subset data from DL3DV (Ling et al., 2024).

C.2 MULTIMODAL

Table 8 summarizes the multimodal task taxonomy for both the 20 training tasks and 3 testing tasks.

C.2.1 TRAINING TASKS.

Training data is curated from three complementary sources: SpatialVID (Wang et al., 2025a), SA-V (Kirillov et al., 2023), and Hypersim (Roberts et al., 2021). Per-pixel modalities for SpatialVID are produced by off-the-shelf estimators: Video Depth Anything (Chen et al., 2025b) for depth, WAFT (Wang & Deng, 2025) for optical flow, OneFormer (Jain et al., 2023) for semantic segmentation, DSINE (Bae & Davison, 2024) for surface normals, and AllTracker (Harley et al., 2025) for point trajectories.

SpatialVID — modality estimation (3): The model is asked to recover a pixel-level modality from RGB. Layout: 2×2 `[RGB | modality] × 2 rows`, with demo and query drawn from different scenes. We include `est_depth`, `est_normal`, and `est_flow`.

Table 8: **Per-task schema for Modal-Guided Generation.** The demo and query rows share the same cell schema; the *rightmost cell* is the model's target (present in the demo, absent and predicted in the query). **Convention:** 🎥 (red film icon) = *RGB video*; · 🎥 (gray film icon, prefixed with the modality name) = *modality video*. Analogously, 🖼 = *RGB image* (single frame) and · 🖼 = *modality image*.

Task	Row Schema	Notes
Training tasks		
`est_depth`	[🎥 | depth 🎥]	
`est_normal`	[🎥 | normal 🎥]	
`est_semantic`	[🎥 | semantic 🎥]	per-pixel semantic class (NYU-40 palette).
`est_flow`	[🎥 | flow 🎥]	
`est_point`	[🎥 | points 🖼* | 🎥†]	*static dots on a black image frame, indicating the initial points to be tracked. †RGB video with point tracks overlaid.
`multimodal_quad`	[🎥 | quad 🎥*]	*nested 2×2 modality grid with four modalities each occupying a quadrant
`gen_depth_guided`	[🖼 | depth 🎥 | 🎥]	—
`gen_normal_guided`	[🖼 | normal 🎥 | 🎥]	—
`gen_edge_guided`	[🖼 | edge 🎥 | 🎥]	—
`gen_position_guided`	[🖼 | XYZ 🎥* | 🎥]	*a video where each pixel's color encodes the 3D location of the surface it shows.
`complete_seg`	[🎥* | 🎥]	*selected objects are masked (black) throughout the video except in *one keyframe* per object where each is revealed; the model must reconstruct the full appearance and trajectory of every masked object from that single visible frame.
`track_seg`	[🎥 | hint 🖼* | masklet 🎥†]	*an image where the target object is hinted with cross or scribbles. †an RGB video overlaid with the target object's contour or mask on every frame.
Held-out evaluation tasks		
`gen_flow_guided`	[🖼 | flow viz 🎥 | 🎥]	—
`gen_point_guided`	[🖼 | points 🎥* | 🎥]	* moving dots on a black frame.
`gen_event_guided`	[🖼* | event viz 🎥 | 🎥]	*the first and last images are given for video interpolation.

SpatialVID — modality-guided RGB generation (3): The model is asked to synthesise an RGB video given a frozen first frame (for appearance) and a per-frame modality video (for geometry/motion). Layout: 2×3 `[frozen first frame | modality video | RGB video] × 2 rows`. We include `gen_depth_guided`, `gen_normal_guided`, and `gen_edge_guided`, where the edge variant computes Sobel/XDoG/Canny edges on the fly with the method drawn per-sample from a fixed mixture.

SpatialVID — multimodal quad (1): A 2×2 layout where the right cell is itself a nested 2×2 grid containing four modalities of the same scene (RGB, depth, normal, flow), each rendered as a 240×240 block. The task forces the model to consume and reproduce multiple modalities of the same scene jointly, and exposes the cross-modality consistency the demo and query rows must share. We refer to it as `multimodal_quad`.

SpatialVID — point estimation (1): In `est_point`, the model is asked to track a sparse set of points across an RGB video. We sample $N \in \{5, 10, 20, 40\}$ trackable points per video using AllTracker, with semantic-aware filtering (visibility >0.7, confidence >0.7, in-frame fraction $\geq 80\%$, sky/road/earth excluded via OneFormer's ADE20K labels, person/car preferred when available) so that the chosen points lie on physically meaningful, persistent surfaces rather than on the sky or distant background. The points are rendered as small colored dots (radius 8–12, jittered colors). Layout:

Table 9: **Per-task schema for Video Manipulation.** Demo and query rows share the same cell schema; the *rightmost cell* is the model's target. Demo and query use independently sampled degradation parameters, so the model must recover the operation from the demo row rather than memorize a fixed setting. Held-out tasks test new operation families or unseen degradations.

Task	Row schema	Notes
Training tasks		
`zoom-in_static`	[full | crop]	Static zoom-in to the same region
`zoom-in_animated`	[full | dolly-in]	Animated zoom-in to the same region
`zoom-out_static`	[crop | full]	—
`zoom-out_animated`	[crop | dolly-out]	—
`change_params`	[orig | degraded]	Samples a tuning curve over time and applies it to the original image; variants include saturation, temperature, tone, and chromatic magnitude
`grid_warp`	[warped | orig]	shared Gaussian-bump field
`shuffle_strip_solve`	[shuffled | orig]	1D strips, per-row permutation
`shuffle_strips_apply`	[orig | shuffled]	1D strips, shared permutation
`shuffle_temporal`	[shuffled | orig] / [orig | shuffled]	restore + apply variants
Held-out evaluation tasks		
`3d_inpaint`	[masked | orig]	10–20 black space-time cubes as masks
`3d_outpaint`	[border-blk | orig]	15–30 black space-time cubes as masks; the $(1-\text{mask})$ region is blacked out
`deblur_spatial`	[blurred | sharp]	spatial blur
`deblur_motion`	[blurred | sharp]	motion-blur kernel
`deblur_temporal`	[blurred | sharp]	temporal averaging
`temporal_interp_easy`	[N frames | full]	16 random frames given for interpolation
`temporal_interp_hard`	[N frames | full]	8 random frames given for interpolation
`jigsaw_solve`	[shuffled-2D | orig]	2D tile grid (train was 1D strips)
`jigsaw_apply`	[orig | shuffled-2D]	2D tile grid, shared perm
`low_light`	[dark+noise | orig]	γ-darken + Gaussian noise
`temporal_flip`	[reversed | orig]	global temporal symmetry
`horizontal_flip`	[mirrored | orig]	global spatial symmetry

2×3 `[RGB video | static initial dots on black | dots overlaid on RGB video] × 2 rows.`

Hypersim — indoor multimodal (6): Hypersim provides diverse, per-pixel modalities, including RGB, depth, surface normal in camera and world frames, semantic, world-space position, and instance) on synthetic indoor scenes. We sample 21 keyframes per (scene, camera) and repeat each keyframe four times to yield 82 frames. We define four 2×2 tasks: `est_depth`; `est_normal` (with camera- and world-frame variants); `est_semantic` (NYU-40 palette); and `complete_seg`, in which we pick 4–5 instance-segmented objects that are visible in at least 18 of the 21 keyframes and render each object unmasked in only one keyframe and mask the objects in all other frames to black, requiring the model to reconstruct the full video from sparse single-frame appearance observations. We additionally define two 2×3-layout tasks: `gen_position_guided`

(frozen first-frame world-space XYZ rendered as RGB serves as guidance for the full position video; and gen_depth_guided.

SA-V — video segmentation (1): SA-V provides ∼190K masklet annotations over real-world video. We use a 2×3 layout for track_seg (left: RGB video; mid: black video whose first frame carries a hint of the masklet, one of {coloured semi-transparent mask, contour outline, three cross markers, bounding box, scribble} drawn per-sample; right: the masklet rendered as a coloured overlay on every frame, the model's target).

C.2.2 Evaluation tasks.

We evaluate the model on two held-out modality-guided RGB-generation tasks built from the SpatialVID source pool, using the same 2×3 layout as the corresponding training tasks but with modalities the model has *never* been trained to consume as guidance: (i) gen_flow_guided: the middle column is a per-frame optical-flow visualisation produced by WAFT, and the model must synthesise the corresponding RGB video; (ii) gen_point_guided: the middle column shows $N = 16$ moving points (from AllTracker, with the same semantic-aware filtering used at training time) on black, and the model must produce the matching RGB video where the points implicitly anchor a sparse, deterministic motion field. (iii) gen_event_guided: the left video is sparse RGB frame videos; the middle column is a event camera data visualization, and the model must synthesise the corresponding RGB video.

For gen_flow_guided, we compare with Go-With-The-Flow (Burgert et al., 2025) and FloVD (Jin et al., 2025). FloVD is a generative model based on CogVideoX that contains a flow synthesis module and a flow-conditioned video synthesis module (FVSM). We use its FVSM module for evaluation.

For gen_event_guided, we compare with CBMNet (Kim et al., 2023) and RE-VDM (Chen et al., 2025a), both taking first and end frame video and raw event camera data (CBMNet gets binned grids) for video frame interpolation. For our model, it takes in two boundary anchor frames, identical to the frames the event baselines are given, with all intermediate frames blanked. Specifically, we prepare the input video by "placing frame 0 at idx 0, and frame 12 at idx 12; while other 79 frames are black".

C.3 Video Manipulation

Table 9 summarize the video manipulation taxonomy for both the training and testing tasks.

C.3.1 Training tasks.

We consider three categories of tasks and enumerate in total 9 tasks.

Zoom tasks (4): We consider four variants spanning the cross product of {zoom-in, zoom-out} × {static, animated}: zoom-in_static, zoom-in_animated, zoom-out_static, zoom-out_animated, all sourced from the 4K (3840×2160) data from DL3DV (Ling et al., 2024). For every scene we deterministically sample two nested square regions in the source frame: an outer "wide" square of side $S \in [720, 1920]$ px (i.e. up to 4× the close-up size, capped by the frame's shorter dimension) and an inner 480×480 "close-up" bounding box placed at a random offset inside the wide square. Both the outer square and the inner bounding box are *shared* between the demo and query rows, so the model must localise the close-up region by comparing the two rows. The two views are rendered as 480×480 blocks: the wide view is the outer square downsampled to 480×480, and the close-up view is the inner bounding box at native pixel scale. In the *static* variants both rows display these two views side by side as still cells, ordered [full | crop] for zoom-in and [crop | full] for zoom-out, with no temporal zoom motion. In the *animated* variants one of the two cells is replaced by a synthetic dolly-zoom: at every frame we recompute the crop window size, smoothly sweeping it from the outer square to the inner bounding box (zoom-in) or the reverse (zoom-out), and resize back to 480×480.

Low-level image tasks (1): We consider a change_params task in which low-level image attributes are altered with a smooth time-varying parameter that is shared between the demo and query rows, so the model must recover the underlying control signal from pixel differences alone. The

control trajectory $p(t) \in [0, 1]$, $t = 1, \ldots, 82$, is generated as follows: we draw 2–3 random anchor values uniformly from $\{\frac{1}{20}, \frac{2}{20}, \ldots, \frac{19}{20}\}$, prepend the two endpoints $\{0, 1\}$ to guarantee that both extremes are visited, randomly shuffle the combined list to determine the visit order, then concatenate equal-length segments interpolating between consecutive anchors using one of four randomly chosen easing modes (linear, ease-in, ease-out, ease-in-out). Each task maps $p(t)$ to a different image-domain parameter: (i) saturation, a vectorised RGB saturation factor in $[0, 2]$ blending each frame with its luminance map; (ii) temperature, a warm/cool shift implemented by per-frame R/B channel offsets up to ± 40; (iii) tone, a per-frame gamma correction with $\gamma \in [0.4, 2.8]$ applied via cached LUTs; (iv) focus, a Gaussian blur whose kernel size sweeps from 1 to 51 pixels along the trajectory; and (v) chromatic aberration, per-frame integer translations of the R and B channels by up to ± 20 px along a smooth 2D path, simulating chromatic aberration as a controllable channel-shift effect. For all five, the same trajectory is used for both rows, so the demo row implicitly defines the time-varying parameter that the query row must match.

Spatiotemporal correspondence tasks (4): These tasks geometrically or temporally permute the input and require the model to either invert the permutation (*restore*) or replicate it (*apply*). We consider: (i) `grid_warp`, which composes 2–6 Gaussian "bumps" on a 720×720 canvas with sliding, rotational, or pulsing behaviours and produces a smooth shared spatial deformation field; (ii) `shuffle_strip_solve`, which slices each 480×480 frame into 3–6 horizontal *or* vertical strips and applies an independent non-identity permutation per row in a `[shuffled | original]` layout, so the model must learn to unshuffle by attending to the demo row; (iii) `shuffle_strips_apply`, which uses the *same* strip permutation for both rows in a `[original | shuffled]` layout, so the model must learn to apply the permutation demonstrated in the demo row; and (iv) `shuffle_temporal`, which permutes frames along the time axis with two hash-selected sub-variants: *restore* (independent shuffle per row) and *apply* (shared shuffle across rows). We note that `shuffle_strip_solve`/`shuffle_strips_apply` differ from the classical Jigsaw pretext task in that our strips are one-dimensional bands (horizontal or vertical only), whereas Jigsaw permutes a 2D tile grid, and our setup explicitly provides a paired demo–query row rather than a single-image classification target.

C.3.2 Evaluation tasks.

We use the 55 scenes from the DL3DV evaluation split and apply 15 video-processing tasks grouped into five categories. All evaluation tasks share the same 2×2 grid layout `[degraded | original] × 2 rows`: each row pairs a degraded input (left) with the clean target (right), and the demo and query rows use independently sampled degradation parameters so that the model recovers the operation from the demo row rather than memorising a fixed setting.

Inpainting (2): Tasks that mask out parts of the video and require the model to fill them in. (i) `3d_inpaint`, 10–20 black space-time cubes of size 50–150 px and 10–40 frames placed at random (x, y, t) offsets, producing time-varying occlusions; (ii) `3d_outpaint`, the inverse of `3d_inpaint`: only 15–30 space-time cubes show the original RGB content while everything else is black, so the model must reconstruct the full video from sparse 3D "windows".

Deblurring (3): Tasks that degrade the input with a blur and require the model to recover sharp content. (i) `deblur_spatial`, a per-scene-randomised spatial blur drawn uniformly from {Gaussian, box, median} with kernel size 11–27 px; (ii) `deblur_motion`, a directional motion blur using a linear streak kernel of size 15–33 px at a uniformly sampled angle in $[0, 180^\circ)$; (iii) `deblur_temporal`, a temporal blur produced by averaging each frame with its ± 4-frame neighbours (window size 9, edges truncated), simulating long-exposure or low-frame-rate capture.

Temporal interpolation (2): Tasks that delete most frames and require the model to interpolate the missing ones. We sparsify the input by keeping exactly N randomly chosen frames intact and zeroing out the rest, with two difficulty levels controlled by N: (i) `temporal_interp_easy` (N=16); (ii) `temporal_interp_hard` (N=8); The set of visible frame indices is sampled independently per row.

Spatial jigsaw (2): Tasks that permute a 2×2 or 3×3 tile grid of each frame. (i) `jigsaw_solve`, where each row is shuffled with an independently sampled non-identity permutation in a `[shuffled | original]` layout, so the model must invert the demonstrated permutation; (ii) `jigsaw_apply`, where both rows share the *same* permutation in an `[original | shuffled]`

layout, so the model must replicate the demonstrated permutation. Unlike the strip-shuffle training tasks (which act on 1D bands), these jigsaw tasks act on a true 2D tile grid and serve as the canonical out-of-distribution counterpart.

Photometric & symmetry (3): Three additional tasks covering low-level photometric correction and global symmetry operations. (i) `low_light`, a low-light enhancement task that gamma-darkens each frame with $\gamma \in [0.15, 0.30]$ (effective exponent $1/\gamma \in [3.3, 6.7]$) and adds Gaussian noise with $\sigma \in [10, 30]$; (ii) `temporal_flip`, which reverses the playback order of the input video, requiring the model to play the scene backwards; (iii) `horizontal_flip`, which mirrors each frame left-to-right, requiring the model to produce the left-right reflected video.

C.4 Camera Pose Metrics

We follow the formulation in CamI2V (Zheng et al., 2024). Given a generated video of S frames, we extract camera extrinsics using VGGT (Wang et al., 2025b) and compare against the ground-truth camera trajectory. Both predicted and ground-truth trajectories are first converted to camera-to-world (c2w) poses, made relative to the first frame (setting frame 0 as identity), and independently scale-normalized by dividing all translations by the distance to the furthest camera from the origin:

$$\hat{\mathbf{t}}_i = \frac{\mathbf{t}_i}{\max_j \|\mathbf{t}_j\|_2}, \quad i = 1, \ldots, S-1, \tag{2}$$

where $\mathbf{t}_i$ denotes the translation of the i-th frame relative to frame 0. This normalization is applied independently to both GT and predicted trajectories, making the metrics scale-invariant.

Rotation Error (RotErr). The sum of geodesic (angular) distances between predicted and ground-truth rotations:

$$\text{RotErr} = \sum_{i=0}^{S-1} \arccos\left(\frac{\text{tr}(\mathbf{R}_i^{\text{est}} \mathbf{R}_i^{\text{gt}\top}) - 1}{2}\right), \tag{3}$$

where $\mathbf{R}_i^{\text{est}}, \mathbf{R}_i^{\text{gt}} \in SO(3)$ are the predicted and ground-truth rotation matrices at frame i, and the argument of $\arccos$ is clipped to $[-1, 1]$.

Translation Error (TransErr). The sum of ℓ_2 distances between predicted and ground-truth translations after scale normalization:

$$\text{TransErr} = \sum_{i=0}^{S-1} \left\| \hat{\mathbf{t}}_i^{\text{est}} - \hat{\mathbf{t}}_i^{\text{gt}} \right\|_2. \tag{4}$$

Camera Motion Error (CamErr). The sum of Frobenius norms of the 3×4 pose matrix differences:

$$\text{CamErr} = \sum_{i=0}^{S-1} \left\| [\mathbf{R}_i^{\text{est}} \mid \hat{\mathbf{t}}_i^{\text{est}}] - [\mathbf{R}_i^{\text{gt}} \mid \hat{\mathbf{t}}_i^{\text{gt}}] \right\|_F. \tag{5}$$

`Camera_view_transfer`. For the camera view task, we treat the input image (pose A) and the generated output image (target pose B) as a two-frame sequence (S=2). Ground-truth extrinsics are obtained by running VGGT on the two real images from the demo pair. Predicted extrinsics are obtained by running VGGT on the query input image and the first frame of the generated output. We then apply the same RotErr, TransErr, and CamErr formulation with S=2. Since frame 0 is canonicalized to identity, only the relative pose of the generated view contributes to the error.

`Camera_motion_transfer`. For the camera motion task, the ground-truth is the full S=81 frame trajectory from the reference video. Predicted extrinsics are extracted from the generated video using VGGT. Since all metrics are sums over S frames, values are not directly comparable between the two tasks.

D Limitations

As the first work to explore video analogies as a unified framework for diverse tasks, we acknowledge several current limitations. We thoroughly detail these in Section 4.3, which discusses test tasks our

model struggles to perform (such as flipping and deblurring) , and Section 5, which characterizes bounds on generalization like task internalization.

Beyond these specific failure modes, our framework opens several promising avenues for future work. First, *incorporating text conditioning*. In its current form, our model relies entirely on visual prompts; future work could investigate how integrating text-based task and video prompts might enhance the performance and generative quality of video analogies.

Second, *expanding task diversity*. Despite our best efforts, the diversity of our training tasks remains bounded. As noted in Section 4 and Section 5, increasing task variety and training the model to generate a wider array of visual appearances are integral steps toward making the system more robust and generalizable.

Finally, *exploring diverse visual modalities*. Future research could extend this approach to other complex data structures, such as spectrograms, sonograms, 3D point clouds, sinograms, or volumetric 3D images (e.g., CT and MRI scans).