LEGO: A LIFTING-FREE APPROACH FOR EXOCENTRIC-TO-EGOCENTRIC VIDEO GENERATION

Anonymous authors
Paper under double-blind review

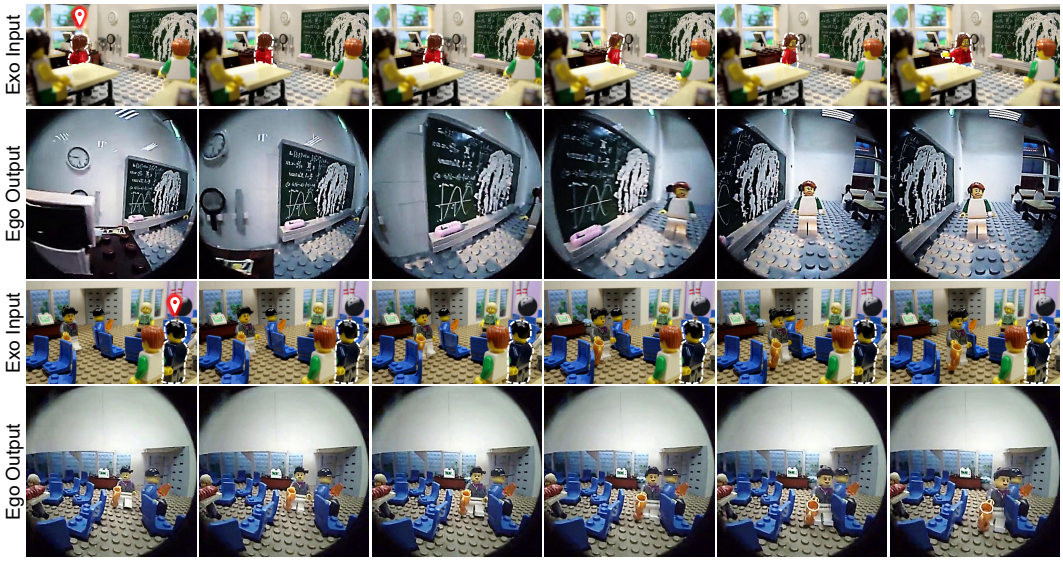

Figure 1: Egocentric videos generated by LEGO from real-world exocentric footage, viewed from the perspective of the LEGO minifigure marked by the dotted outline.

ABSTRACT

Generating an egocentric video from a single exocentric recording is a challenging case of novel view synthesis, as the two cameras share little overlap and much of the target view is unobserved. Current state-of-the-art methods reconstruct the scene explicitly by estimating depth, lifting the video into a point cloud, and re-rendering it from the egocentric camera to condition a video diffusion model. This deterministic mapping assigns each pixel to a single reprojected location, which preserves texture but translates depth errors into misplaced content. We ask what a video diffusion model should receive as its condition and propose a lifting-free answer: a learned view synthesizer, an LVSM-style transformer fine-tuned to render the egocentric view directly without depth, point clouds, or reprojection, resolving cross-view correspondence internally. In contrast, its probabilistic mapping averages each region over candidate source locations according to a learned correspondence distribution, preserving structure while fine texture is averaged away. We argue that this trade-off suits a diffusion generator, whose denoising training excels at restoring detail, so an effective condition should prioritize structural alignment over sharpness. This distribution's concentration also yields a per-region confidence, used both to mask low-confidence regions and to guide the generator toward high-confidence areas during early layout-forming denoising steps. Our approach consistently outperforms the state-of-the-art explicit pipeline and generalizes to other datasets without retraining. The synthesizer thus supplies view structure, and the diffusion model its detail.

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.

1 INTRODUCTION

We study exocentric-to-egocentric video generation: given a single third-person recording of a scene and a target egocentric camera trajectory, the goal is to synthesize the video a camera following this trajectory would capture. The task matters wherever first-person data is scarce and third-person footage abundant (Grauman et al., 2024), and it is demanding: the two views share little content, so much of the target must be synthesized rather than copied. Existing methods condition implicitly on exocentric features (Luo et al., 2024b; Cheng et al., 2024); assume privileged inputs such as surrounding cameras (Liu et al., 2024a), rich multimodal observations (Park et al., 2026), or a ground-truth egocentric first frame (Xu et al., 2025); or reconstruct the scene explicitly. The current state of the art (Kang et al., 2026) takes the explicit route, conditioning a video diffusion model (Wan et al., 2025) on a point-cloud render obtained by lifting the exocentric video with monocular depth and replaying it from the egocentric trajectory. This route inherits the errors of the estimated depth: the point-cloud render leaves much of the egocentric frame uncovered, and depth errors misplace the content it does cover.

We present LEGO (**L**ifting-free Exocentric-to-**EGO**centric generation; Fig. 2), a lifting-free alternative that replaces the reconstruction pipeline with a learned view synthesizer that renders the egocentric view directly. An LVSM-style transformer (Jin et al., 2025) is fine-tuned to map the exocentric frame and the target camera rays to the egocentric image, so cross-view correspondence is learned from the rendering objective alone. The frozen synthesizer's render replaces the point-cloud render in the generator's conditioning input; no depth estimation, point cloud, or reprojection is involved.

The design follows from the paper's central question: what should a video diffusion model be given as its condition? The two renders fail in opposite ways. The point-cloud render is deterministic: each pixel is assigned a single reprojected location, so its errors are misplaced content. The synthesizer's render is probabilistic: each region averages over candidate source locations according to a learned correspondence distribution, so its errors are blur. These failures are not equally costly: blur is the corruption that denoising training teaches the generator to undo, and diffusion models routinely recover sharp detail from blurred inputs (Saharia et al., 2023; Ho et al., 2022; Meng et al., 2022), whereas rough or misplaced conditions have needed dedicated mechanisms to relax them (Liu et al., 2024b; Seo et al., 2024). We therefore argue that a condition should be judged by alignment first and sharpness second, since the generator supplies the second far more readily than the first: the condition should say where things are, and the generator will decide how they look. The correspondence distribution supplies a further signal for this: its concentration at each region is a confidence, and we mask from the conditioning those regions where it is low.

Alignment first, detail second also describes the sampling trajectory itself: layout is decided in the early, high-noise steps and detail emerges later, so structural guidance is most useful early. We therefore reuse the same confidence at inference, guiding the early denoising steps toward the render where it is high and leaving the rest unconstrained. This requires no additional training; we call it Adaptive Rendering Consistency (ARC, Section 3.3).

On Ego-Exo4D, the synthesizer's render outperforms the point-cloud render as a condition, ARC at inference alone outperforms the depth-derived attention bias that prior work trains into the generator, and the full system outperforms the state-of-the-art explicit pipeline on seen and unseen scenes; the same weights also generalize to EgoHumans and Nymeria without retraining.

Our contributions are: **(1)** LEGO, a lifting-free exocentric-to-egocentric video generation framework that conditions a video diffusion model on the confidence-gated render of a learned view synthesizer instead of a point-cloud reconstruction; **(2)** evidence that a structurally aligned but soft condition serves the generator better than a sharp but misplaced one; **(3)** ARC, training-free guidance that reuses the synthesizer's confidence to steer the early denoising steps toward its render; and **(4)** state-of-the-art results on Ego-Exo4D and generalization to EgoHumans and Nymeria without retraining.

2 RELATED WORK

Exocentric-to-egocentric video generation. Early systems relied on privileged inputs: surrounding cameras (Liu et al., 2024a), rich multimodal exocentric observations (Park et al., 2026), or the ground-truth egocentric first frame (Xu et al., 2025). Others condition implicitly on exocentric features (Luo et al., 2024b; Cheng et al., 2024). The state of the art, EgoX (Kang et al., 2026), takes

a single exocentric video and reconstructs the scene explicitly: depth from a video pose engine (Huang et al., 2025a) lifts the frames into a point cloud, which is re-rendered along the egocentric trajectory to condition a video diffusion model (Wan et al., 2025), and a depth-derived attention bias is trained into the generator. Concurrent and adjacent work extends this explicit line (Gu et al., 2026; Song et al., 2026; Mahdi et al., 2026), reverses the direction or transfers exocentric knowledge (Luo et al., 2024a; Xiu et al., 2026; Zhang et al., 2026), or pairs the two views for recognition (Sigurdsson et al., 2018; Li et al., 2021). LEGO keeps the single-video setting and diffusion backbone of EgoX, but replaces the reconstruction with a learned view synthesizer and the trained attention bias with training-free guidance.

Camera-controlled video generation. Camera poses can steer video diffusion directly, through extrinsics or Plücker embeddings injected into the backbone (He et al., 2025; Wang et al., 2024; Bahmani et al., 2025) or by re-rendering a video along a new trajectory (Bai et al., 2025a). For larger viewpoint changes, systems lift the video into explicit 3D and condition generation on a render of the target trajectory (Yu et al., 2025b;a; Ren et al., 2025; Huang et al., 2025b). EgoX follows this recipe in a setting where the view change is extreme and estimated depth is least reliable; LEGO takes the same pose inputs but is lifting-free.

Learned view synthesis and emergent correspondence. Novel view synthesis has moved from per-scene optimization to feed-forward models that predict radiance fields, 3D Gaussians, or triplanes in a single pass (Yu et al., 2021; Liu et al., 2023; Hong et al., 2024; Charatan et al., 2024; Chen et al., 2024); LVSM (Jin et al., 2025) maps posed input tokens to target-ray tokens with a plain transformer. Rendering supervision is known to teach geometry: photometric view synthesis losses made monocular depth learnable without ground truth (Garg et al., 2016; Zhou et al., 2017). Self-supervised and diffusion features likewise acquire correspondence they were never supervised for (Caron et al., 2021; Amir et al., 2022; Tang et al., 2023; Zhang et al., 2023; Luo et al., 2023). We rely on the same effect in a synthesizer fine-tuned for the exocentric-to-egocentric view change: its render serves as the generator's condition, and the concentration of its correspondence distribution as a per-region confidence (Section 3.2).

What a diffusion model needs from its condition. Conditional diffusion models recover fine detail from coarse inputs: super-resolution and cascaded generation upsample low-resolution images, and blurring the condition during training helps rather than hurts (Saharia et al., 2023; Ho et al., 2022); SDEdit synthesizes realistic images from stroke layouts (Meng et al., 2022); GeNVS produces sharp novel views from a volume-rendered feature image (Chan et al., 2023); and blur can serve as the forward corruption a diffusion model learns to invert (Rissanen et al., 2023; Hoogeboom & Salimans, 2023), with coarse structure decided at high noise and detail at low noise (Choi et al., 2022; Rissanen et al., 2023). Conversely, precise conditions are neither required nor always safe: layout-level depth suffices for depth-conditioned control (Bhat et al., 2024), control must be relaxed where a rough condition is wrong (Liu et al., 2024b), and generation from a depth-based reprojection must learn where to trust it (Seo et al., 2024). We read these results as favoring a condition that is aligned before it is sharp.

Training-free guidance in diffusion models. Pretrained diffusion models can be steered at sampling time: ILVR (Choi et al., 2021), RePaint (Lugmayr et al., 2022), and Blended Diffusion (Avrahami et al., 2022) blend a reference into the intermediate sample; DDRM (Kawar et al., 2022) and DDNM (Wang et al., 2023) project each step onto measurement-consistent solutions; DPS (Chung et al., 2023), MCG (Chung et al., 2022), and FreeDoM (Yu et al., 2023) apply guidance gradients. All assume the reliable content and its mask are given; ARC obtains both from the synthesizer and acts only in the early steps.

3 METHOD

We follow the input setting of EgoX (Kang et al., 2026): a single exocentric video with its camera calibration, and a target egocentric camera trajectory along which the video is to be generated. All experiments use the same image-to-video diffusion backbone and canvas formulation as EgoX, in which the exocentric and egocentric views are concatenated along the width and the egocentric half of the conditioning latent carries the condition for the target view. Prior work fills this conditioning input with a point-cloud render and, in addition, trains a depth-derived bias into the cross-view

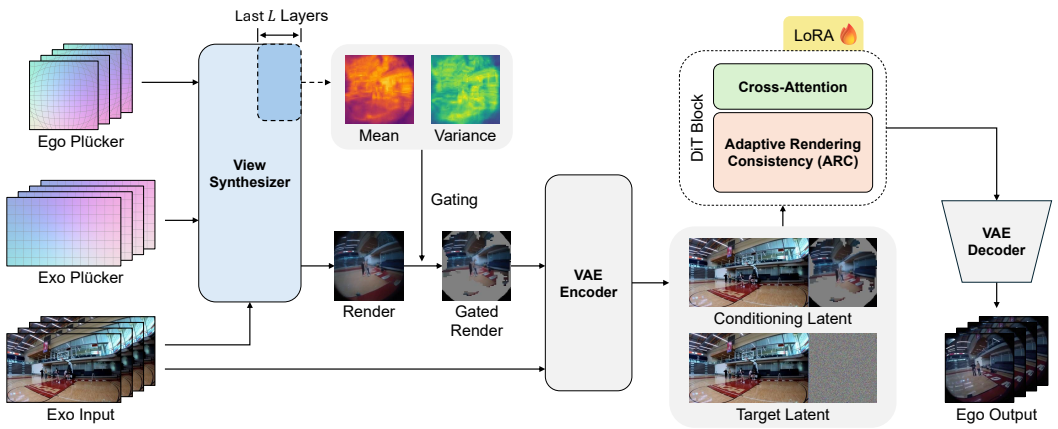

Figure 2: Overview of LEGO. The frozen synthesizer F renders the egocentric view once per latent frame and exposes a confidence map. The render, gated to gray where confidence is low, fills the egocentric half of the conditioning latent; the confidence also drives ARC during the early denoising steps. The generator is unchanged.

attention. LEGO fills it with the render of a frozen view synthesizer (Section 3.1), gates the render by the confidence of the synthesizer's correspondence distribution (Section 3.2), and uses the same confidence to guide the early denoising steps toward the render (Section 3.3). The pipeline is lifting-free: no depth is estimated and no point cloud is built at any stage. We do not modify the generator. We change the condition it receives and how it is sampled (Fig. 2).

3.1 Learned View Synthesizer

We train a view synthesizer F, an LVSM-style transformer (Jin et al., 2025), to render the egocentric view directly. Geometry enters F only through per-pixel ray maps: a calibrated camera assigns each pixel $\mathbf{u}$ the Plücker embedding $\mathbf{r}(\mathbf{u}) = (\mathbf{d}(\mathbf{u}), \mathbf{o} \times \mathbf{d}(\mathbf{u})) \in \mathbb{R}^6$ of its viewing ray, with unit direction $\mathbf{d}$ and camera center $\mathbf{o}$. All poses are expressed in the exocentric camera frame, so the moment $\mathbf{o} \times \mathbf{d}(\mathbf{u})$ of each egocentric ray carries the metric baseline between the two cameras. For frame t, F takes the exocentric frame and the two ray maps and predicts the egocentric frame,

$$\hat{I}_t^{\text{ego}} = F(I_t^{\text{exo}},\, \mathbf{r}^{\text{exo}},\, \mathbf{r}_t^{\text{ego}})\,, \tag{1}$$

and is trained with an ℓ_2 and a perceptual loss against the ground-truth egocentric frame. The target frame is the only supervision; no depth or correspondence labels are used.

Correspondence distribution. Inside F, each egocentric query $\mathbf{u}$ attends over exocentric locations $\mathbf{v}$ with normalized weights $w(\mathbf{u}\to\mathbf{v})$. Averaging these weights over heads and over the last L layers gives, for every target location, a distribution over source locations,

$$p_{\mathbf{u}}(\mathbf{v}) = \underset{\text{heads, last } L \text{ layers}}{\text{mean}} w(\mathbf{u}\to\mathbf{v}), \tag{2}$$

which we call the correspondence distribution. At the image level, it makes the difference between the two pipelines explicit:

$$\underbrace{\hat{I}^{\text{ego}}(\mathbf{u}) \approx \sum_{\mathbf{v}} p_{\mathbf{u}}(\mathbf{v})\, I^{\text{exo}}(\mathbf{v})}_{\text{learned (probabilistic)}} \qquad \underbrace{\hat{I}_{\pi}^{\text{ego}}(\mathbf{u}) = I^{\text{exo}}(\pi_d(\mathbf{u}))}_{\text{explicit (deterministic)}}, \tag{3}$$

where $\pi_d(\mathbf{u})$ denotes the exocentric source location corresponding to the ego-view pixel $\mathbf{u}$, as determined by geometric reprojection using the estimated depth d. The approximation on the left reflects that F mixes features across layers rather than averaging pixels directly.

Two kinds of error. The two mappings fail differently. The explicit mapping assigns each target pixel a single source pixel, which keeps texture sharp but translates a depth error directly into

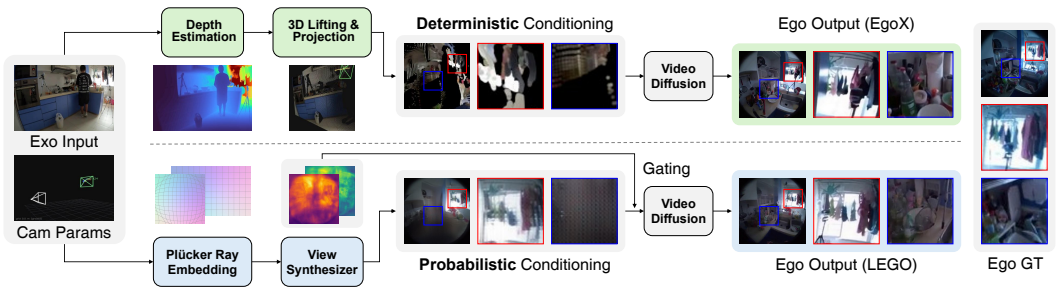

Figure 3: Condition generation. EgoX estimates depth, lifts the exocentric video into a point cloud, and re-renders it along the egocentric trajectory; LEGO renders the egocentric view directly with a frozen synthesizer.

misplaced content; in statistical terms, its error is a bias. The learned mapping averages over the candidate sources in $p_{\mathbf{u}}$, which blurs texture wherever the distribution is spread out but keeps content close to its correct place; its error is a variance. Denoising training makes the generator well suited to restoring detail (Saharia et al., 2023; Ho et al., 2022; Chan et al., 2023), so blur is the failure it is best equipped to correct, whereas conditions that are sharp but misplaced have required dedicated mechanisms to relax or override them (Liu et al., 2024b; Seo et al., 2024). This is why we condition on F's render rather than on the point-cloud render; Section 4 tests the choice, and Fig. 3 contrasts the two pipelines.

Adapting to the egocentric camera. The published LVSM checkpoint, which we call the untuned synthesizer, is trained to interpolate between nearby pinhole views, whereas our target is an extrapolation to a fisheye headset camera that shares little overlap with the source. Because every camera-specific quantity enters F through the ray map, no architectural change is needed: we invert the calibrated fisheye projection per pixel to obtain the target rays, and fine-tune all of its weights on exocentric-egocentric pairs, supervised by the raw fisheye recordings. Supervising on real recordings also teaches F where the lens does not see, which the ray map alone cannot express (Appendix C.3). The fine-tuning is cheap, 10,000 steps or about 3.5 hours on four NVIDIA H200 GPUs, and F is frozen afterwards.

3.2 Confidence-Gated Conditioning

The render of F replaces the point-cloud render in the egocentric half of the conditioning latent. The two differ in one further respect: the point-cloud render leaves a hole wherever the geometry offers no support, whereas the probabilistic mapping fills every location, including those with no exocentric evidence, and so cannot abstain on its own. The concentration of $p_{\mathbf{u}}$ supplies the missing signal. We define the confidence of a target location as the probability mass of its k most likely sources, and set locations below a threshold to neutral gray:

$$c(\mathbf{u}) = \sum_{\mathbf{v} \in \mathrm{TopK}_k(p_{\mathbf{u}})} p_{\mathbf{u}}(\mathbf{v}), \qquad C(\mathbf{u}) = \begin{cases} \hat{I}^{\mathrm{ego}}(\mathbf{u}), & c(\mathbf{u}) \geq \tau, \\ \gamma, & \text{otherwise}, \end{cases} \tag{4}$$

where $c(\mathbf{u}) \in [0, 1]$, τ is the gating threshold, and γ is a neutral gray that the VAE maps near zero. The gated render C is encoded into the generator's latent space and used as the conditioning input. Since F is frozen, its renders and confidence maps are precomputed once per clip, and the generator is otherwise unchanged, so the comparison in Section 4 isolates the effect of the conditioning input alone.

3.3 Adaptive Rendering Consistency

The gated render enters the generator as its conditioning input. Prior work additionally steers the generator with a depth-derived bias on cross-view attention, trained into the model. We instead act on the denoising trajectory at inference time: during the early steps, in which the layout of the sample is determined (Choi et al., 2022; Rissanen et al., 2023), the generator's current estimate of

Table 1: Comparison on Ego-Exo4D. Unmarked rows are quoted from Kang et al. (2026); †released EgoX checkpoint rerun under our scoring stack. TF, MS, and DD are VBench temporal flickering, motion smoothness, and dynamic degree. Best in bold.

		Image Criteria				Video Criteria			
	Method	PSNR↑	SSIM↑	LPIPS↓	CLIP-I↑	FVD↓	TF↑	MS↑	DD↑
Seen	Exo2Ego-V	14.53	0.384	0.569	0.774	622.47	0.960	0.966	0.985
	TrajectoryCrafter	13.05	0.375	0.606	0.780	546.09	0.960	0.980	0.947
	Wan Fun Control	12.25	0.463	0.617	0.810	595.07	0.968	0.980	0.901
	Wan VACE	12.95	0.413	0.626	0.829	508.69	**0.989**	**0.994**	0.673
	EgoX	16.05	0.556	0.498	0.896	184.47	0.977	0.990	0.974
	EgoX†	15.57	0.556	0.466	0.903	197.95	0.985	0.993	**0.990**
	LEGO	**20.28**	**0.689**	**0.310**	**0.925**	**155.56**	**0.989**	**0.994**	0.987
Unseen	Exo2Ego-V	12.70	0.439	0.597	0.679	1283.50	0.971	0.976	0.978
	TrajectoryCrafter	12.24	0.297	0.619	0.778	821.71	0.966	0.984	0.944
	Wan Fun Control	13.59	0.439	0.604	0.799	968.78	0.971	0.985	0.944
	Wan VACE	12.17	0.345	0.638	0.820	1045.45	**0.995**	**0.996**	0.427
	EgoX	14.38	0.457	0.552	0.877	440.64	0.981	0.992	0.989
	EgoX†	13.53	0.437	0.542	0.886	455.06	0.986	0.994	0.990
	LEGO	**15.91**	**0.505**	**0.504**	**0.897**	**404.77**	0.990	0.995	**1.000**

the clean video is moved toward the render in proportion to the confidence, and the remaining steps are left unconstrained. We call this *Adaptive Rendering Consistency* (ARC).

The generator is trained with flow matching: a noisy latent $z_\sigma = (1-\sigma)\, z_0 + \sigma\, \epsilon$ is formed at noise level $\sigma \in [0, 1]$, the transformer predicts the velocity $v \approx \epsilon - z_0$, and sampling follows a decreasing schedule $\sigma_0 = 1 > \sigma_1 > \cdots > \sigma_n = 0$. At step i the solver moves from z_i to z_{i+1}, and the same velocity implies an estimate of the clean video at the current step:

$$z_{i+1} = z_i + (\sigma_{i+1} - \sigma_i)\, v, \qquad \hat{z}_0 = z_i - \sigma_i\, v. \tag{5}$$

Since $\sigma_{i+1} < \sigma_i$, each step moves the sample a fixed fraction of the way toward $\hat{z}_0$, so the clean estimate sets the direction of every step. ARC acts there: during the first hn steps it replaces the egocentric part of the clean estimate by a confidence-weighted blend with the render, and passes the corresponding velocity to the solver:

$$\hat{z}_0'^{\,\mathrm{ego}} = \left(1 - \mathbf{c}^2\right) \odot \hat{z}_0^{\,\mathrm{ego}} + \mathbf{c}^2 \odot E, \qquad v'^{\,\mathrm{ego}} = \frac{z_i^{\,\mathrm{ego}} - \hat{z}_0'^{\,\mathrm{ego}}}{\sigma_i} = v^{\mathrm{ego}} - \frac{\mathbf{c}^2}{\sigma_i} \odot \left(E - \hat{z}_0^{\,\mathrm{ego}}\right), \tag{6}$$

where E is the ungated render encoded into latent space and $\mathbf{c}$ is the confidence map of Eq. 4, resampled to the latent grid. The second equality follows from $\hat{z}_0 = z_i - \sigma_i v$. The denoising step is then taken exactly as in Eq. 5 with v' in place of v,

$$z_{i+1}^{\,\mathrm{ego}} = z_i^{\,\mathrm{ego}} + (\sigma_{i+1} - \sigma_i)\, v'^{\,\mathrm{ego}}, \tag{7}$$

so the sample is drawn toward the blended estimate instead of toward $\hat{z}_0$ alone: strongly toward the render where confidence is high, and almost unchanged where it is low. The correction is deterministic and adds no noise, and the solver itself is untouched; in practice we use a second-order multistep solver (Zhao et al., 2023), which consumes v' through the same clean estimate. After the first hn steps the trajectory is unconstrained, so the render shapes the layout in the early, high-noise steps while the remaining steps add detail that the render does not contain.

We weight by $\mathbf{c}^2$ rather than $\mathbf{c}$: the squared weight lowers the correction in regions of intermediate confidence, where the render is often blurred, while leaving high-confidence regions near full strength. The ungated render is used because $\mathbf{c}^2$ already suppresses low-confidence regions continuously. The noise level σ_i in Eq. 6 is lower-bounded by a small constant for numerical stability.

4 Experiments

Experimental setup. We train and evaluate on Ego-Exo4D (Grauman et al., 2024) following the protocol of EgoX (Kang et al., 2026). The benchmark has two test splits: *seen*, whose scenes appear

Table 2: Ablation under one recipe. *Top*: what fills the conditioning input (exocentric tokens hidden). *Middle*: cumulative additions up to the full conditioning. *Bottom*: how denoising is driven: GGA retrained in, ARC at inference, and ARC anchored to the point-cloud render with its hole mask as binary confidence. ARC (synthesizer render) is the LEGO row of Table 1. Best in bold.

	Seen				Unseen			
Arm	PSNR↑	LPIPS↓	CLIP-I↑	FVD↓	PSNR↑	LPIPS↓	CLIP-I↑	FVD↓
Conditioning input: what fills it								
No condition	12.30	0.591	0.862	310.00	11.38	0.640	0.838	695.44
Point-cloud render	15.59	0.478	0.895	195.80	13.42	0.595	0.853	549.77
Untuned synthesizer render	15.71	0.503	0.888	212.94	13.56	0.596	0.848	560.51
Synthesizer render	18.24	0.364	0.904	183.40	14.35	0.567	0.860	581.90
Completing the configuration (cumulative)								
+ confidence gating	18.27	0.375	0.904	166.86	14.40	0.566	0.868	502.47
+ exocentric tokens	18.91	0.340	0.921	159.04	14.98	0.515	**0.897**	415.08
Denoising process: how it is driven								
GGA	18.62	0.356	0.915	194.98	15.20	0.525	0.896	429.42
ARC (point-cloud render)	13.96	0.586	0.801	735.50	13.72	0.618	0.795	812.88
ARC (synthesizer render)	**20.28**	**0.310**	**0.925**	**155.56**	**15.91**	**0.504**	**0.897**	**404.77**

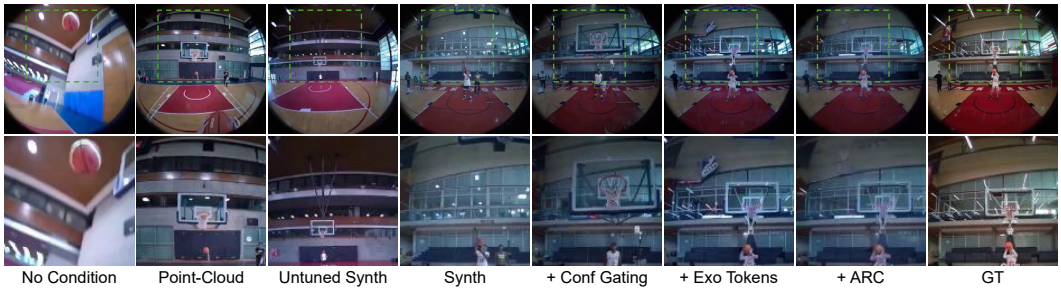

Figure 4: Qualitative ablation on one unseen clip, following the rows of Table 2. The second row shows enlarged crops of the corresponding regions in the first row.

in the training set, and *unseen*, whose scenes do not. PSNR, SSIM, LPIPS, CLIP-I, and FVD are computed on the generated egocentric frames, and the VBench scores for temporal flickering (TF), motion smoothness (MS), and dynamic degree (DD) on the full exocentric–egocentric canvas. We compare with the numbers reported in Kang et al. (2026) and with the released EgoX checkpoint rerun under our evaluation code, denoted EgoX†. For a fair comparison, all ablations use the same backbone, LoRA rank, training schedule, random seed, and camera trajectories. They differ only in the conditioning input and in how denoising is guided. Implementation details, the evaluation protocol, and object-level metrics are given in Appendices A, B.1, and B.3, respectively.

Quantitative evaluation. Table 1 reports the comparison with prior methods on both splits. LEGO outperforms EgoX on all image criteria and FVD, with the largest gains on PSNR and LPIPS. The other baselines perform substantially worse than both. Temporal flickering and motion smoothness are near saturation for all methods, and dynamic degree is comparable.

Ablation Study. Table 2 ablates the conditioning input and the denoising guidance under the same training recipe, and Fig. 4 shows the corresponding generations.

Conditioning input. We first compare what fills the conditioning input, with the exocentric tokens hidden. The point-cloud render improves over an empty condition, and the fine-tuned synthesizer render improves further on both splits. The untuned synthesizer render performs on par with the point-cloud render, which shows that the gain comes from adapting F to the egocentric camera (Section 3.1) rather than from its architecture.

Table 3: Condition quality on the unseen split. ZNCC and DINO co-location cosine measure alignment with the ground truth inside the valid circle; Sharpness is the Laplacian-variance ratio to the ground truth (1 = the recording); Time is seconds per clip on one GPU. Protocol in Appendix B.4.

Condition	ZNCC↑ (patch $p\times p$)			DINO↑	Sharpness↑	Time (s)↓
	p=16	32	64			
Point-cloud render	0.034	0.047	0.059	0.476	**0.97**	69.2
Untuned synthesizer render	0.008	0.016	0.030	0.508	0.50	**4.7**
Synthesizer render	**0.169**	**0.217**	**0.275**	**0.586**	0.28	**4.7**

Gating and exocentric tokens. Confidence gating discards part of the rendered region, yet it lowers FVD on both splits and leaves the other metrics unchanged. Restoring the exocentric tokens brings a further gain on all metrics, as the model can also refer to the source view directly.

Denoising guidance. On top of the full conditioning, ARC improves all metrics without any retraining and yields the final `LEGO` model in Table 1. GGA, the depth-guided attention of EgoX trained into the same model, does not improve over the full conditioning. Applying ARC to the point-cloud render, with its hole mask as a binary confidence, is worse than no guidance at all. ARC reinforces the render it is anchored to, so it requires an aligned render and a confidence map that indicates where the render is reliable.

Alignment versus sharpness. We claim in Section 3.1 that a blurry but geometrically aligned condition is better than a sharp but misaligned one. Table 3 tests this on the unseen split by measuring both properties of each conditioning render against the ground truth. Alignment is measured by zero-mean normalized cross-correlation (ZNCC) and the similarity of DINO features (Siméoni et al., 2025) at the same location. Sharpness is measured by the Laplacian variance relative to the ground truth. The two renders are opposite: the synthesizer render is the blurriest but the best aligned, while the point-cloud render is sharp but poorly aligned. Despite its blur, the synthesizer render conditions the diffusion model better (Table 2). The blur of the condition does not remain in the output: generations from the two renders reach the same sharpness (Appendix C.2), which indicates that the diffusion model restores the details missing from the condition. The synthesizer render is also much cheaper to compute than depth estimation and point-cloud rendering (Table 3).

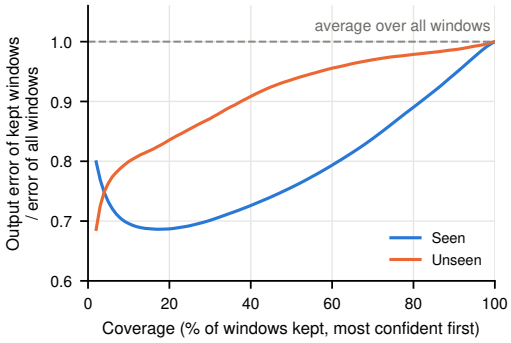

Figure 5: Confidence is informative. Windows of the generated video are sorted by confidence; the curve is the error of the most confident fraction relative to all windows, on Ego-Exo4D. Dashed: the average over all windows.

Confidence. Confidence gating and ARC both assume that the confidence of Eq. 4 indicates where the render is correct. Fig. 5 tests this assumption without committing to a threshold. We sort the windows of the generated videos by confidence and plot the error of the most confident windows against the ground truth as a function of how many are kept. Windows with higher confidence have lower error, well below the average over all windows at every cut-off, on seen and unseen scenes alike. The confidence therefore indicates where the render can be trusted.

Generalization to other datasets. Table 4 evaluates both methods on EgoHumans (Khirodkar et al., 2023) and Nymeria (Ma et al., 2024) with the weights trained on Ego-Exo4D, without any fine-tuning. `LEGO` outperforms EgoX† on both datasets on all metrics. Since F is fine-tuned on Ego-Exo4D, one may suspect that it has overfit to the training environments. This is unlikely, as the new datasets contain entirely different scenes and people. The fine-tuning instead adapts F to the egocentric camera model and to the large view change (Section 3.1), which are shared across datasets. In addition, the gate keeps a similar fraction of the render on the new datasets as on unseen Ego-Exo4D scenes, roughly 15% in each case, indicating that the synthesizer's confidence does not

Table 4: Evaluation on EgoHumans (Khirodkar et al., 2023) and Nymeria (Ma et al., 2024) with the weights trained on Ego-Exo4D only. Each dataset has 200 test clips. Best in bold.

		Image Criteria				Video Criteria			
Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	CLIP-I↑	FVD↓	TF↑	MS↑	DD↑
EgoHumans	EgoX†	13.74	0.460	0.573	0.806	421.83	0.983	0.991	**1.000**
	LEGO	**15.52**	**0.530**	**0.545**	**0.819**	**332.41**	**0.985**	**0.992**	1.000
Nymeria	EgoX†	11.64	0.451	0.637	0.804	296.98	0.984	0.993	**1.000**
	LEGO	**15.23**	**0.534**	**0.561**	**0.821**	**210.27**	**0.988**	**0.994**	1.000

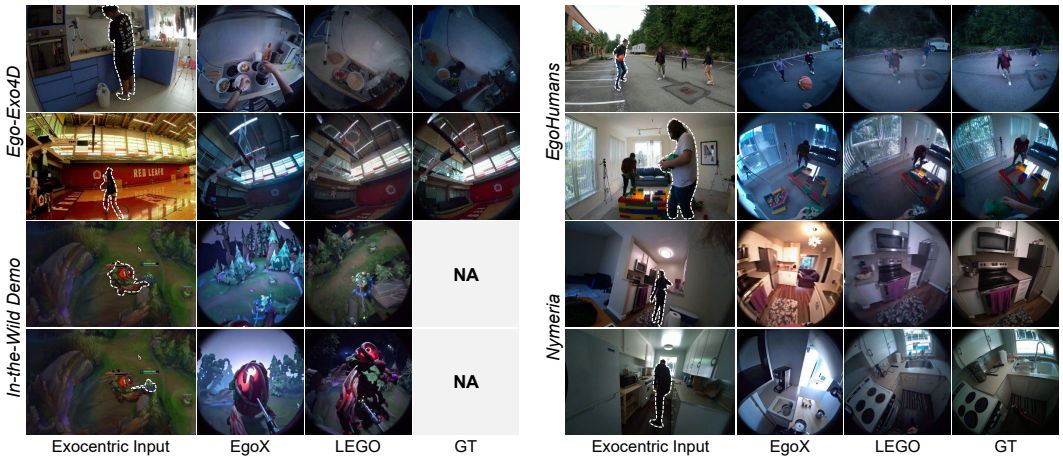

Figure 6: Qualitative comparison between EgoX and LEGO from the viewpoint indicated by the dotted outline. GT is unavailable (NA) for the in-the-wild clips.

depend on scene familiarity. Detailed dataset construction and camera preprocessing for this transfer evaluation are provided in Appendix B.5.

Qualitative evaluation. Fig. 6 compares EgoX and LEGO on the benchmark datasets and on in-the-wild videos. EgoX fails in two typical ways. Where the point cloud provides no support, the render leaves holes, which the diffusion model fills from its own prior, and the layout drifts. Where the estimated depth is wrong, the content is sharp but misplaced, most noticeably for hands and nearby objects. LEGO preserves the layout of the ground truth in both cases, and the diffusion model recovers the fine details that the condition lacks. On in-the-wild videos, for which no ground truth exists, the gap widens: the point-cloud render of EgoX becomes unreliable and its generated views break away from the exocentric video, whereas LEGO keeps them consistent. Fig. 1 further shows LEGO on domains far from the training data, where it produces plausible egocentric views along user-specified camera trajectories.

5 Conclusion

LEGO conditions a video diffusion model on the render of a learned view synthesizer, not a point-cloud reconstruction, on one criterion: a condition should be aligned before it is sharp, because the generator restores detail far more readily than it corrects misplaced content. The design has three practical advantages. It is lifting-free, with no depth estimation, point cloud, or reprojection at any stage. Its confidence comes free from the synthesizer's correspondence distribution and drives both the gating and a training-free guidance that replaces a bias trained into the generator. And the frozen synthesizer enters only through its render and confidence, so the same weights carry over to new scenes and recordings without retraining, and a stronger synthesizer can be dropped in as one appears. The synthesizer supplies the structure of the view, and the diffusion model its detail.

Reproducibility Statement

The training and inference protocol, including every hyperparameter of the synthesizer and the generator and the fixed random seed, is given in Section 4 and Appendix A. Appendix B.1 audits our protocol parameter by parameter against the official stack of the state of the art, and Appendix B.4 specifies the condition quality protocol. Code, configuration files, model weights, the precomputed renders and confidence maps, and the test clip lists of EgoHumans and Nymeria, sufficient to reproduce every reported configuration, will be released upon publication.

Ethics Statement

This work uses three publicly released benchmarks, Ego-Exo4D, EgoHumans, and Nymeria, under their licenses; all were recorded with participant consent, and we collect no new data and run no human study. Generating egocentric video of a person carries privacy and impersonation risks if applied to identifiable individuals outside such consented settings. We will release our models under research-only use terms aligned with the dataset licenses, and we do not release generated videos of identifiable individuals.

The Use of Large Language Models

Large language models were used as a writing assistant for this paper, to draft and edit text under the authors' direction. The research questions, the method, the experiments, and all reported numbers are the authors' own, and every model-assisted passage was reviewed by the authors, who take full responsibility for the content.

References

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep ViT features as dense visual descriptors. In *Proc. ECCV Workshop*, 2022.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proc. CVPR*, 2022.

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. AC3D: Analyzing and improving 3D camera control in video diffusion transformers. In *Proc. CVPR*, 2025.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, and Di Zhang. ReCamMaster: Camera-controlled generative rendering from a single video. In *Proc. ICCV*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Shariq Farooq Bhat, Niloy J. Mitra, and Peter Wonka. LooseControl: Lifting ControlNet for generalized depth conditioning. In *Proc. ACM SIGGRAPH*, 2024.

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. *arXiv preprint arXiv:2311.15127*, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021.

Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3D-aware diffusion models. In *Proc. ICCV*, 2023.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *Proc. CVPR*, 2024.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D gaussian splatting from sparse multi-view images. In *Proc. ECCV*, 2024.

Feng Cheng, Mi Luo, Huiyu Wang, Alex Dimakis, Lorenzo Torresani, Gedas Bertasius, and Kristen Grauman. 4Diff: 3D-aware diffusion model for third-to-first viewpoint translation. In *Proc. ECCV*, 2024.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *Proc. ICCV*, 2021.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proc. CVPR*, 2022.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *Proc. NeurIPS*, 2022.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *Proc. ICLR*, 2023.

Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proc. ECCV*, 2016.

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in Fréchet video distance. In *Proc. CVPR*, 2024.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *Proc. CVPR*, 2024.

Qiao Gu, Lingni Ma, Adam W. Harley, Richard Newcombe, Florian Shkurti, and Julian Straub. E^3C: Video generation with 3D environmental memory and ego-exo human pose control. *arXiv preprint arXiv:2605.26316*, 2026.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *Proc. ICLR*, 2025.

Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *Proc. ICLR*, 2024.

Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. In *Proc. ICLR*, 2023.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. ViPE: Video pose engine for 3D geometric perception. *arXiv preprint arXiv:2508.10934*, 2025a.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson W. H. Lau, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3D scene generation. In *Proc. ACM SIGGRAPH Asia*, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proc. CVPR*, 2024.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A large view synthesis model with minimal 3D inductive bias. In *Proc. ICLR*, 2025.

Taewoong Kang, Kinam Kim, Dohyeon Kim, Minho Park, Junha Hyung, and Jaegul Choo. EgoX: Egocentric video generation from a single exocentric video. In *Proc. CVPR*, 2026.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Proc. NeurIPS*, 2022.

Rawal Khirodkar, Aayush Bansal, Lingni Ma, Richard Newcombe, Minh Vo, and Kris Kitani. Ego-Humans: An egocentric 3D multi-human benchmark. In *Proc. ICCV*, 2023.

Jiatong Li, Leo Liang, Linghe Kong, and Yulun Zhang. FreqForcing: Autoregressive long video generation via spectral self-anchoring. *arXiv preprint arXiv:2607.27110*, 2026.

Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *Proc. CVPR*, 2021.

Jia-Wei Liu, Weijia Mao, Zhongcong Xu, Jussi Keppo, and Mike Zheng Shou. Exocentric-to-egocentric video generation. In *Proc. NeurIPS*, 2024a.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *Proc. ICCV*, 2023.

Xiaoyu Liu, Yuxiang Wei, Ming Liu, Xianhui Lin, Peiran Ren, Xuansong Xie, and Wangmeng Zuo. SmartControl: Enhancing ControlNet for handling rough visual conditions. In *Proc. ECCV*, 2024b.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proc. CVPR*, 2022.

Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Proc. NeurIPS*, 2023.

Hongchen Luo, Kai Zhu, Wei Zhai, and Yang Cao. Intention-driven ego-to-exo video generation. *arXiv preprint arXiv:2403.09194*, 2024a.

Mi Luo, Zihui Xue, Alex Dimakis, and Kristen Grauman. Put myself in your shoes: Lifting the egocentric perspective from exocentric videos. In *Proc. ECCV*, 2024b.

Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, Kevin Bailey, David Soriano Fosas, C. Karen Liu, Ziwei Liu, Jakob Engel, Renzo De Nardi, and Richard Newcombe. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *Proc. ECCV*, 2024.

Mohammad Mahdi, Nedko Savov, Danda Pani Paudel, and Luc Van Gool. From synchrony to sequence: Exo-to-ego generation via interpolation. In *Proc. ECCV*, 2026.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *Proc. ICLR*, 2022.

Junho Park, Andrew Sangwoo Ye, and Taein Kwon. EgoWorld: Translating exocentric view to egocentric view using rich exocentric observations. In *Proc. ICLR*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *Proc. ICLR*, 2025.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-informed world-consistent video generation with precise camera control. In *Proc. CVPR*, 2025.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *Proc. ICLR*, 2023.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4), 2023.

Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. GenWarp: Single image to novel views with semantic-preserving generative warping. In *Proc. NeurIPS*, 2024.

Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *Proc. CVPR*, 2018.

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

Quanjian Song, Yiren Song, Kelly Peng, Yuan Gao, and Mike Zheng Shou. WorldWander: Bridging egocentric and exocentric worlds in video generation. In *Proc. ECCV*, 2026.

Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *Proc. NeurIPS*, 2023.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *Proc. ICLR*, 2023.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *Proc. ACM SIGGRAPH*, 2024.

Jingqiao Xiu, Fangzhou Hong, Yicong Li, Mengze Li, Wentao Wang, Sirui Han, Liang Pan, and Ziwei Liu. EgoTwin: Dreaming body and view in first person. In *Proc. ICLR*, 2026.

Jilan Xu, Yifei Huang, Baoqi Pei, Junlin Hou, Qingqiu Li, Guo Chen, Yuejie Zhang, Rui Feng, and Weidi Xie. EgoExo-Gen: Ego-centric video prediction by watching exo-centric videos. In *Proc. ICLR*, 2025.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proc. CVPR*, 2021.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *Proc. ICCV*, 2023.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proc. ICCV*, 2025a.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Haoyu Zhang, Qiaohui Chu, Meng Liu, Haoxiang Shi, Yaowei Wang, and Liqiang Nie. Exo2Ego: Exocentric knowledge guided MLLM for egocentric video understanding. In *Proc. AAAI*, 2026.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable Diffusion complements DINO for zero-shot semantic correspondence. In *Proc. NeurIPS*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, 2018.

Min Zhao, Hongzhou Zhu, Chendong Xiang, Kaiwen Zheng, Chongxuan Li, and Jun Zhu. Identifying and solving conditional image leakage in image-to-video diffusion model. In *Proc. NeurIPS*, 2024.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Proc. NeurIPS*, 2023.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017.

A Implementation Details

A.1 Camera Models

The exocentric camera is treated as a pinhole with the released intrinsics. Ego-Exo4D also ships radial distortion coefficients, but they describe the lens before rectification: applying them raises the reprojection error of the annotated 3D body joints from 4 pixels to 143 pixels at 3840×2160, so we do not use them. The egocentric camera follows the Aria fisheye model, $\theta_d = \theta\,(1 + \sum_i k_i \theta^{2i})$ with small tangential and thin-prism terms, which we invert per pixel by Newton iteration to obtain the target rays. All rays are expressed in the exocentric camera frame in metric units. Queries outside the image circle are not masked, so the synthesizer learns the boundary from the recordings.

A.2 View Synthesizer

F has the architecture of the released scene-level LVSM decoder (Jin et al., 2025): 24 transformer blocks of width 768 with 12 heads and patch size 8. Source tokens carry RGB and the six Plücker channels; target tokens carry the six ray channels only. We initialize from the published checkpoint and fine-tune all weights on Ego-Exo4D at 448×448 egocentric and 448×784 exocentric resolution, with an ℓ_2 loss and an LPIPS loss weighted by 0.5. Training uses AdamW with learning rate 2×10^{-5}, cosine decay with 500 warm-up steps, a global batch size of 16, and 10k steps, about 3.5 hours on 4 H200 GPUs. The confidence of Eq. 4 is read from the attention of the last 8 layers, averaged over heads, with the softmax restricted to exocentric keys, and average-pooled 2×2 to the token grid of the generator (28×28 egocentric queries).

A.3 Generator

The backbone is Wan2.1-I2V-14B at 480P (Wan et al., 2025), fine-tuned with LoRA of rank 256 and α=256 on a $49\times448\times1232$ canvas, the exocentric frame (784 wide) concatenated with the egocentric frame. Training runs for 20k steps with learning rate 2×10^{-5}, bf16, a batch size of 1 per GPU on 4 H200 GPUs, and seed 42; all ablations share this recipe. The backbone is also conditioned on a text prompt, at training and at inference alike, as in EgoX. On Ego-Exo4D we use the prompts released with the benchmark. EgoHumans and Nymeria come with none, so we generate one for every test clip from eight exocentric frames, following the system prompt of EgoX and its two-block format, an `[Exo view]` description of the scene followed by an `[Ego view]` description of the target viewpoint. We replace only the vision-language model, GPT-4o in EgoX, with the open Qwen2.5-VL (Bai et al., 2025b), and match the prompt length to the distribution of the released Ego-Exo4D prompts. At inference both methods receive the same prompt for a given clip. Inference uses 50 denoising steps with classifier-free guidance 5.0. Confidence gating uses k=16 and τ=0.3 on the synthesizer's 56×56 query grid. ARC applies Eq. 6 after each denoising step during the first 80% of the schedule (h=0.8), with the noise level floored at 0.05. The ARC hyperparameters (h and the squared weighting) were chosen on the unseen split, since Ego-Exo4D provides no separate validation split.

B Evaluation Details

B.1 Comparison Protocol

To compare fairly with EgoX (Kang et al., 2026), we match its training and inference configuration wherever the released code specifies it: backbone checkpoint, LoRA rank and alpha, canvas resolution, optimizer, learning rate, precision, batch size, seed, noising and loss masks, sampler, number of denoising steps, guidance scale, and frame count. The GGA ablation uses the official implementation of the geometry-guided attention bias, and we verified that the bias is applied identically at training and inference. Two differences remain. We train for 20k steps with a batch size of 1 per GPU on 4 H200 GPUs for about four days, whereas EgoX trains with the same batch size on 8 H200 GPUs for about one day. Second, since the evaluation code is not public, metrics are our own implementation. Under our implementation (Appendix B.2), the released checkpoint scores 15.57 (16.05) dB PSNR on seen and 13.53 (14.38) dB on unseen scenes, where the numbers in parentheses are those reported in Kang et al. (2026).

B.2 Image and Video Metrics

Since the evaluation code of EgoX (Kang et al., 2026) is not publicly available, we reimplemented all metrics following the descriptions in the paper and calibrated our implementation so that the released EgoX checkpoint approximately reproduces the published numbers (Appendix B.1). Generated and ground-truth videos are compared over the 49 frames of each clip; frame scores are averaged within a clip and clip scores over the split.

Image criteria. PSNR, SSIM, LPIPS, and CLIP-I are computed on the 448×448 egocentric half of the 1232×448 canvas. SSIM uses an 11×11 Gaussian window with $\sigma=1.5$. LPIPS (Zhang et al., 2018) uses the AlexNet backbone, and CLIP-I is the cosine similarity between CLIP ViT-B/32 (Radford et al., 2021) embeddings of the generated and ground-truth frames.

Video criteria. FVD (Unterthiner et al., 2018) is computed with the `cd-fvd` implementation (Ge et al., 2024) on the egocentric half: I3D features of one 16-frame sample per video, resized to 224×224, compared between the generated and ground-truth clips of a split. Temporal flickering, motion smoothness, and dynamic degree are computed with the official VBench code (Huang et al., 2024) on the full canvas, which reproduces the values reported by EgoX (Appendix B.1). Dynamic degree is a per-clip binary decision based on RAFT optical flow; we keep the VBench sampling and count rule and apply the same flow threshold to every method.

Sharpness and chroma. For the analysis of the conditioning renders, sharpness is the variance of the Laplacian response of a frame on a region Ω, reported as the ratio to the ground-truth frame on the same region. Ω is the valid egocentric image circle, or, for point-cloud renders, the rendered pixels eroded by 3 px. Chroma is the mean chroma magnitude inside the valid image circle, reported as the ratio of the generated to the ground-truth video. Both ratios are averaged over the 13 frames aligned with the latent frames ($t=0, 4, \ldots, 48$) and over clips; a value of 1 means the same amount as the ground truth.

B.3 Object-Level Metrics

EgoX (Kang et al., 2026) reports three object-level metrics: *LocErr*, the distance between the bounding-box centers of matched objects in the generated and ground-truth frames; *IoU*, the overlap of their bounding boxes; and *Contour*, the agreement of their shapes. Computing them requires a detection and matching pipeline that is not part of the public release. We therefore fix one protocol, validate it against the published numbers, and apply it unchanged to all methods.

Protocol. Objects are segmented and tracked with SAM 2 (Ravi et al., 2025) on the generated and the ground-truth video independently, using automatic mask generation on the first frame (32 points per side, seed boxes of at least 32 px, mask regions of at least 2000 px^2). Each object crop is embedded with DINOv3 ViT-L (Siméoni et al., 2025), and two objects in the same frame are matched when the cosine similarity of their embeddings is at least 0.9, without one-to-one assignment, as described in EgoX. LocErr and IoU are computed on the matched boxes and averaged per clip and then over clips. For Contour, the literal definition in EgoX is the mask IoU in image coordinates, but it cannot reproduce the published values, which exceed the published box IoU by a wide margin; the boundary F-score after aligning the mask centroids does reproduce them, so we adopt it.

Calibration. EgoX defines the three metrics but not the segmentation, the object features, or the matching threshold used to compute them. We chose these settings by searching over the corresponding options and scoring each combination by the relative error of the released EgoX checkpoint against the six published numbers. Combinations that retain fewer than 30 matched pairs on either split are discarded.

Validation. Under the protocol above, the released checkpoint reproduces the published unseen row almost exactly (LocErr 146.5 against 149.9, IoU 0.091 against 0.092), and `LEGO` outperforms it on all three metrics on both splits (Table 5). The published seen row is not reproduced by any combination in the search: the closest one still leaves a large gap for the released checkpoint (LocErr 87.5 against 61.8, IoU 0.222 against 0.363) and no longer agrees on the unseen row. We therefore quote the published seen row for reference only. The lower block of Table 5 scores both methods

Table 5: Object-level criteria under two protocols. *Top*: our fixed protocol, which reproduces the published unseen row from the released checkpoint. *Bottom*: the sweep combination closest to the published seen row. Published numbers are quoted.

	Seen			Unseen		
System	LocErr↓	IoU↑	Contour↑	LocErr↓	IoU↑	Contour↑
Fixed protocol (ours; validated on the published unseen row)						
EgoX (published)	61.8	0.363	0.546	149.9	0.092	0.481
EgoX (released)†	125.3	0.176	0.622	146.5	0.091	0.572
LEGO	107.7	0.279	0.674	143.3	0.097	0.609
Seen-fitted protocol (closest sweep combination to the published seen row)						
EgoX (released)†	87.5	0.222	0.546	115.4	0.119	0.477
LEGO	65.3	0.434	0.634	116.5	0.124	0.488

under that closest combination, a permissive segmenter with DINOv3-B patch features, one-to-one matching at τ=0.56, and registered mask-IoU contour, where LEGO reaches the level of the published seen row while the released checkpoint does not.

B.4 Condition Quality Metrics

Table 3 compares each conditioning render against the ground-truth egocentric frame on the unseen split, with metrics that depend on neither the synthesizer nor the diffusion model. We evaluate the 13 frames that align with the latent frames of the generator (t=$0, 4, \ldots, 48$).

Evaluation region. The valid circle of a clip is the set of pixels that are lit in any ground-truth frame, eroded by 4 px. For the point-cloud render the region is further restricted to the rendered pixels, eroded by 3 px so that the edges of the holes are excluded.

Metrics. ZNCC is the zero-mean normalized cross-correlation of luminance, computed over non-overlapping $p\times p$ patches in which at least half of the pixels are valid, and averaged over patches, frames, and clips. DINO co-location is the cosine similarity between the DINOv3 ViT-B/16 patch tokens of the render and of the ground truth at the same location, averaged over tokens whose 16×16 window is at least half valid. Sharpness is the ratio of the Laplacian variance of the render to that of the ground truth on the same pixels, and coverage is the fraction of the valid circle that the render fills. Alignment in dB is the PSNR of the render against the ground truth on the same evaluation region.

B.5 Cross-Dataset Evaluation

This subsection describes how the test clips of EgoHumans and Nymeria are selected and how their videos are prepared for evaluation. Both datasets record the egocentric view with Project Aria glasses, so the egocentric camera is the same fisheye model as in Ego-Exo4D.

EgoHumans. This dataset (Khirodkar et al., 2023) records multi-person activities with several exocentric cameras and is not used for training. We select 200 clips of 49 frames, distributed evenly over the seven activity scenarios and drawn from multiple takes per scenario. Within a take, the wearer and the starting frame are sampled at regular intervals with a fixed seed. For each clip we use a single exocentric camera, the one in which the wearer is closest to the image center. The exocentric frames are resized and center-cropped to the canvas resolution, and the egocentric frames are the raw fisheye recordings, as in Ego-Exo4D.

Nymeria. This dataset (Ma et al., 2024) records a participant and an observer, both wearing Aria glasses, and is not used for training. The observer takes the role of the exocentric camera. Since the observer moves, we use only 49-frame windows in which the observer camera is effectively static, with at most 4 cm of translation and 2.5° of rotation. We further require that the participant's head is visible in the observer's view throughout the window, which we check by projecting the participant's camera center into the observer's frame and by detecting a person at the projected location. From the

resulting pool we select 200 clips over 26 sequences, balancing the number of clips per sequence. The observer's fisheye frames are undistorted to a pinhole camera with a focal length of 260 pixels and cropped to the canvas resolution.

Evaluation. Both datasets are scored with the evaluation code on the egocentric frames of the same clips. EgoX is run with the authors' released checkpoint and pipeline, including depth estimation, point-cloud rendering, and geometry-guided attention. Both methods use the same hyperparameters as on Ego-Exo4D.

C Discussions

C.1 Alignment, Sharpness, and Coverage

The synthesizer render is better aligned than the point-cloud render by 4.2 dB inside the valid circle. This advantage remains even when we evaluate alignment only on pixels where the point-cloud render is valid, indicating that the difference is not simply due to the point cloud's missing regions.

High coverage also does not guarantee good alignment. The untuned synthesizer covers 99.6% of the valid circle and is sharper than the fine-tuned synthesizer (0.50 vs. 0.28), yet it has the worst ZNCC alignment of the three. The point-cloud render is locally sharp where geometry is available, but occlusion and depth errors reduce its coverage to 53.4%, compared with 98.1% for the synthesizer render.

C.2 Output Sharpness and Color

Despite the difference in sharpness between the conditioning renders, their generated outputs have similar sharpness. On unseen scenes, the Laplacian-variance ratio to the ground truth is 1.10 for outputs conditioned on the point-cloud render and 1.04 for those conditioned on the synthesizer render. This indicates that output sharpness is not strongly determined by the sharpness of the conditioning render.

Color behaves differently. On unseen scenes, the synthesizer render has a chroma ratio of only 0.41 relative to the ground truth, and the generated output shows a similar reduction, reaching 0.45. In comparison, the output reaches 0.82 when conditioned on the point-cloud render and 1.26 without conditioning. Gating the synthesizer render partially recovers the chroma ratio to 0.62, suggesting that removing unreliable conditioning allows the generator to rely more on its own appearance prior. Thus, unlike sharpness, output color remains more sensitive to the conditioning render.

The mechanism is an asymmetry between the two models. The synthesizer is a regression model, so under correspondence uncertainty it tends toward a conditional mean, in which chroma averages toward gray while luminance structure largely survives. The generator under the same uncertainty samples a plausible mode, and its color prior, acquired at pretraining scale and adapted here only through a low-rank update, is unaffected by the synthesizer's collapse. Such conditioning over-reliance is documented for image-to-video models (Zhao et al., 2024). The standard remedy, for it and for the exposure bias, is to degrade or noise-augment the conditioning during training (Ho et al., 2022; Blattmann et al., 2023). Color drift is itself a known signature of conditioning on model-produced imagery (Li et al., 2026). FVD is particularly sensitive to such per-frame appearance statistics (Ge et al., 2024), which accounts for part of the unseen FVD gap in Table 2.

C.3 Adapting to the Fisheye Camera

The untuned synthesizer does not reliably respect the target camera model, even when given the correct camera rays. On unseen scenes, it fills 95.9% of the region outside the valid fisheye circle with scene content. We measure this behavior using *spill*, defined as the fraction of the region outside the image circle that a signal fills with content.

Simply masking this region would not solve the problem. Inside the valid field of view, the untuned synthesizer is also the worst aligned of the three conditions, trailing the fine-tuned synthesizer by 5.4 dB on unseen scenes and 10.3 dB on seen scenes. Conditioning the generator on this render further reduces performance by 0.8 dB on unseen scenes and 2.5 dB on seen scenes.

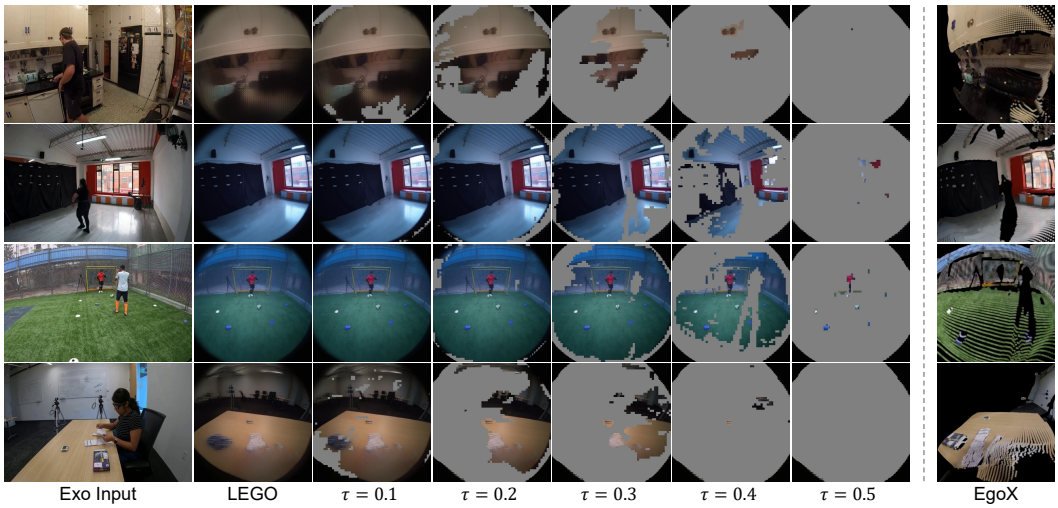

Figure 7: Effect of the gate threshold. The synthesizer render gated at $\tau = 0.1, \ldots, 0.5$, shown between the ungated render and the point-cloud render of Kang et al. (2026).

Table 6: Cost of generating one clip on one H200 GPU, excluding the one-time model load. Generation assumes that the condition is already built, and end-to-end includes it. Best in bold.

	EgoX†	LEGO
Denoiser bias	GGA	ARC
Denoising loop (s)↓	565.3	**540.3**
Generation (s)↓	597.5	**573.2**
End-to-end (s)↓	666.7	**577.9**
Peak VRAM (GiB)↓	69.9	**68.2**

Fine-tuning substantially corrects both behaviors. The adapted synthesizer leaves the region outside the image circle essentially empty, with only 0.1% spill, while the point-cloud render fills 22.4%. Adapting the synthesizer to the target camera is therefore necessary, not optional.

C.4 Confidence Gating

The gate of Eq. 4 restores abstention from the same distribution that produced the render. The confidence is the probability mass of the top k=16 sources, read from the last 8 layers with head averaging. Tokens below τ=0.3 are set to the neutral gray that the VAE maps near zero, which is the same value as an empty conditioning latent (Fig. 7).

Gate statistics (Fig. 5). To test whether confidence tracks render reliability independently of the threshold, we sort 16×16 windows by their mean confidence and plot the luminance error of the most confident fraction. We report errors as (kept, discarded). At τ=0.3, render errors are $(11.6, 16.6)$ on seen scenes and $(25.9, 30.7)$ on unseen scenes; output errors show the same trend, at $(13.1, 21.4)$ and $(26.1, 32.8)$, respectively. On both splits, kept windows consistently have lower render and output errors than discarded windows, indicating that the confidence score tracks both rendering reliability and final generation quality. The render curves have the same shape as the output curves shown in Fig. 5. We report absolute luminance error rather than ZNCC because window-level ZNCC is confounded by texture; within strata of ground-truth texture, kept windows still have higher ZNCC than discarded ones.

C.5 Computational Cost

Table 6 reports the cost of generating one clip on a single H200 GPU, averaged over the 100 clips. Generation covers the denoising loop together with the encoding and decoding around it, and as-



Figure 8: A limitation from F. Where the exocentric frame does not cover the egocentric view, the condition is empty and the generator fills the region from its own prior.

sumes that the condition is already built. End-to-end adds the cost of building the condition. During denoising, the two methods differ only in the bias they apply. GGA builds a dense attention bias over the whole canvas and reapplies it in every transformer block, which adds 252 ms to each forward pass. ARC instead updates the egocentric part of the clean estimate once per step, in 2.7 ms. The larger difference is upstream. Building our condition takes 4.7 s per clip, against 69.2 s for depth estimation and point-cloud rendering (Table 3). Overall, LEGO generates a clip faster than EgoX and uses slightly less memory.

C.6 Limitations

LEGO inherits the limitations of the view synthesizer. F renders each frame independently from a single exocentric frame, so it cannot recover content that the frame does not show (Fig. 8), and it blurs where the correspondence is uncertain. Its colors also collapse on unfamiliar scenes (Section C.2). The pipeline is lifting-free but still requires the camera poses of both views. Since F is frozen and enters only through the conditioning input and the confidence map, a stronger feed-forward view synthesizer can replace it without changing the diffusion model or ARC.

D Additional Qualitative Results

Figures 9 and 10 show further comparisons on the seen and unseen splits of Ego-Exo4D, and Figures 11 and 12 on EgoHumans and Nymeria. Figure 13 additionally shows results on in-the-wild exocentric footage. Each method runs its own checkpoint, the released one for EgoX and ours for LEGO, and both receive the same camera poses, the same text prompt, and the same random seed.

Across the examples, the two methods fail in the way described in Section 4. EgoX either fills a region that its point-cloud render leaves empty, which makes the layout drift from the ground truth, or reproduces content sharply at the wrong position, which is most visible for the hands and for objects close to the camera. LEGO keeps the scene where the ground truth places it and lets the diffusion model add the details that its condition does not carry. The same holds on EgoHumans and Nymeria. The remaining errors of LEGO are softer details and, on unseen scenes, less saturated colors (Section C.2).

Figure 13 applies the same weights to in-the-wild film footage, for which no ground truth exists. The in-the-wild clips have neither camera calibration nor an egocentric recording, so we place a virtual head camera on one person and aim it at another, with the exocentric focal length fixed to the training value and the camera poses set by hand. EgoX places the wearer's own head in the view in the first two examples and collapses to a nearly black frame in the third, whereas LEGO keeps the wearer out of frame and holds the layout across frames.

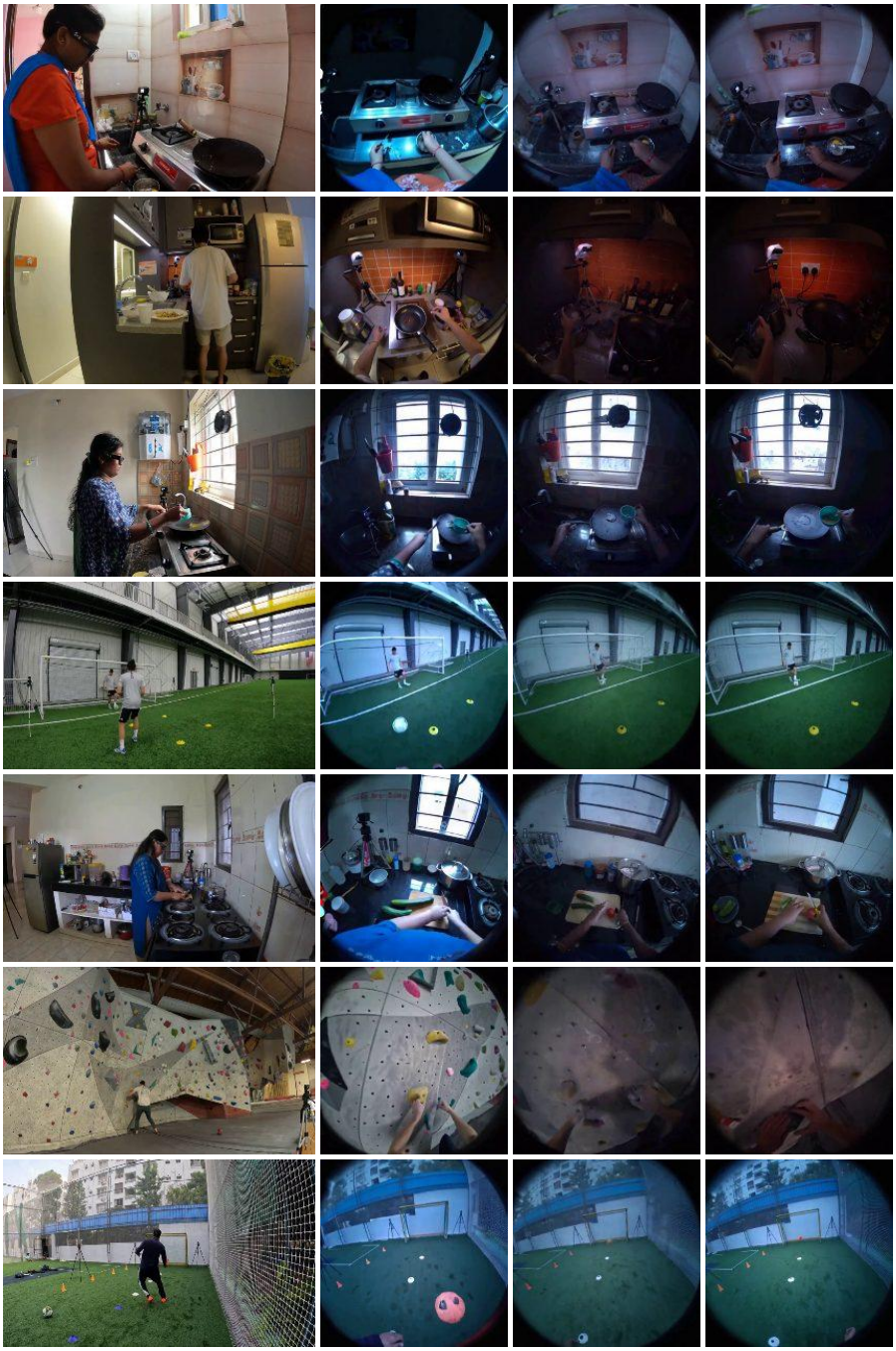

Figure 9: Additional qualitative comparisons on the Ego-Exo4D seen split.

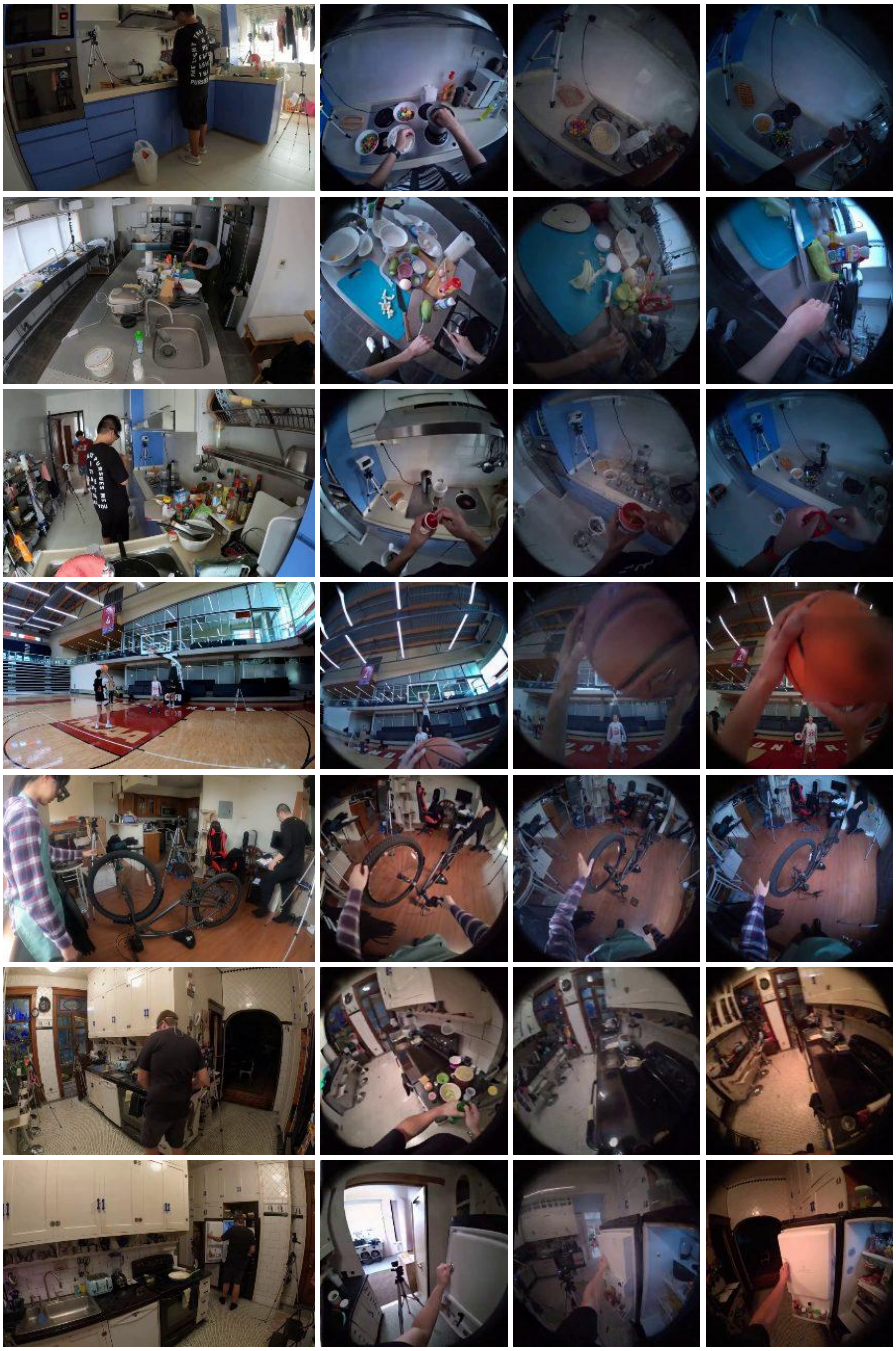

Figure 10: Additional qualitative comparisons on the Ego-Exo4D unseen split.

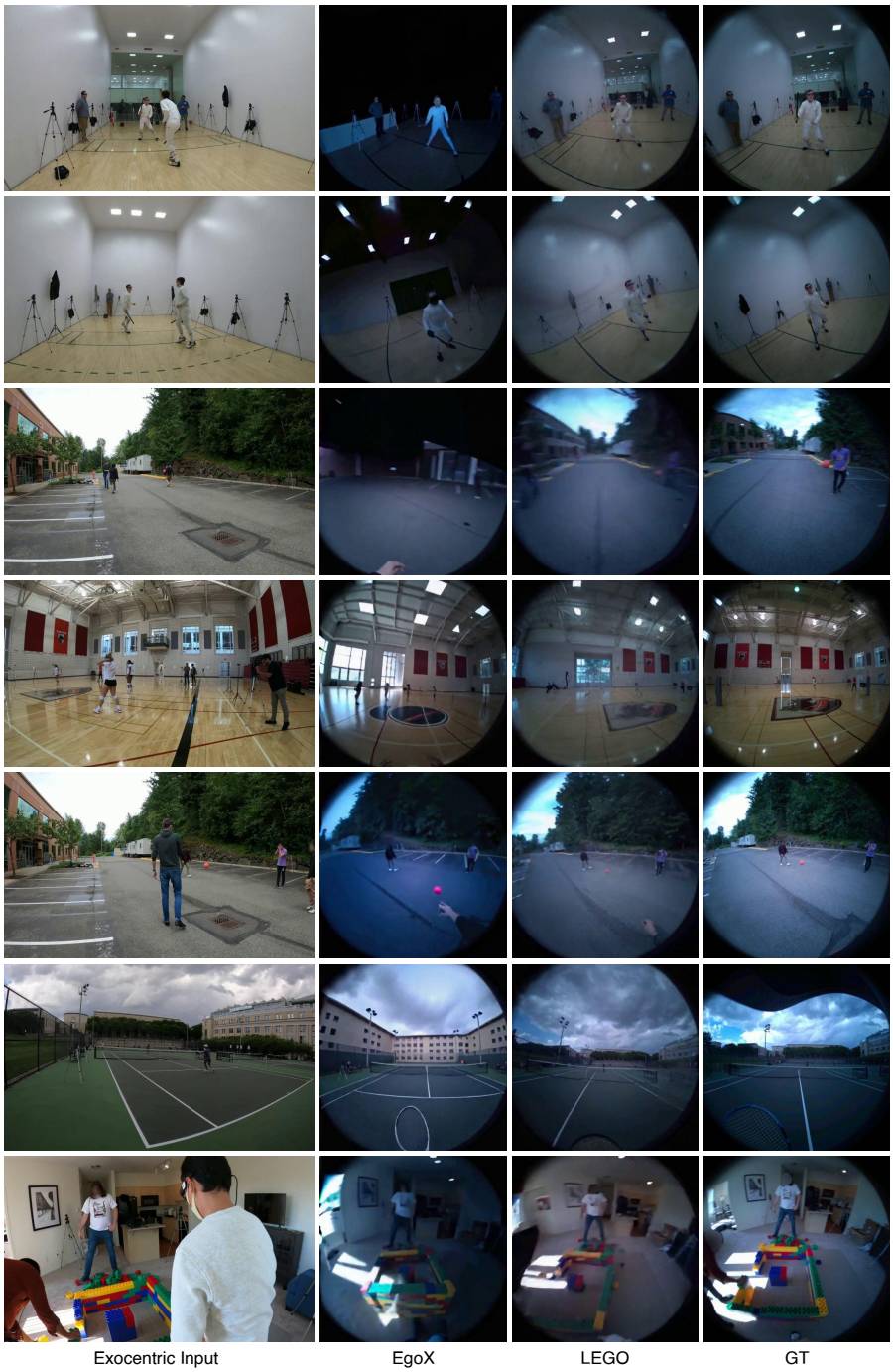

Figure 11: Additional qualitative comparisons on EgoHumans.

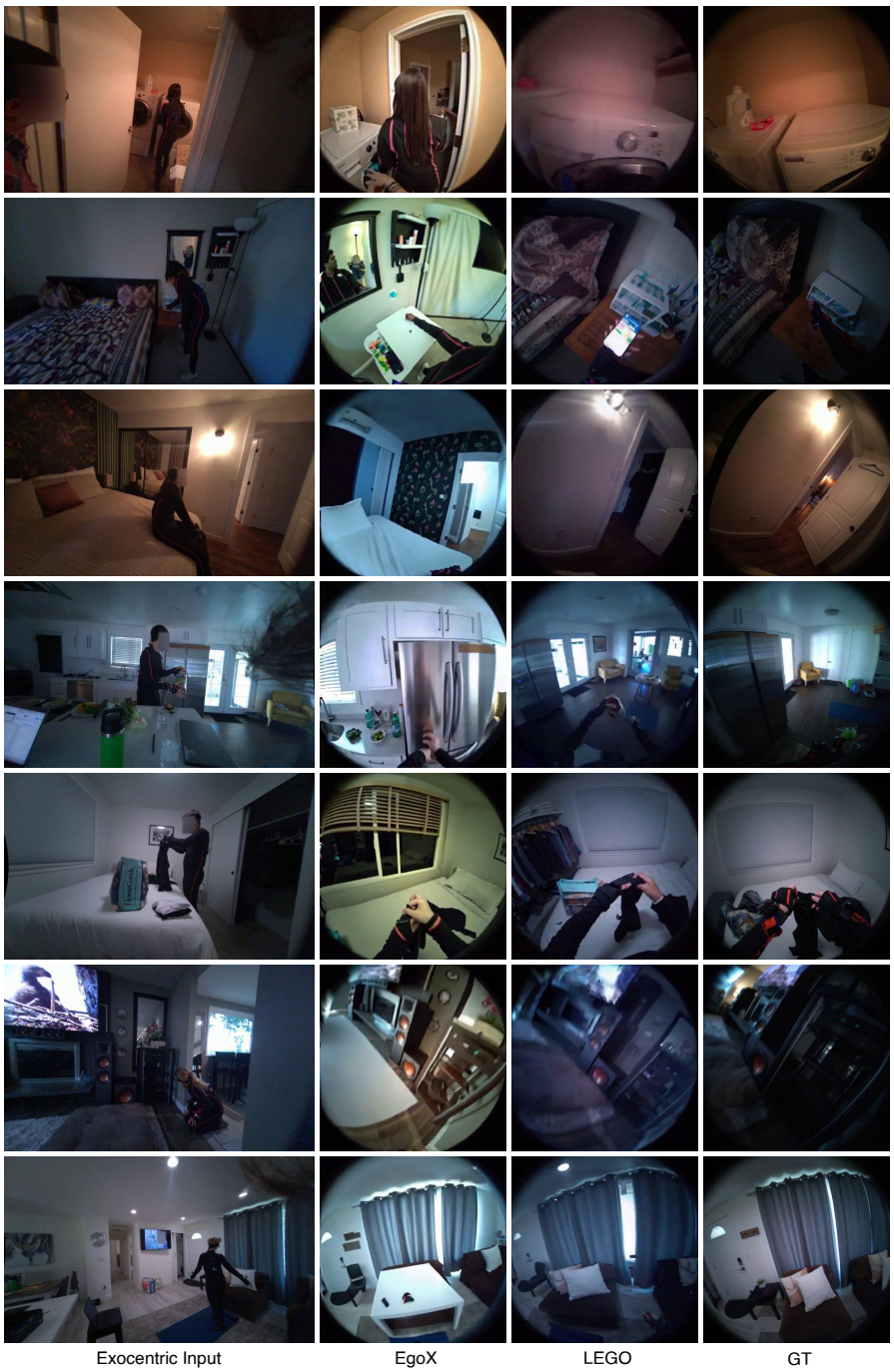

Figure 12: Additional qualitative comparisons on Nymeria.

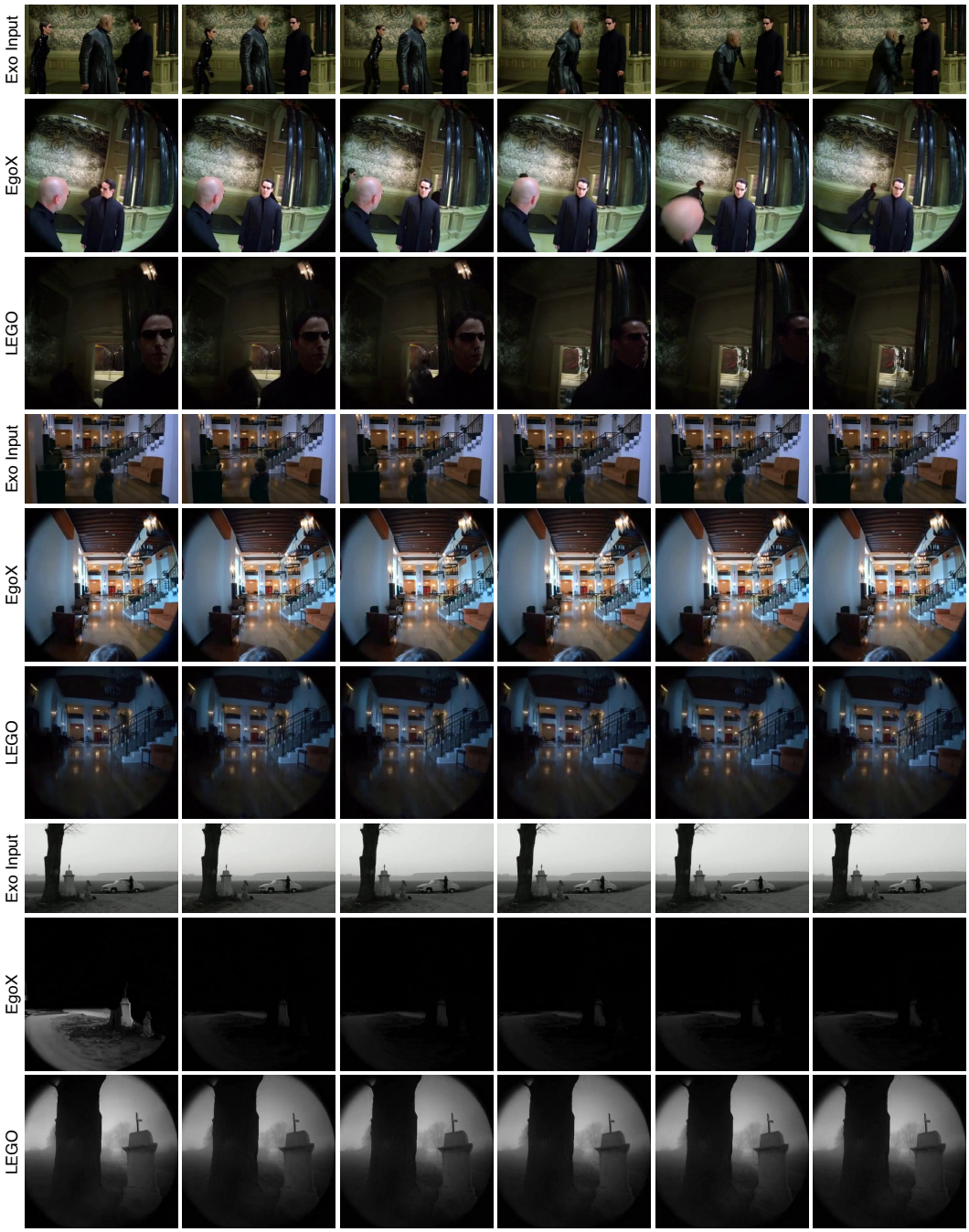

Figure 13: Additional qualitative comparisons on in-the-wild videos.