GibbsLens: Training-Free Camera Control via Block-Conditional Gibbs Refinement

Yuzhu Wang[*1,2] **Xi Ye**[*1] **Duo Su**[1] **Yangyang Xu**[1] **Jun Zhu**[1†]
[1]Department of Computer Science and Technology, Tsinghua University
[2]South China University of Technology

Abstract

Training-free camera control for pretrained flow-matching video generators is a partial-observation inverse problem: a depth-warped guidance video supplies noisy evidence on a subset of latent sites, which the sampler must reconcile with the pretrained prior. Existing methods struggle to balance the trade-off between trajectory adherence and visual quality and the heuristic guidance-strength tuning lacks robustness. We propose ***GibbsLens***, which resolves this dilemma through a structural change to the sampler: each outer hard-replacement guidance step is augmented with an inner-loop *block-conditional pseudo-Gibbs refinement* on the unobserved complement at the same noise level, with provable convergence to the partial-observation conditional data law. To accelerate convergence on high-dimensional video latents, we exploit their conditional locality, partitioning the unobserved complement into 3D patches, each tracked by a custom mixing indicator that adaptively freezes converged patches. On RealEstate10K and DAVIS, ***GibbsLens*** attains the best FVD against all seven training-free and training-based competitors, outperforming every training-free baseline on every reported metric. Project page: https://xiye20.github.io/projects/GibbsLens/.

1 Introduction

Reliable camera control is essential in filmmaking, simulation, virtual reality, and embodied AI, where the desired camera trajectory is given at inference time. The conventional approach involves retraining a video generation model to follow trajectory instructions, which is costly [1]. Training-free camera control sidesteps this cost but remains difficult even for strong modern video diffusion and flow-matching models. During sampling, camera motion is entangled with object motion, scene layout, and appearance synthesis, making it much harder to prescribe a target viewpoint trajectory than to specify text semantics alone.

Specifically, existing camera-control methods can be organized into two paradigms. *Training-based* methods encode camera trajectories and fine-tune the generator to learn the mapping between these control signals and the output video [2, 3, 4, 5, 1]. While effective, it requires model-specific training pipelines and relies on paired trajectory–video datasets that are expensive to acquire. *Training-free* methods instead synthesize a warped guidance video from the desired trajectory geometry and inject it as inference-time evidence [6, 7, 8], sidestepping the training cost entirely.

Essentially, training-free conditional generation is an inverse-problem question. Two competing forces arise during the sampling process. The *posterior* (trajectory codes) pulls the sample toward the conditioning evidence, while the *prior* (pretrained models) shapes it according to the learned distribution [9, 10, 11]. Tilting toward the posterior tightens adherence to the evidence but exposes the sample to noise and artifacts in the observation; tilting toward the prior preserves visual quality

[*]Equal contribution.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

but loosens adherence. Existing training-free samplers balance these forces through guidance strength (the inverse temperature in energy guidance, or the gradient step size in DPS-style samplers [11, 12]). Tuning this scalar is heuristic and lacks robustness across settings.

Training-free camera control is a representative instance of this dilemma. Hard latent replacement [13, 14, 15, 16] operates at maximum guidance strength, treating the warped video as noise-free evidence and overwriting the masked latent regions at every denoising step. This delivers tight trajectory control; however, the warp is inherently imperfect due to depth-estimation errors, projection artifacts, and downsampled visibility, so the strong observation introduces unstable hallucinated content and artifacts along mask boundaries. Soft posterior guidance [11] and windowed guidance [17] relax the guidance, recovering visual quality at the cost of trajectory adherence.

Rather than retuning the guidance strength, we propose *GibbsLens* to resolve the dilemma through a structural change to the sampler. Started with a hard masked latent replacement at the outer denoising step, which preserves tight trajectory control, we introduce a *block-conditional pseudo-Gibbs refinement*: an inner loop that refines the prior on the unobserved region at the same noise level. The refinement extends Bengio's generalized denoising-auto-encoder Markov chain [18] to a partial-observation conditional setting, and a sub-state ergodicity argument shows its iterates converge to the partial-observation conditional data law. To make this primitive practical at video scale, we leverage the property of latent flow-matching models that video latents exhibit short-range conditional dependence along all spatial-temporal axes. Building on this locality, we partition the unobserved complement into 3D patches. Each patch is then tracked by a mixing indicator that freezes it once its inner iterations have converged, accelerating convergence and improving overall sample quality.

Our contributions are three-fold:

1. We construct a generalized denoising-auto-encoder chain on the unobserved complement, provably converging to the partial-observation conditional data law, which yields a principled inner-loop alternative to heuristic guidance-strength tuning.
2. We empirically validate the conditional locality of video flow-matching latents, justifying a block-conditional Gibbs sampler that adaptively freezes locally-converged patches to accelerate convergence.
3. On RealEstate10K and DAVIS, *GibbsLens* attains the best FVD against all seven competitors and dominates training-free baselines across reported metrics.

2 Preliminaries

Latent flow matching. We work with pretrained latent flow-matching video generators sampling in the VAE latent space $z = \mathcal{E}(x)$. Under the variance-exploding reparameterization of the probability-flow ODE [19, 20, 21], the marginals are $z_t = (1 - \sigma_t)\, z_0 + \sigma_t\, \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ along the schedule $\sigma_0 = 0, \sigma_1 = 1$. The learned velocity $u_\theta(z_t, t, c)$, regressed on $\epsilon - z_0$, implements the deterministic ODE $dz_t/dt = u_\theta(z_t, t, c)$ with c the external condition (e.g., a text prompt for T2V). Under the velocity–score identity $u_\theta = (\dot\sigma_t/\sigma_t)\, z_t - \dot\sigma_t \sigma_t\, \nabla_{z_t} \log p_t(z_t \,|\, c)$, score-based reasoning transfers to flow matching. The plug-in clean-prediction estimator

$$\hat z_0(z_t, t, c) \;=\; z_t - \sigma_t\, u_\theta(z_t, t, c) \tag{1}$$

is the primary interface to the guidance objectives in Section 3.

Doob's h-transform for conditional sampling. Many tasks require conditioning the reverse process on an endpoint event, e.g., a noisy inverse problem $y = \mathcal{A}(z_0) + n$. Doob's h-transform [22] is the general framework, tilting the path measure by a non-negative terminal weight $h(z_0)$ (the likelihood $p(y \,|\, z_0)$ for soft conditioning). With time-dependent $h_t(z_t) = \mathbb{E}[h(z_0) \,|\, z_t]$ [23, 16], the conditioned reverse SDE adds an h-induced drift,

$$dz_t = \big[\, f(z_t, t) - g(t)^2 \big(\nabla_{z_t} \log p_t(z_t) + \nabla_{z_t} \log h_t(z_t)\big) \,\big]\, dt + g(t)\, d\bar w_t, \tag{2}$$

with drift f, diffusion g, and reverse-time Wiener process $\bar w_t$; via the velocity–score identity, this gives the controlled flow-matching velocity

$$u_\theta^{\mathrm{ctrl}}(z_t, t, c) \;=\; u_\theta(z_t, t, c) - \sigma_t\, \dot\sigma_t\, \nabla_{z_t} \log h_t(z_t). \tag{3}$$

The h-induced drift is intractable in general; practical samplers replace it with a tractable surrogate [11, 16, 24], e.g., the first-order Tweedie surrogate $\nabla_{z_t} \log h_t(z_t) \approx \nabla_{z_t} \log h(\hat{z}_0(z_t, t, c))$ recovering DPS [11]. Appendix B.4 reviews surrogate families and locates *GibbsLens*.

3 Method

We treat training-free camera-controlled video generation as a partial-observation inverse problem [11, 13]. Given a target camera trajectory and a source video or first frame, we follow standard camera-control practice [14]: a monocular depth estimator lifts the source to a point cloud, which is reprojected under the target poses to produce a warped guidance video $\tilde{x}_0$ together with its pixel-space visibility map. Encoding $\tilde{x}_0$ through the pretrained VAE and downsampling the visibility map to the latent resolution yields the warped latent $\tilde{z}_0 = \mathcal{E}(\tilde{x}_0)$ and a binary mask $M \in \{0,1\}^{L\times H\times W}$ marking the geometrically observed sites on the latent grid of temporal length L and spatial size $H \times W$. Both signals are noisy because they inherit depth-estimation errors, boundary artifacts, and M is only approximate due to the downsampling.

Vanilla h-transform guidance. We frame camera control as posterior sampling under a partial Gaussian observation model. With the pretrained video generator as the prior $p(z_0)$ and the warped latent $\tilde{z}_0$ as noisy evidence on the masked sites,

$$p(z_0 \mid \tilde{z}_0, M) \;\propto\; p(z_0)\,\ell(\tilde{z}_0 \mid z_0, M), \qquad \ell(\tilde{z}_0 \mid z_0, M) \;\propto\; \exp\!\Big(-\tfrac{1}{2\tau^2}\|M \odot (z_0 - \tilde{z}_0)\|_2^2\Big), \quad (4)$$

with τ^{-2} the per-site observation confidence. This is the Doob-transformed reverse process of Eq. (2) with terminal weight $h(z_0) = \ell(\tilde{z}_0 \mid z_0, M)$. Following DPS [11], we approximate the intractable h-induced drift by the first-order Tweedie surrogate $\nabla_{z_t} \log h_t(z_t) \approx \nabla_{z_t} \log h(\hat{z}_0(z_t, t, c))$ and drop the denoiser Jacobian (Appendix B.4). Substituting into Eq. (3) yields

$$u_\theta^{\mathrm{ctrl}}(z_t, t, c) \;=\; u_\theta(z_t, t, c) \;+\; \frac{\sigma_t \dot{\sigma}_t}{\tau^2}\, M \odot \big(\hat{z}_0(z_t, t, c) - \tilde{z}_0\big), \quad (5)$$

a soft pull toward $\tilde{z}_0$ on observed sites with strength τ^{-2}. As $\tau \to 0$, the likelihood concentrates on $\{z_0 : M \odot z_0 = M \odot \tilde{z}_0\}$ and $\hat{z}_0^{\mathrm{ctrl}} = z_t - \sigma_t u_\theta^{\mathrm{ctrl}}$ collapses to the hard masked replacement

$$\hat{z}_0^{\mathrm{obs}} \;=\; M \odot \tilde{z}_0 \;+\; (1 - M) \odot \hat{z}_0(z_t, t, c). \quad (6)$$

We adopt Eq. (6) as the outer guidance step of *GibbsLens*: the pretrained flow is preserved on $1-M$, and finite-confidence behavior is restored by the inner refinement of Section 3.1.

3.1 Conditional Pseudo-Gibbs Refinement

The hard-replacement update Eq. (6) is the canonical training-free inpainting rule [13, 14, 15], but two assumptions break here. The warped latent $\tilde{z}_0$ inherits depth-estimation errors and projection artifacts. And the unobserved support $1-M$, left to the pretrained prior alone, must here follow the target camera trajectory's spatio-temporal geometry rather than merely look plausible. We address both by coupling each outer step with an inner refinement loop at the same noise level.

Inner-loop refinement. At fixed σ_t, the flow-matching backbone acts as a denoiser: $\hat{z}_0(z_t, t, c)$ is its reconstruction from a noisy input. Bengio et al. [18] show that iterating perturb-and-redenoise defines a generalized denoising auto-encoder (DAE) Markov chain whose stationary distribution converges to $p(z_0)$. We extend this primitive to a partial-observation conditional setting by (i) holding the observed sites at a freshly drawn noised pin from the OT-FM forward kernel at level σ_t,

$$\bar{z}_t \;=\; (1 - \sigma_t)\,\tilde{z}_0 \;+\; \sigma_t\,\xi_{\mathrm{obs}}, \qquad \xi_{\mathrm{obs}} \sim \mathcal{N}(0, I), \quad (7)$$

and (ii) restricting the perturb step to the unobserved support. Initializing $\hat{z}_0^{(0)} = \hat{z}_0^{\mathrm{obs}}$ from Eq. (6), the inner chain alternates

$$z_t^{(j)} \;=\; (1 - M) \odot \big((1 - \sigma_t)\,\hat{z}_0^{(j-1)} + \sigma_t\,\xi^{(j)}\big) \;+\; M \odot \bar{z}_t, \qquad \xi^{(j)} \sim \mathcal{N}(0, I), \quad (8)$$

$$\hat{z}_0^{(j)} \;=\; z_t^{(j)} \;-\; \sigma_t\, u_\theta\big(z_t^{(j)}, t, c\big), \qquad j = 1, \ldots, J_{\max}, \quad (9)$$

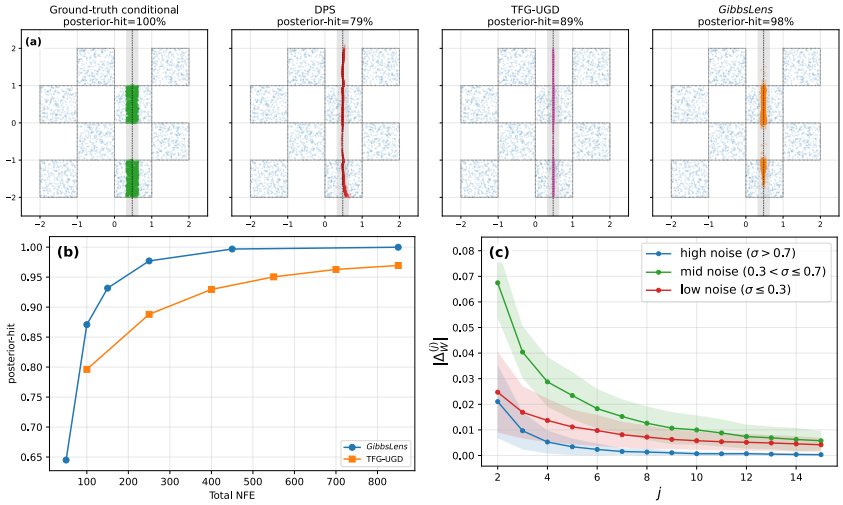


Figure 1: 2D checkerboard toy example. **(a)** Sample clouds at $y_{\text{obs}} \approx 0.5$ for ground truth, DPS [11], TFG-UGD [12], and *GibbsLens* (left to right). **(b)** Posterior-hit rate vs. total NFE for *GibbsLens* (varying J_{max}) and TFG-UGD (varying N_{recur}). **(c)** $|\Delta_W^{(j)}|$ vs. inner iteration j binned by noise band.

followed by a hard-mask write-back that composes the two branches before one FlowMatch Euler step advances the latent. The write-back consumes the *Polyak-averaged Gibbs readout* $\bar{z}_0 = \frac{1}{J_{\max}} \sum_{j=1}^{J_{\max}} \hat{z}_0^{(j)}$ rather than the last iterate $\hat{z}_0^{(J_{\max})}$:

$$\hat{z}_0^{\text{final}} \;=\; M \odot \hat{z}_0^{\text{obs}} \;+\; (1 - M) \odot \bar{z}_0. \tag{10}$$

By Polyak and Juditsky [25] on iterate averaging, this trades the per-sample posterior variance for an $\mathcal{O}(\tau_{\text{int}}/J_{\max})$ Monte Carlo variance — exactly what the FlowMatch Euler step, which is linear in $\hat{z}_0^{\text{final}}$, expects. Pinning observed sites at $\bar{z}_t$ rather than at the outer-state $z_t|_M$ keeps the inner-loop conditioning at the correct noised-observation distribution and decouples the chain from outer-state drift. With σ_t and $\bar{z}_t|_M$ held fixed, Eqs. (8)–(9) form a generalized DAE chain on the sub-state $z|_{1-M}$ whose stationary distribution admits a closed-form characterization.

Proposition 1 (Stationary distribution of conditional pseudo-Gibbs)**.** *Fix denoising step t with noise level $\sigma_t > 0$, mask M, warped latent $\tilde{z}_0$, and pin $\bar{z}_t$ from Eq. (7). Assume* (A1) *the denoiser $u_\theta(\cdot, t, c)$ is consistent with the level-σ_t joint conditional on c (denoising-score-matching optimality), and* (A2) *the perturbation kernel restricted to $1-M$ has full support on a bounded volume. Then Eqs. (8)–(9) define an ergodic Markov chain whose clean-prediction iterates $\hat{z}_0^{(j)}|_{1-M}$ converge in distribution, as $j \to \infty$, to the partial-observation conditional data law $p\big(z_0|_{1-M} \,\big|\, \bar{z}_t|_M\big)$.*

The proof proceeds by reduction to the sub-state version of Bengio et al. [18] Theorem 1; it is given in Appendix C, alongside the comparison with RePaint [13] and SRVS [26].

Mixing detection. We monitor convergence through the trajectory of $\hat{z}_0^{(j)}$ itself. Let $\sigma_W^{(j)}$ denote the running standard deviation of the inner iterates on the unobserved support, maintained by Welford's online algorithm [27]. The Δ-Welford indicator is its first difference,

$$\Delta_W^{(j)} \;=\; \sigma_W^{(j)} \;-\; \sigma_W^{(j-1)}, \tag{11}$$

with $|\Delta_W^{(j)}| \to 0$ signalling that the chain has stopped exploring.

2D toy validation. We validate the construction on a 2D 8-square checkerboard with a noisy partial observation on the first coordinate, anchored at $y_{\text{obs}} \approx 0.5$ where the vertical constraint line yields a 2-mode conditional posterior. The toy experiment establishes three claims: *(i) Mode coverage.* As in Fig. 1a, DPS [11] and TFG-UGD [12] spread mass along the constraint line without locking onto either mode, while *GibbsLens* concentrates on both conditional modes. Disabling the inner loop reduces *GibbsLens* to DPS-like behaviour, isolating the refinement as the source of the gain. *(ii) Compute scaling.* The posterior-hit rate of *GibbsLens* rises monotonically with the per-step probe

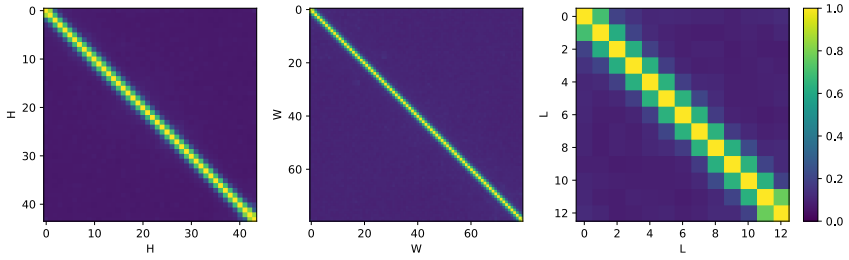


Figure 2: Top canonical partial correlation $\rho_1(\widehat{R}_{\beta\gamma})$ along the H, W, L axes of the Wan 2.2 latent ($N = 200$ encoded videos). Off-diagonal mass concentrates within $|\beta - \gamma| \leq 2$.

count and strictly dominates TFG-UGD at every matched NFE budget (Fig. 1b): compute spent inside the unobserved sub-state pays off more than TFG-style global recurrence. *(iii) Indicator fidelity.* $|\Delta_W^{(j)}|$ decays over the same j-range in which the posterior-hit curve of (b) plateaus, across every noise band (Fig. 1c); the indicator is therefore a reliable ground-truth-free trigger for the early-freeze gate of Section 3.2. Detailed toy example implementation is in Appendix F.

3.2 From Toy to Video: Locality and Block-Conditional Gibbs

At video scale — the Wan 2.2 latent has shape (C, L, H, W) with $C = 48$ and $\sim 10^5$ sites — a generalized DAE chain on this full sub-state mixes too slowly (per-probe sampling variance scales with state dimension). We therefore propose a novel *block-conditional Gibbs* sampler over 3D patches, paired with a per-patch Δ-Welford freeze gate (Section 3.1); block updates dominate single-site Gibbs in autocorrelation reduction [28, 29]. Such a partition is legitimate only if the latent's conditional dependence decays along each axis on a short range — which we verify empirically next.

Latents are local. Let $z^{(n)}_{l,h,w} \in \mathbb{R}^C$ denote the Wan 2.2 latent token of video n at grid position (l, h, w). The diagnostic asks a single question, one axis at a time: pick $\alpha \in \{L, H, W\}$ and two positions β, γ on α; are the tokens at β and γ conditionally independent given the tokens at all other positions on that line? The natural unit-free statistic for multichannel tokens is the top *block partial correlation* $\rho_1(\widehat{R}_{\beta\gamma}) \in [0, 1]$, the strongest linear dependence between $z^{(n)}_\beta$ and $z^{(n)}_\gamma$ *after regressing out the rest of the line*; under joint Gaussianity, $\rho_1 = 0$ is the exact conditional-independence test. See Appendix D for the detailed implementation of the block partial correlation.

Figure 2 shows the off-diagonal $\rho_1(\widehat{R}_{\beta\gamma})$ decays to a finite-sample noise floor of ≤ 0.1 within $|\beta - \gamma| \leq 2$ on each of L, H, W: the latent is approximately order-2 Markov. This is conditional independence, not separability — the joint $p(z_0)$ stays non-factorisable, with distant coordinates marginally correlated via chains of neighbours — and is consistent with the bounded receptive fields of convolutional VAE encoders [30, 31, 32, 33]. The same locality is preserved on the model's clean prediction $\hat{z}_0(z_t, t, c)$ across $\sigma_t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ (Appendix D).

Patch-freeze block-conditional Gibbs. The locality structure justifies partitioning the unobserved support $1-M$ into 3D patches $\{\mathcal{P}_g\}_{g=1}^G$ of size $p_l \times p_h \times p_w$ chosen larger than the order-2 dependence radius. Locality then constrains $p(z_0|_{\mathcal{P}_g} \mid z_0|_{\bar{\mathcal{P}}_g})$ to depend only on a halo around $\mathcal{P}_g$, bounding the block-Gibbs autocorrelation and allowing each patch to be frozen independently once its inner chain stabilizes. Within each patch the inner-loop perturb-and-redenoise of Eqs. (8)–(9) is a block Gibbs update on the sub-state $z|_{\mathcal{P}_g \cap (1-M)}$ conditioned on a thin halo of neighboring patches plus the pinned observed sites. We index the Δ-Welford indicator of Section 3.1 by patch as $\Delta_W^{(j)}(g)$ and track its running peak $\Delta_W^{\max}(g) = \max_{j' \leq j} |\Delta_W^{(j')}(g)|$. We declare a patch *stable* when its $|\Delta_W(g)|$ stays below a fraction $\kappa \in (0, 1]$ of $\Delta_W^{\max}(g)$ in two consecutive iterations:

$$S_g^{(j)} \;=\; \mathbf{1}\big[\,|\Delta_W^{(j)}(g)| < \kappa \Delta_W^{\max}(g)\big] \;\wedge\; \mathbf{1}\big[\,|\Delta_W^{(j-1)}(g)| < \kappa \Delta_W^{\max}(g)\big]. \tag{12}$$

Stable patches hold their previous clean-prediction iterate while the rest update as usual:

$$\hat{z}_0^{(j)}\big|_{\mathcal{P}_g} \;\leftarrow\; S_g^{(j)}\, \hat{z}_0^{(j-1)}\big|_{\mathcal{P}_g} \;+\; (1 - S_g^{(j)})\, \hat{z}_0^{(j)}\big|_{\mathcal{P}_g}, \quad g = 1, \ldots, G. \tag{13}$$

The loop exits early when the fraction of stable patches $\sum_g S_g^{(j)}/G$ exceeds ν, and is capped at $J_{\max}$.

Algorithm 1 *GibbsLens*: training-free camera-controlled flow-matching sampler.

Require: pretrained u_θ, conditioning c; warped $\tilde{z}_0$, mask M; window $[s, e)$; $J_{\max}$; $\kappa, \nu \in (0, 1]$.

for $k = 0, \dots, K-1$ **do**
  $\hat{z}_0 \leftarrow z_k - \sigma_k u_\theta(z_k, t_k, c)$
  if $s \le k < e$ **then**
    $\hat{z}_0^{\text{obs}} \leftarrow M \odot \tilde{z}_0 + (1 - M) \odot \hat{z}_0$ Eq. (6)
    $\bar{z}_k \leftarrow (1 - \sigma_k)\tilde{z}_0 + \sigma_k \xi_{\text{obs}}$; $\hat{z}_0^{(0)} \leftarrow \hat{z}_0^{\text{obs}}$; $\bar{z}_0$, $\Delta_W^{(0)}(g)$, $\Delta_W^{\max}(g)$, $S_g^{(0)} \leftarrow 0$ Eq. (7)
    for $j = 1, \dots, J_{\max}$ **do**
      $z_k^{(j)} \leftarrow (1 - M) \odot \big((1 - \sigma_k)\hat{z}_0^{(j-1)} + \sigma_k \xi^{(j)}\big) + M \odot \bar{z}_k$ Eq. (8)
      $\hat{z}_0^{(j)} \leftarrow z_k^{(j)} - \sigma_k u_\theta(z_k^{(j)}, t_k, c)$ Eq. (9)
      Refresh running $\Delta_W^{(j)}(g), \Delta_W^{\max}(g)$ (Welford bookkeeping)
      $S_g^{(j)} \leftarrow \mathbf{1}\big[|\Delta_W^{(j)}(g)| < \kappa \Delta_W^{\max}(g)\big] \wedge \mathbf{1}\big[|\Delta_W^{(j-1)}(g)| < \kappa \Delta_W^{\max}(g)\big]$ Eq. (12)
      $\hat{z}_0^{(j)}|_{\mathcal{P}_g} \leftarrow S_g^{(j)} \hat{z}_0^{(j-1)}|_{\mathcal{P}_g} + (1 - S_g^{(j)}) \hat{z}_0^{(j)}|_{\mathcal{P}_g}$ Eq. (13)
      $\bar{z}_0 \leftarrow \bar{z}_0 + (\hat{z}_0^{(j)} - \bar{z}_0)/j$ Polyak-averaged readout
      break if $\sum_g S_g^{(j)}/G > \nu$ early exit on stable fraction
    end for
    $\hat{z}_0^{\text{final}} \leftarrow M \odot \hat{z}_0^{\text{obs}} + (1 - M) \odot \bar{z}_0$ Eq. (10)
  else
    $\hat{z}_0^{\text{final}} \leftarrow \hat{z}_0$
  end if
  $z_{k+1} \leftarrow z_k + (\sigma_{k+1} - \sigma_k)(z_k - \hat{z}_0^{\text{final}})/\sigma_k$
end for

Algorithm 1 gives the full *GibbsLens* sampler over the discretized schedule $\{t_k\}_{k=0}^{K}$ ($\sigma_k = \sigma_{t_k}$, $z_k = z_{t_k}$) with camera-control window $[s, e) \subseteq [0, K)$ and at most $J_{\max}$ inner iterations per active outer step, where K denotes the number of discrete sampling steps. Proposition 1 characterizes the asymptotic limit of the underlying inner chain under its stated assumptions; it does not provide a finite-step convergence guarantee for this implementation with adaptive patch freezing, early stopping, and a Polyak-averaged readout.

4 Experiments

Setup. We evaluate on *RealEstate10K* (200 randomly sampled test scenes) and *DAVIS* (84 videos obtained from three controlled trajectories of varying angular extent on each of 28 dynamic scenes), comparing *GibbsLens* against four training-free (WorldForge [15], TTM [14], RePaint [13], Coarse-Guided [16]) and three training-based (†; TrajectoryAttention [34], TrajectoryCrafter [35], ReCamMaster [1]) camera controllers. *GibbsLens* and every training-free baseline share the public Wan 2.2 [33] (Text-Image-to-Video-5B) flow-matching backbone, ensuring a controlled comparison in which the sampler is the only varied factor; the warped guidance video and its visibility mask are produced by the TTM/WorldForge depth-based lift-and-reproject pipeline, while ReCamMaster takes its native camera-trajectory input. All experiments run on NVIDIA A40 GPUs in mixed precision.

Evaluation metrics. For visual quality, we retain Fréchet Video Distance (FVD) [36] for comparability, while acknowledging its limited sensitivity to temporal distortions [37]. On both benchmarks, we complement it with Fréchet Video Motion Distance (FVMD) [38] for motion consistency and VisionReward's overall score (VR-Overall) [39] as a learned proxy for human visual preferences. We also report frame-wise LPIPS and SSIM against paired held-out reference views on RealEstate10K, and CLIP-v [40] and CLIP-f [41] on both benchmarks. For trajectory adherence, camera poses re-extracted with Mega-SAM [42] yield the Absolute Trajectory Error (ATE), Relative Rotation Error (RRE), and Relative Translation Error (RTE).

4.1 Comparison with State-of-the-Art Methods

Static scenes (RealEstate10K). Table 1 reports quantitative results. *GibbsLens* matches or beats every training-free baseline across all ten metrics, reducing FVD from 157.08 to **129.25** (17.7%) and FVMD from 1606.55 to 1314.96 (18.2%) relative to Coarse-Guided. It also achieves the highest VR-Overall (**0.1277** vs. 0.1153 for TTM), alongside gains in trajectory error, LPIPS, and CLIP-based scores, with SSIM tied at 0.62. Across all methods, *GibbsLens* ranks first on seven of ten metrics and second on FVMD. TrajectoryCrafter achieves lower FVMD (1151.24) and RRE (0.4235), and TrajectoryAttention higher SSIM (0.64), but their FVDs are 2–5× ours and their VR-Overall scores are lower (0.0525 and 0.0200, respectively). These results support improvements over training-free baselines in motion consistency and preference-aligned quality, beyond FVD and paired-frame fidelity. Figure 3 shows qualitative comparisons: *GibbsLens* synthesises photorealistic novel views aligned with the target poses, while baselines exhibit visible warp seams and disocclusion artifacts.

Table 1: Visual quality and trajectory-error results on **RealEstate10K**. Best across all methods in **bold**, second-best underlined. † marks training-based methods.

	FVD↓	FVMD↓	VR-Overall↑	ATE (cm)↓	RRE (deg)↓	RTE (cm)↓	LPIPS↓	SSIM↑	CLIP-f↑	CLIP-v↑
WorldForge [15]	158.47	1915.55	0.0968	5.8838	0.4373	1.4176	0.42	0.56	97.87	92.54
TTM [14]	192.88	1611.57	0.1153	3.8700	0.4751	1.1745	0.48	0.52	98.16	93.39
Coarse-Guided [16]	157.08	1606.55	0.0737	3.5719	0.4403	0.9535	0.42	0.62	97.84	91.60
RePaint [13]	265.00	6515.38	0.0812	10.3861	0.7438	2.8434	0.52	0.54	97.66	90.13
TrajectoryAttention† [34]	616.81	9858.78	0.0200	18.0606	0.8492	3.2817	0.45	**0.64**	97.10	87.23
TrajectoryCrafter† [35]	253.27	**1151.24**	0.0525	4.1568	**0.4235**	0.9794	0.43	0.63	97.71	91.50
ReCamMaster† [1]	299.33	11554.00	0.0111	5.1924	0.4627	1.6436	0.59	0.53	98.21	89.34
***GibbsLens* (Ours)**	**129.25**	1314.96	**0.1277**	**3.4920**	0.4321	**0.9483**	**0.41**	0.62	**98.52**	**93.80**

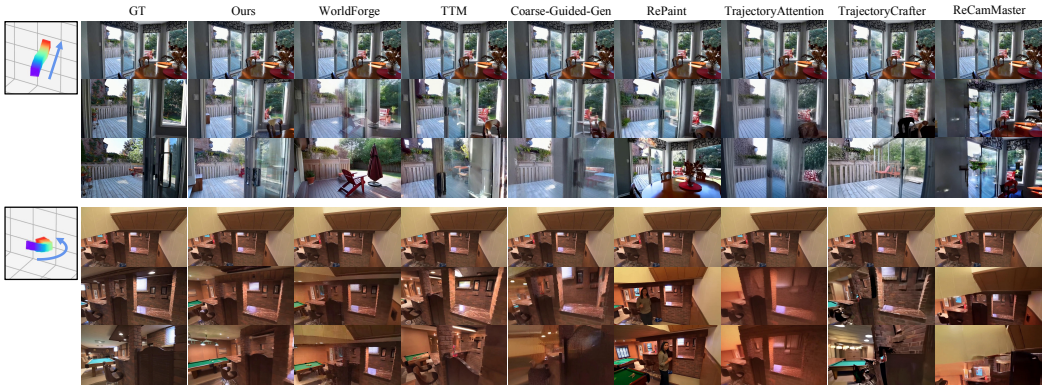


Figure 3: Qualitative results on RealEstate10K. Compared with the baselines, our method generates photorealistic novel views that align with the target camera poses.

Dynamic scenes (DAVIS). Quantitative results are reported in Table 2. Against the four training-free baselines, *GibbsLens* is again best on every metric. Relative to the strongest baseline for each quality metric, it reduces FVD from 383.53 to **356.73** and FVMD from 9288.76 to **8782.67** (5.4%), both against WorldForge, and increases VR-Overall from 0.0918 to **0.0974** against TTM. These gains provide complementary evidence of improved motion consistency and preference-aligned quality in dynamic scenes. The comparison with training-based controllers exposes a clear quality-versus-control trade-off: ReCamMaster, fine-tuned end-to-end as a pose-conditioned generator, achieves the lowest trajectory errors in terms of ATE, RRE, and RTE. This trajectory-metric lead is structural: only ReCamMaster conditions directly on the camera trajectory, whereas all others operate on a depth-warped guidance video whose projection errors inherently cap trajectory accuracy. However, its FVD (670.42), FVMD (25387.40), and VR-Overall (−0.0406) indicate weaker visual quality than *GibbsLens*. *GibbsLens* instead achieves the best FVD, FVMD, VR-Overall, CLIP-v, and CLIP-f overall, together with the second-best trajectory errors after ReCamMaster, attaining the most favourable balance between trajectory adherence and visual quality among the seven controllers. Figure 4 visualises the same trade-off: hard-replacement baselines show seams and

unstable disoccluded content, ReCamMaster's appearance degrades under dynamic content, while *GibbsLens* remains seam-free and geometrically faithful.

Table 2: Trajectory-error and visual quality results on **DAVIS**. Best across all methods in **bold**, second-best underlined. † marks training-based methods.

	FVD↓	FVMD↓	VR-Overall↑	ATE (cm)↓	RRE (deg)↓	RTE (cm)↓	CLIP-f↑	CLIP-v↑
WorldForge [15]	383.53	9288.76	0.0658	8.2694	1.9323	5.1275	97.05	89.25
TTM [14]	398.12	11948.70	0.0918	7.8849	1.9111	4.3735	97.35	89.85
Coarse-Guided [16]	402.14	11414.45	0.0116	9.0672	2.1216	4.3978	96.40	87.35
RePaint [13]	690.83	10820.76	−0.1147	9.9780	2.2468	5.1842	95.33	74.76
TrajectoryAttention† [34]	837.37	16028.85	−0.0816	18.3749	3.6871	7.2034	93.88	74.23
TrajectoryCrafter† [35]	576.30	10246.31	−0.0626	7.3827	2.1045	4.7158	95.86	83.21
ReCamMaster† [1]	670.42	25387.40	−0.0406	**4.9770**	**0.6352**	**1.7271**	95.39	85.06
***GibbsLens* (Ours)**	**356.73**	**8782.67**	**0.0974**	6.8832	1.8907	4.2285	**97.45**	**90.34**

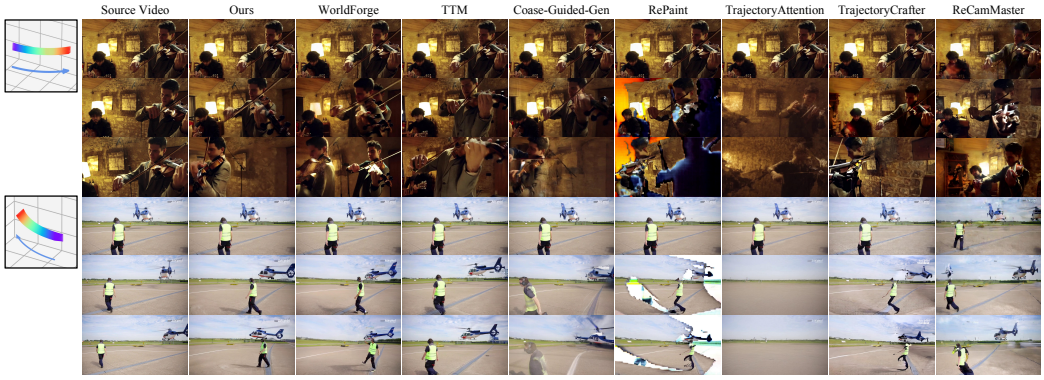


Figure 4: Qualitative results on DAVIS. On dynamic scenes, *GibbsLens* achieves accurate camera control and good visual quality simultaneously, while baselines trade one for the other.

Across both regimes, *GibbsLens* matches or exceeds every training-free baseline on every reported metric and achieves the best FVD and VR-Overall among all compared methods, with the best FVMD on DAVIS and second-best on RealEstate10K. These complementary metrics support a favourable balance of visual quality, motion consistency, and trajectory adherence without fine-tuning.

Computational cost. Table 3 compares training-free methods using the same Wan 2.2 backbone to generate a 49-frame video at 720×1280 on one NVIDIA A40 GPU. Relative to TTM, *GibbsLens* increases runtime from 227 to 413 seconds (81.9%), with nearly unchanged peak GPU memory (26.1641 vs. 26.1719 GiB). It remains faster than RePaint and WorldForge in this setting. Its full-budget denoiser count is $48 + 3\times10 = 78$ evaluations; early stopping in Algorithm 1 can reduce this count. NFE alone does not determine wall-clock cost, since auxiliary operations and implementation differences also contribute to runtime.

Table 3: Computational cost of training-free camera control under a shared backbone and evaluation setting. NFE counts denoiser evaluations; for *GibbsLens*, the full-budget count is reported.

Method	Time (s)	NFE	Peak GPU memory (GiB)
WorldForge[15]	751	65	37.0781
TTM[14]	227	48	26.1641
Coarse-Guided[16]	235	50	26.1289
RePaint[13]	472	100	25.6328
***GibbsLens* (Ours)**	413	78	26.1719

4.2 Analysis and Ablation Studies

Table 4: Component ablation on **DAVIS**. ✓ marks active components; top row is the outer h-transform guidance alone, bottom row the full sampler. Best in **bold**, second-best underlined.

Pseudo-Gibbs refinement	Polyak readout	Patch-freeze gate	FVD↓	ATE (cm)↓	RRE (deg)↓	RTE (cm)↓	CLIP-f↑	CLIP-v↑
			392.13	7.9558	2.0128	4.4130	96.63	88.93
✓			382.26	7.6343	1.9554	4.3351	96.98	89.71
✓	✓		375.67	7.6051	2.0758	4.5340	96.91	89.49
✓		✓	382.26	7.8734	**1.8775**	4.5266	96.96	89.62
✓	✓	✓	**356.73**	**6.8832**	1.8907	**4.2285**	**97.45**	**90.34**

Component ablation. Table 4 isolates the three sampler components introduced in Section 3 — *pseudo-Gibbs refinement* (Section 3.1), the *Polyak-averaged readout* (Eq. (10)), and the Δ-*Welford patch-freeze gate* (Section 3.2) — by toggling each on top of the outer h-transform step on DAVIS. Activating *pseudo-Gibbs refinement* alone (row $1 \to 2$) reduces FVD from 392.13 to 382.26 and improves every other metric uniformly, confirming that the perturb-and-redenoise primitive on the unobserved support already provides a moderate uniform gain over the outer h-transform step alone. Adding the *Polyak-averaged readout* (row $2 \to 3$) further drops FVD to 375.67 with CLIP scores essentially unchanged, consistent with the variance reduction afforded by iterate averaging on the linear FlowMatch step; trajectory errors drift slightly upward (RRE 2.08, RTE 4.53) because averaging smooths the per-iterate signal that drives sharp pose alignment. The Δ-*Welford patch-freeze gate* (row $2 \to 4$) instead attains the best RRE in the table ($\mathbf{1.8775}$) while keeping CLIP scores stable; we hypothesise that the gain stems from fixing converged patches at their stabilised iterate before drift sets in on rotation-sensitive regions. FVD, however, remains unchanged (382.26): the gate protects already-mixed structure rather than improving distributional fidelity on its own. The two enhancements are therefore *complementary*: combining all three (row 5) attains the best score on five of six metrics and second-best on RRE, with the freeze gate and the Polyak readout compensating for each other's costs and yielding gains substantially larger than their individual contributions.

Block-conditional Gibbs mixing. Figure 5 visualises the per-patch stability mask S_g across inner iterations on three frames. The top two rows show S_g at $j = 4$ and $j = 12$, and the bottom row the corresponding warped frames. As j grows, the stable region (frozen patches) expands from a sparse seed and eventually fills the unobserved support. This patch-by-patch convergence directly validates the latent locality assumption: if patches were globally coupled, no patch could stabilise before the rest, and the stability mask would never exhibit a spatially structured growth pattern.

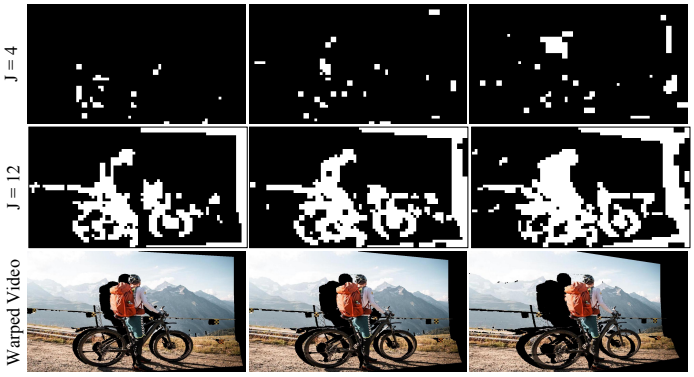


Figure 5: **Stability mask S_g evolution** at the initial outer guidance step (noise level σ_{t_s}). Stable region grows with j to cover the unobserved support.

5 Related Works

Camera-Controllable Video Generation. Existing camera-control methods can be broadly divided into *training-based* and *training-free* paradigms. Training-based methods inject camera trajectories, camera-aware representations, or multi-view synchronization modules into the generator and learn the

mapping from control signals to motion through additional supervision [2, 3, 4, 5, 43, 44, 1, 45]; they achieve strong controllability but rely on model-specific retraining and suitable camera-annotated or multi-view data. By contrast, training-free methods construct geometric guidance at inference time, e.g., via point-cloud or depth-based warping, latent reframing, or explicit view-conditioned rendering cues [7, 8, 6, 46], with related trajectory-editing and novel-view synthesis systems combining explicit geometry with generative models for monocular video re-rendering or sparse-view synthesis [47, 35, 48]. Our method belongs to the training-free family but differs in how the warped signal is used: instead of treating warped content as a replacement target, we interpret it as *partial and spatially non-uniform evidence* and refine the unobserved complement through a block-conditional pseudo-Gibbs sampler equipped with an adaptive Δ-Welford patch-freeze gate.

Training-Free Conditional Sampling. Diffusion Posterior Sampling (DPS) [11] instantiates Doob's h-transform under a Gaussian observation likelihood via a Tweedie surrogate; the noise-aware weighted-h-transform of Wang et al. [16] extends DPS with a global-scalar confidence weight, and Zhu et al. [49] pursues the same formal object via fine-tuning rather than inference-time guidance. Inference-time conditional samplers such as the Twisted Diffusion Sampler and Feynman–Kac steering [24, 50] steer pretrained models without retraining, and MPGD [10] shows that aggressive guidance can push trajectories off the data manifold. Our inner refinement extends Bengio's generalised denoising auto-encoder [18] to the partial-observation conditional setting; SRVS [26] uses the same primitive unconditionally on the full latent, and plug-and-play / restoration-prior methods [51, 52, 53, 54] share the spirit of using a pretrained denoiser as an implicit prior. Detailed positioning and SRVS distinctions are in Appendix C.3.

6 Conclusion and Discussion

GibbsLens is a training-free posterior sampler for pretrained flow-matching video generators that recasts camera control as conditional posterior sampling under a partial Gaussian observation likelihood. Its technical contribution is a *block-conditional pseudo-Gibbs* refinement on the unobserved sub-state. We accelerate convergence on high-dimensional video latents through a locality-justified patch partition, made adaptive by a Δ-Welford patch-freeze gate that holds each patch fixed once its inner iterates have converged. Across RealEstate10K and DAVIS, *GibbsLens* attains the best FVD against all seven training-free and training-based competitors and improves over every training-free baseline on every metric, while remaining competitive with training-based controllers without fine-tuning.

Limitations. Two caveats are worth noting. First, the order-2 Markov property used to size the block-conditional patches is calibrated to the Wan 2.2 latent (Appendix D). Latent locality itself is a generic property of VAE-encoded video models, but the precise dependence order must be re-measured per backbone before *GibbsLens* is applied. Second, the outer step inherits the geometric accuracy of the depth-and-reproject warping pipeline — severe depth errors degrade the conditioning likelihood, a constraint shared by every training-free method that consumes warped guidance.

Future directions. Applying *GibbsLens* to other partial-observation video inverse problems such as video inpainting, super-resolution and masked novel-view synthesis is a promising direction, since these tasks share the same partial-evidence structure and could be tackled by the same training-free sampler. Another open challenge is to accelerate the refinement process: the global attention still incurs a full forward pass over every token regardless of which patches are frozen, so sparse-attention or KV-caching variants of DiT that skip frozen tokens, or distilled re-denoising surrogates, are needed to translate the freeze gate's adaptivity into wall-clock speed-ups.

Acknowledgments

This work was supported in part by the NSFC Special Fund and Major Program under Grants 62550004 and 62595773; in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM101); in part by the CPSF (No. 2025M771546).

References

[1] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14834–14844, 2025.

[2] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

[3] Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13416–13426, 2025.

[4] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. *arXiv preprint arXiv:2407.12781*, 2024.

[5] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22875–22889, 2025.

[6] David Junhao Zhang, Roni Paiss, Shiran Zada, Nikhil Karnad, David E Jacobs, Yael Pritch, Inbar Mosseri, Mike Zheng Shou, Neal Wadhwa, and Nataniel Ruiz. Recapture: Generative video camera controls for user-provided videos using masked video fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2050–2062, 2025.

[7] Chen Hou and Zhibo Chen. Training-free camera control for video generation. *arXiv preprint arXiv:2406.10126*, 2024.

[8] Zhenghong Zhou, Jie An, and Jiebo Luo. Latent-reframe: Enabling camera control for video diffusion models without training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12779–12789, 2025.

[9] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, pages 22825–22855. PMLR, 2023.

[10] Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, et al. Manifold preserving guided diffusion. *arXiv preprint arXiv:2311.16424*, 2023.

[11] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[12] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. *Advances in Neural Information Processing Systems*, 37:22370–22417, 2024.

[13] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

[14] Assaf Singer, Noam Rotstein, Amir Mann, Ron Kimmel, and Or Litany. Time-to-move: Training-free motion controlled video generation via dual-clock denoising. *arXiv preprint arXiv:2511.08633*, 2025.

[15] Chenxi Song, Yanming Yang, Tong Zhao, Ruibo Li, and Chi Zhang. Taming video models for 3d and 4d generation via zero-shot camera control. *arXiv preprint arXiv:2509.15130*, 2025.

[16] Yanghao Wang, Ziqi Jiang, Zhen Wang, and Long Chen. Coarse-guided visual generation via weighted h-transform sampling. *arXiv preprint arXiv:2603.12057*, 2026.

[17] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

[18] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. *Advances in neural information processing systems*, 26, 2013.

[19] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *The Ninth International Conference on Learning Representations*, 2021.

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

[21] 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*, 2022.

[22] L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*, volume 2. Cambridge university press, 2000.

[23] Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula J Komorowska, and Pietro Lio. Deft: Efficient fine-tuning of diffusion models by learning the generalised h-transform. *Advances in Neural Information Processing Systems*, 37:19636–19682, 2024.

[24] Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

[25] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

[26] Sangwon Jang, Taekyung Ki, Jaehyeong Jo, Saining Xie, Jaehong Yoon, and Sung Ju Hwang. Self-refining video sampling. *arXiv preprint arXiv:2601.18577*, 2026.

[27] Barry Payne Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

[28] Jun S Liu, Wing Hung Wong, and Augustine Kong. Covariance structure of the gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, pages 27–40, 1994.

[29] Gareth O Roberts and Sujit K Sahu. Updating schemes, correlation structure, blocking and parameterization for the gibbs sampler. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 59(2):291–317, 1997.

[30] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[32] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[33] 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.

[34] Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. *arXiv preprint arXiv:2411.19324*, 2024.

[35] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 100–111, 2025.

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

[37] Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in Frechet Video Distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7277–7288, 2024.

[38] Jiahe Liu, Youran Qu, Qi Yan, Xiaohui Zeng, Lele Wang, and Renjie Liao. Fréchet Video Motion Distance: A metric for evaluating motion consistency in videos. *arXiv preprint arXiv:2407.16124*, 2024.

[39] Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, Jiayan Teng, Zhuoyi Yang, Wendi Zheng, Xiao Liu, Dan Zhang, Ming Ding, Xiaohan Zhang, Shiyu Huang, Xiaotao Gu, Minlie Huang, Jie Tang, and Yuxiao Dong. VisionReward: Fine-grained multi-dimensional human preference learning for image and video generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (13):11269–11277, 2026. doi: 10.1609/aaai.v40i13.38107.

[40] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas J Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. *Advances in Neural Information Processing Systems*, 37:16240–16271, 2024.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[42] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10486–10496, 2025.

[43] 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 *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[44] 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.

[45] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision*, pages 313–331. Springer, 2024.

[46] Jangho Park, Taesung Kwon, and Jong Chul Ye. Zero4d: Training-free 4d video generation from single video using off-the-shelf video diffusion. *arXiv preprint arXiv:2503.22622*, 2025.

[47] 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. *arXiv preprint arXiv:2409.02048*, 2024.

[48] Junyoung Seo, Jisang Han, Jaewoo Jung, Siyoon Jin, Joungbin Lee, Takuya Narihira, Kazumi Fukuda, Takashi Shibuya, Donghoon Ahn, Shoukang Hu, et al. Vid-camedit: Video camera trajectory editing with generative rendering from estimated geometry. *arXiv preprint arXiv:2506.13697*, 2025.

[49] Qijie Zhu, Zeqi Ye, Han Liu, Zhaoran Wang, and Minshuo Chen. Training-free adaptation of diffusion models via doob's h-transform. *arXiv preprint arXiv:2602.16198*, 2026.

[50] Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

[51] Suhas Sreehari, S Venkat Venkatakrishnan, Brendt Wohlberg, Gregery T Buzzard, Lawrence F Drummy, Jeffrey P Simmons, and Charles A Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2 (4):408–423, 2016.

[52] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM journal on imaging sciences*, 10(4):1804–1844, 2017.

[53] Yuyang Hu, Mauricio Delbracio, Peyman Milanfar, and Ulugbek Kamilov. A restoration network as an implicit prior. In *The Twelfth International Conference on Learning Representations*, 2023.

[54] Matthieu Terris, Ulugbek S Kamilov, and Thomas Moreau. Fire: Fixed-points of restoration priors for solving inverse problems. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23185–23194, 2025.

[55] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[56] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[57] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[58] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[59] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

A Extended Related Work and Positioning

This appendix expands Section 5 with a technical positioning of *GibbsLens* against the four research lines it sits between: camera-controllable video generation, training-free conditional sampling on pretrained diffusion/flow models, plug-and-play denoiser-as-prior methods, and generalized denoising auto-encoders.

A.1 Camera-controllable video generation

Training-based controllers. CameraCtrl [2] and CameraCtrl II [3] inject Plücker-coordinate camera embeddings into a CogVideoX-style backbone. VD3D [4] and AC3D [5] train video DiTs with explicit 3D-aware tokenization. MotionCtrl [43] trains a separate motion module conditioned on camera and object motion fields. Multi-view extensions: SyncCammaster [44] synchronizes generation across views, and ReCamMaster [1] learns to re-render an input video under a new trajectory by joint training on synchronized multi-camera data. Several systems build a 4D scaffold from a source video and guide the generator with it: TrajectoryAttention [34] introduces trajectory-aware attention; TrajectoryCrafter [35] renders explicit point clouds from monocular input and trains a refinement diffusion model on top. ViewCrafter [47] and Vid-camedit [48] solve sparse-view novel-view synthesis by combining explicit 3D priors with generative refinement; Van Hoorick et al. [45] provides a generative-modeling perspective on the same problem. The common requirement across this family is that at least one component — backbone, adapter, or refinement head — is fine-tuned to internalize the trajectory-to-video correspondence.

Training-free controllers. TTM [14] and WorldForge [15] construct a warped guidance video by lifting the source to a point cloud through monocular depth and reprojecting under the target camera, then run a pretrained generator with hard latent replacement on the visibility mask at every denoising step. Recapture [6] and Latent Reframing [8] use a similar warp-and-replace primitive with mask construction tuned for static scenes. Zero4D [46] extends this to free-camera 4D generation in a zero-shot fashion, again via masked replacement. Coarse-Guided [16] drops the partial mask and instead applies a global noise-aware soft pull toward the warped reference. Both ends of this spectrum are degenerate boundary limits of *GibbsLens*: at $\sigma_y \to 0$ the warp likelihood concentrates on $\{z_0 : M \odot z_0 = M \odot \tilde{z}_0\}$ and the write-back collapses to hard replacement; at $M \equiv 1$ the partial mask becomes the whole grid and the soft pull reduces to Coarse-Guided. These are sanity-check consistency properties, not the regime where *GibbsLens* operates. The novel object is the interior: at finite σ_y on a non-trivial partial mask, the inner block-conditional Gibbs refinement targets the partial-observation conditional posterior $p(z_0|_{1-M} \mid \bar{z}_t|_M)$ (Proposition 1), which has no counterpart in either baseline and is what delivers the gains in Section 4.1.

A.2 Training-free conditional sampling

First-order surrogates: DPS family. Diffusion Posterior Sampling [11] replaces the intractable h-induced drift with the gradient of the observation likelihood evaluated at the Tweedie mean $\hat{z}_0$. A line of follow-ups substitutes alternative point estimates for $\hat{z}_0$ or alternative likelihoods (FreeDoM, ΠGDM, MPGD [10], TFG-UGD), all sharing the property that the surrogate is a single point evaluation of $\nabla \log h$. *Position of GibbsLens:* the outer step is exactly a DPS-style surrogate *restricted to the partial mask* M; the structural addition is the inner block-conditional Gibbs refinement on $1-M$.

Noise-aware weighted surrogates. The DPS surrogate is most reliable at high σ_t and degrades as $\sigma_t \to 0$, where $\hat{z}_0$ becomes deterministic. Weighted h-transform sampling [16] introduces a noise-level-aware scalar λ_{σ_t} that down-weights the surrogate at low noise. DEFT [23] amortizes the surrogate by training a small network that predicts the h-induced drift directly, and Zhu et al. [49] fine-tunes the base model to internalize the same drift. *Position of GibbsLens:* we extend the global scalar λ_{σ_t} to a *spatially non-uniform* mask M and pair it with a novel inner refinement on the unobserved support, while keeping the conditioning entirely at inference time — no extra network and no fine-tuning.

Sequential Monte Carlo and Feynman–Kac. Twisted Diffusion Samplers [24] and Feynman–Kac steering [50] maintain a population of trajectories and reweight them by the running terminal

compatibility, recovering an unbiased estimator of $\nabla \log h_t$ in the large-particle limit. The price is $N\times$ inference compute per step. *Position of GibbsLens:* the inner Gibbs chain is a single-trajectory variance-reduction primitive operating on the same physical sample, achieving similar conditional adherence at a small per-step multiplier rather than $N\times$.

A.3 Plug-and-play and denoiser-as-prior methods

PnP-ADMM [51], RED [52], and score-based extensions [53, 54] share the "denoiser as implicit prior" perspective: a pretrained denoiser is invoked at a single fixed noise level inside a half-quadratic or proximal solver, acting as a projection-onto-prior step against a hard data-fidelity constraint. *GibbsLens* reuses the same intuition but in a structurally different setting — a multi-step flow-matching sampler with a time-varying schedule, where the implicit prior is queried at every σ_t and conditioned on a partial observation, so the inner chain targets $p(z_0|_{1-M} \mid \bar{z}_t|_M)$ rather than the unconditional data law that PnP targets.

B Flow matching and Doob's h-transform background

This appendix expands the brief summary in Section 2 into a self-contained derivation, starting from score-based diffusion as a forward SDE, deriving the velocity–score identity that underwrites flow matching, and arriving at the controlled flow-matching velocity used in Eq. (5) of the method section. Readers fluent in score-based diffusion and Doob's h-transform can skip directly to Section B.4.

B.1 Score-based diffusion via SDEs

Score-based generative models [19] cast sampling as the time reversal of a noising SDE that transports data $z_0 \sim p_0$ to a tractable prior $p_T = \mathcal{N}(0, I)$:

$$dz_t \;=\; f(z_t, t)\, dt \;+\; g(t)\, dw_t, \qquad t \in [0, T], \tag{14}$$

with drift f, diffusion coefficient g, and Wiener process w_t. Its time-reversed dynamics [55], integrated from $z_T \sim p_T$ back to $t = 0$, recover samples from p_0:

$$dz_t \;=\; \big[f(z_t, t) - g(t)^2\, \nabla_{z_t} \log p_t(z_t)\big]\, dt \;+\; g(t)\, d\bar{w}_t, \tag{15}$$

where p_t is the marginal of z_t and $\bar{w}_t$ a reverse-time Wiener process. The score $\nabla_{z_t} \log p_t$ is learned as $s_\theta(z_t, t)$ via denoising score matching [56]. The same marginals $\{p_t\}$ can be induced by the deterministic probability-flow ODE [19],

$$dz_t \;=\; \big[f(z_t, t) - \tfrac{1}{2}\, g(t)^2\, \nabla_{z_t} \log p_t(z_t)\big]\, dt. \tag{16}$$

B.2 Flow matching as reparameterization, and the velocity–score identity

Flow matching [20, 21] is equivalent to a reparameterization of the probability-flow ODE Eq. (16). Choosing the variance-exploding coefficients [19] $f(z_t, t) = (\dot{\sigma}_t / \sigma_t)\, z_t$ and $g(t) = \sqrt{2\, \dot{\sigma}_t\, \sigma_t}$ in Eq. (14) produces marginals $z_t = (1 - \sigma_t) z_0 + \sigma_t \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ along a monotone schedule $\sigma_0 = 0, \sigma_1 = 1$. Substituting these into Eq. (16) reduces it to the deterministic ODE $dz_t/dt = u_\theta(z_t, t, c)$ through the *velocity–score identity*

$$u_\theta(z_t, t, c) \;=\; \frac{\dot{\sigma}_t}{\sigma_t}\, z_t \;-\; \dot{\sigma}_t\, \sigma_t\, \nabla_{z_t} \log p_t(z_t \mid c), \tag{17}$$

where c is the external condition (e.g., text for T2V). Flow matching trains u_θ to regress $\epsilon - z_0$; sampling integrates the ODE backward from $z_1 \sim \mathcal{N}(0, I)$ to z_0. Eq. (17) is the load-bearing bridge in *GibbsLens*'s derivation: every step that adds an h-induced drift to the score in Eq. (2) can be rewritten as an additive correction to u_θ, giving the controlled velocity Eq. (3). The plug-in clean-prediction estimator Eq. (1) follows from Eq. (17) by Tweedie's identity $\hat{z}_0(z_t) = z_t + \sigma_t^2 \nabla_{z_t} \log p_t(z_t)$ and the forward parameterization above.

B.3 Doob's h-transform for conditional sampling

Many practical tasks impose an *endpoint constraint* on the reverse SDE rather than just a terminal distribution — for instance, the noisy inverse problem $y = \mathcal{A}(z_0) + n$ with known operator $\mathcal{A}$,

observation y, and noise n. Conditioning the marginals p_t in Eq. (15) on such an event is intractable, since the conditional dynamics couples through all future steps. Doob's h-transform [22] resolves this by tilting the unconditioned path measure with a non-negative terminal weight $h(z_0)$ — e.g. the indicator $\mathbf{1}_B(z_0)$ for hard conditioning, or the likelihood $p(y \mid z_0)$ for soft conditioning on y — yielding a Markov diffusion whose endpoint follows the desired conditional and whose drift carries an additive correction.

Following Denker et al. [23], Wang et al. [16], define the time-dependent h-function as the conditional expectation of the terminal weight:

$$h_t(z_t) \;=\; \mathbb{E}\big[\, h(z_0) \;\big|\; z_t \big] \;=\; \int p_{0|t}(z_0 \,|\, z_t)\, h(z_0)\, dz_0, \tag{18}$$

where $p_{0|t}$ is the unconditional posterior of the clean endpoint given the current state. Doob's theorem yields the conditioned reverse SDE Eq. (2) with an additive *h-induced drift* $\nabla_{z_t} \log h_t(z_t)$, and via Eq. (17) the controlled flow-matching velocity Eq. (3). For the noisy inverse problem with $h(z_0) = p(y \mid z_0)$, Eq. (18) specializes to $h_t(z_t) = p_t(y \mid z_t)$, recovering the standard posterior-sampling identity $\nabla_{z_t} \log p_t(z_t \,|\, y) = \nabla_{z_t} \log p_t(z_t) + \nabla_{z_t} \log p_t(y \,|\, z_t)$ [23].

B.4 Tractable surrogates for $\nabla \log h_t$

The h-induced drift $\nabla_{z_t} \log h_t(z_t)$ in Eq. (2) is intractable in general, since evaluating the h-function Eq. (18) requires the unconditional posterior $p_{0|t}$ of a large pretrained model and a marginalization over future trajectories. Practical conditional sampling therefore replaces $\nabla \log h_t$ with a tractable surrogate. The useful taxonomic axis turns out to be *whether the surrogate backpropagates through the denoiser to compute its gradient*; we group the established families along this axis and locate *GibbsLens* among them.

Common first-order point surrogate. All single-trajectory families below share the same starting point: replace the intractable expectation in Eq. (18) by h evaluated at the Tweedie posterior mean $\hat{z}_0(z_t)$ (Eq. (1)),

$$h_t(z_t) \;\approx\; h\big(\hat{z}_0(z_t)\big) \;\Longrightarrow\; \nabla_{z_t} \log h_t(z_t) \;\approx\; \nabla_{z_t} \log h\big(\hat{z}_0(z_t)\big). \tag{19}$$

The chain rule expands the right-hand side as

$$\nabla_{z_t} \log h\big(\hat{z}_0(z_t)\big) \;=\; \underbrace{\big(\nabla_{\hat{z}_0} \log h\big)\big(\hat{z}_0(z_t)\big)}_{\text{cheap, no autograd}} \cdot \underbrace{\nabla_{z_t} \hat{z}_0(z_t)}_{= I - \sigma_t \nabla_{z_t} u_\theta(z_t, t, c)}, \tag{20}$$

so a faithful evaluation requires the denoiser Jacobian $\nabla_{z_t} u_\theta$. Whether this Jacobian is computed in full or dropped is what distinguishes families (i) and (ii) below.

(i) Jacobian-aware first-order surrogates (gradient through the denoiser). This family computes Eq. (20) in full, backpropagating through one or more denoiser forwards. Diffusion Posterior Sampling (DPS) [11] is the canonical instance: for a Gaussian observation $h(z_0) = \mathcal{N}(y; \mathcal{A}(z_0), \tau^2 I)$ the surrogate becomes the gradient of a measurement-consistency loss with respect to z_t, computed by autograd through $\hat{z}_0(z_t)$ and hence through u_θ. Manifold Preserving Guided Diffusion (MPGD) [10] additionally projects the Jacobian-computed gradient onto the data-manifold tangent to prevent off-manifold drift. Training-Free Guidance with universal-guidance recurrence (TFG-UGD) [12] runs several inner mean-guidance iterations per outer step, each backpropagating through the denoiser. The defining feature of this family is the autograd-through-u_θ requirement: each surrogate evaluation costs at least one denoiser-sized backward pass and the activations must be retained for backprop, so the memory footprint scales with the model and resolution. This is the price paid for using the full first-order Taylor expansion of $\log h$ around z_t.

(ii) Jacobian-free first-order surrogates (stop-grad through the denoiser). This family uses the same point surrogate Eq. (19) but drops the Jacobian in Eq. (20): $\nabla_{z_t} \hat{z}_0 \approx I$, equivalent to detaching $\hat{z}_0$ from z_t before gradient computation. The surrogate then collapses to $\nabla_{z_t} \log h(\hat{z}_0) \approx \nabla_{\hat{z}_0} \log h(\hat{z}_0)$, which is cheap and requires no backprop through u_θ. This "stop-grad" approximation is documented in DPS itself (see Chung et al. 11, App. A) and is the standard choice for inference-time guidance on large pretrained backbones where the Jacobian-aware variant is computationally infeasible. Two

strategies within this family compensate for the loss of Jacobian information. *(a) Noise-aware weighting.* Weighted h-transform sampling [16] introduces $\lambda_{\sigma_t} = \sigma_t^{\alpha}$ on the surrogate,

$$\nabla_{z_t} \log h_t(z_t) \;\approx\; \lambda_{\sigma_t}\, \nabla_{\hat{z}_0} \log h\big(\hat{z}_0(z_t)\big), \tag{21}$$

down-weighting at low σ_t where the Jacobian carries the most information and the stop-grad approximation degrades fastest. For coarse-guided flow-matching with $h(z_0) \propto \delta(z_0 - \tilde{y})$ replaced by a tractable proxy $\tilde{y}$, this yields the closed-form controlled velocity

$$u_t^{\text{ctrl}} \;=\; u_\theta \;+\; \lambda_{\sigma_t}\Big(\frac{z_t - \tilde{y}}{\sigma_t} - u_\theta\Big), \tag{22}$$

interpolating between the pretrained velocity ($\lambda_{\sigma_t} \to 0$) and a hard pull toward the warped reference ($\lambda_{\sigma_t} \to 1$). *(b) Amortized surrogate.* DEFT [23] trains a small network to predict $\nabla_{z_t} \log h_t$ directly, sidestepping the inference-time Jacobian by amortizing it offline; Zhu et al. [49] fine-tunes the base model itself to internalize the same correction. Both incur an extra training stage but, like the weighted variant, avoid autograd through the pretrained denoiser at sampling time.

(iii) Particle-based surrogates (Sequential Monte Carlo / Feynman–Kac). Twisted Diffusion Samplers [24] and Feynman–Kac steering [50] maintain a population of trajectories and reweight them by the running terminal compatibility, recovering an unbiased estimate of $\nabla \log h_t$ in the large-particle limit at the cost of $N\times$ inference compute per step. They are orthogonal to (i)/(ii) along the Jacobian axis: the bias of the surrogate is reduced by particle resampling rather than by a better single-point estimator, so neither the Jacobian nor a stop-grad proxy enters the picture.

B.5 Position of our work

GibbsLens belongs to family (ii) (Jacobian-free first-order surrogate): the surrogate is the Tweedie point estimate Eq. (19) with the denoiser Jacobian $\nabla_{z_t} u_\theta$ dropped. The Jacobian-free choice is mandatory rather than optional in our setting — the Wan 2.2 backbone has ~ 5B parameters at $720 \times 1280 \times 49$ resolution, so each backward through u_θ would dominate the per-step compute of any Jacobian-aware surrogate from family (i). What *GibbsLens* adds on top of the existing family (ii) is a *spatially non-uniform* confidence map: we replace the global scalar weight λ_{σ_t} of Eq. (21) by the warp-derived mask M over latent sites, so the surrogate is applied at full strength on observed sites and zero on $1 - M$, recovering most of the Jacobian's spatial-information content at zero autograd cost. The complementary inner block-conditional Gibbs refinement on $1 - M$ (Section 3) is what closes the gap on the unobserved support that family (ii) leaves open by construction.

C Block-conditional pseudo-Gibbs sampling

This appendix reads *GibbsLens*'s inner refinement loop as a *block-conditional pseudo-Gibbs Markov chain*: we recall the generalized DAE / pseudo-Gibbs framework of Bengio et al. [18], prove Proposition 1 as its sub-state extension targeting the partial-observation conditional law, and contrast the resulting chain with SRVS [26]. All notation follows Section 2: the pretrained latent flow-matching model induces a velocity field $u_\theta(z_k, t_k, c)$ and a clean-prediction estimator $\hat{z}_0(z_k, t_k, c) = z_k - \sigma_k\, u_\theta(z_k, t_k, c)$ via Eq. (1).

C.1 Generalized denoising auto-encoders as generative models

Let X denote the clean random variable and $\tilde{X}$ its corruption under a fixed conditional $\mathcal{C}(\tilde{X} \mid X)$. Bengio et al. [18] define the *pseudo-Gibbs* Markov chain associated with a learned reconstruction conditional $P_\theta(X \mid \tilde{X})$ as

$$X^{(j)} \;\sim\; P_\theta(X \mid \tilde{X}^{(j-1)}), \qquad \tilde{X}^{(j)} \;\sim\; \mathcal{C}(\tilde{X} \mid X^{(j)}), \tag{23}$$

with transition operator $T(X^{(j)} \mid X^{(j-1)}) = \int P_\theta(X^{(j)} \mid \tilde{X})\, \mathcal{C}(\tilde{X} \mid X^{(j-1)})\, d\tilde{X}$ on X. Their Theorem 1 states that if (i) P_θ is a consistent estimator of the true posterior $p(X \mid \tilde{X})$ induced by the joint $p(X)\, \mathcal{C}(\tilde{X} \mid X)$, and (ii) T is ergodic (positivity of P_θ and $\mathcal{C}$ on a bounded volume is sufficient), then the asymptotic distribution of Eq. (23) converges to the data-generating distribution $p(X)$.

Translated to a latent flow-matching generator at fixed sampling step k, the identifications $\mathcal{C} \leftrightarrow R_{\sigma_k}$ (re-noising at level σ_k) and $P_\theta \leftrightarrow D_{\sigma_k}$ (the implicit conditional induced by the denoiser/clean-prediction estimator $\hat{z}_0(\cdot, t_k, c)$) give a perturb-and-redenoise iteration on z_k at fixed σ_k. Section C.2 extends this primitive to the partial-observation *conditional* setting that *GibbsLens*'s inner loop targets.

C.2 Sub-state ergodicity and stationary distribution

This subsection proves Proposition 1 from Section 3.1 by reduction to a sub-state instance of Bengio et al. [18] Theorem 1. Throughout we hold $\sigma_t > 0$, mask M, warped latent $\tilde{z}_0$, and pin $\bar{z}_t = (1-\sigma_t)\tilde{z}_0 + \sigma_t\, \xi_{\text{obs}}$ fixed; we write $z|_M$ and $z|_{1-M}$ for the corresponding restrictions of any latent tensor.

Lemma 1 (Sub-state DAE chain)**.** *With $\bar{z}_t|_M$ held as fixed exogenous context and σ_t held fixed, the inner-loop chain Eqs. (8)–(9) restricted to coordinates in $1-M$ is a generalized DAE chain in the sense of Bengio et al. [18], with substitutions $X \leftrightarrow z_0|_{1-M}$ and $\tilde{X} \leftrightarrow \tilde{z}|_{1-M}$.*

Proof of Lemma 1. The corruption restricted to $1-M$ is $\mathcal{C}(\tilde{z}|_{1-M} \mid z_0|_{1-M}) = \mathcal{N}\big((1-\sigma_t)\, z_0|_{1-M},\, \sigma_t^2\, I\big)$, a full-support Gaussian on $z_0|_{1-M}$. The conditional reconstruction $P_\theta(z_0|_{1-M} \mid \tilde{z}|_{1-M},\, \tilde{z}|_M{=}\bar{z}_t|_M)$ is the implicit conditional induced by the denoiser $u_\theta(\cdot, t, c)$ applied to the perturbed iterate with the observed coordinates pinned at $\bar{z}_t|_M$. Coordinates in M are never re-noised during the inner loop; they remain pinned to $\bar{z}_t|_M$, so the chain on $1-M$ is a generalized DAE chain with $\bar{z}_t|_M$ as fixed exogenous context. □

Proof of Proposition 1. By Lemma 1, the chain on $1-M$ is a generalized DAE chain with $\bar{z}_t|_M$ as fixed exogenous context. Bengio et al. [18] Theorem 1 states that if the reconstruction conditional P_θ is consistent with the true posterior $p(X \mid \tilde{X})$ induced by the joint $p(X)\,\mathcal{C}(\tilde{X} \mid X)$, and the transition operator is ergodic on a bounded volume, then the chain's asymptotic distribution converges to the data-generating distribution $p(X)$. In our setting, the joint at level σ_t is conditioned on $\bar{z}_t|_M$, so $p(X) \leftrightarrow p(z_0|_{1-M} \mid \bar{z}_t|_M)$. Under assumption (A1), $P_\theta \leftrightarrow u_\theta$ is consistent with this conditional joint; under (A2), positivity of the perturbation kernel on $1-M$ gives ergodicity. Therefore the clean-prediction iterates converge in distribution to $p(z_0|_{1-M} \mid \bar{z}_t|_M)$. □

Connection to coordinate descent / block Gibbs sampling. The chain restricts the perturb-and-redenoise primitive to a fixed coordinate block $1-M$ while holding M frozen. This is the generative-modeling analog of block-coordinate descent in optimization, in which a fixed subset of variables is updated at each iteration with the rest held constant; here "descent" on a loss is replaced by "mixing" of an implicit prior. The use of a fixed block (rather than random or systematic scans across blocks) is a deliberate task-driven choice: the partition $(1-M, M)$ is given by the camera-control mask geometry and is the same at every probe j.

C.3 Comparison with SRVS

The pseudo-Gibbs chain primitive is shared with Self-Refining Video Sampling (SRVS, Jang et al. [26]). At a fixed denoising step with noise level σ_k, SRVS iterates

$$z_k^{(j)} \;=\; (1-\sigma_k)\,\hat{z}_0^{(j-1)} \;+\; \sigma_k\,\xi^{(j)}, \qquad \hat{z}_0^{(j)} \;=\; z_k^{(j)} \;-\; \sigma_k\,u_\theta(z_k^{(j)}, t_k, c), \qquad \xi^{(j)} \sim \mathcal{N}(0, I). \tag{24}$$

With identifications $X \leftrightarrow \hat{z}_0$, $\tilde{X} \leftrightarrow z_k$, $\mathcal{C} \leftrightarrow R_{\sigma_k}$ and $P_\theta \leftrightarrow D_{\sigma_k}$, Eq. (24) is a structural instance of Eq. (23): SRVS uses Bengio's pseudo-Gibbs primitive at the chain level. The original SRVS paper invokes this informally — it does not state or apply Theorem 1 of Bengio et al. [18] explicitly — and motivates the practice as "mode-seeking" toward the data manifold. Beyond the chain primitive, SRVS adds an outer selective-refinement scheme applied between outer steps as a per-iteration mixing rule on the next-level noisy latent $z_{t_{i+1}}^{(k)}$, with the eligibility mask derived from the lag-1 self-disagreement statistic $U(z^{(k-1)}, z^{(k)}) = \frac{1}{C}\|D_\theta(z^{(k-1)}) - D_\theta(z^{(k)})\|_1$ thresholded at a fixed τ. Both the inner chain and the outer mixing rule act on every coordinate of the latent; SRVS does not condition the chain on any external observation.

Where *GibbsLens* differs. *GibbsLens* extends Bengio's primitive in five non-degenerate respects.

(i) Conditional vs. unconditional target (theorem correctness). The inner chain runs on the unobserved sub-state $z|_{1-M}$ with the observed coordinates pinned at the freshly noised observation $\bar{z}_t|_M = (1-\sigma_t)\tilde{z}_0|_M + \sigma_t\,\xi_{\text{obs}}|_M$. Proposition 1 extends Bengio's Theorem 1 to the sub-state setting and proves that the chain's stationary distribution is the partial-observation conditional data law $p(z_0|_{1-M} \mid \bar{z}_t|_M)$. SRVS targets the unconditional $p(z_0)$ informally without a formal theorem statement or proof; *GibbsLens*'s claim is sharper, conditioned on the observed evidence, and proven.

(ii) Block-conditional vs. full-state chain (the algorithmic novelty). GibbsLens acts on the fixed coordinate block $1-M$ given by the warp-derived mask, and within $1-M$ updates 3D patches of latent voxels jointly. The patch partition is justified by the order-2-Markov locality validation of Section D and is what makes the sub-state chain feasible at video scale: the full-state chain on the $\sim 10^5$ Wan 2.2 latent sites would mix slowly per inner-iteration budget. SRVS does not partition the state; the chain is full-state and the only spatial structure is per-coordinate.

(iii) Mask source and mask placement. SRVS's eligibility mask is implicit (model self-disagreement) and applied *outside* the inner chain, as a mixing rule between outer steps. *GibbsLens*'s mask is external (camera-warp visibility) and defines the partition $(M, 1-M)$ that the inner chain itself respects: observed sites are pinned, unobserved sites are refined. The role the mask plays in the algorithm is categorically different in the two methods.

(iv) Convergence diagnostic. SRVS gates refinement by the lag-1 paired-probe self-disagreement $U(z^{(k-1)}, z^{(k)}) < \tau$, which requires a noise-aware $\tau(\sigma_k)$ to remain calibrated across the schedule. *GibbsLens* uses the per-patch Δ-Welford running-variance plateau detector Eq. (12) with a single relative threshold $\kappa \in (0, 1]$ that is scale-free at the patch level and works unchanged across all outer steps; the per-patch granularity matches the locality scale at which neighboring coordinates' chain trajectories are correlated.

(v) Readout from the inner loop. SRVS commits the last iterate $\hat{z}_0^{(K_f)}$ as the inner-loop output. *GibbsLens* consumes the Polyak-averaged readout $\bar{z}_0 = J_{\max}^{-1} \sum_j \hat{z}_0^{(j)}$, which feeds the FlowMatch Euler step a $\sqrt{\tau_{\text{int}}/J_{\max}}$-times more concentrated estimator of the stationary mean at one in-place accumulator's cost (Section E).

In short, the genuinely new theoretical contribution of *GibbsLens* is (i): the sub-state extension of Bengio's Theorem 1 with a pinned exogenous context, giving the inner chain a proven stationary distribution equal to the partial-observation conditional posterior. The genuinely new algorithmic contribution is (ii): the block-conditional Gibbs primitive on patches of the unobserved support, made adaptive by the per-patch Δ-Welford gate and locality-justified at video scale. Neither contribution exists in SRVS or in any prior generalized-DAE-based sampler we are aware of.

D Block-precision locality validation

Setup. Stack the N VAE-encoded videos into a tensor of shape (N, C, L, H, W), with C-channel voxel $z^{(n)}_{l,h,w} \in \mathbb{R}^C$ at grid position (l, h, w) of video n. Following Section 3.2, we work one axis at a time. For axis $\alpha \in \{L, H, W\}$, an *axis-aligned line stack* is built by permuting α to slot 1 and the channel dim to slot 2, then merging the two off-axis dimensions and the video index into one row index:

$$(N, C, L, H, W) \xrightarrow{\text{permute + flatten off-axis}} \big(N_s,\, |\alpha|,\, C\big), \qquad N_s \;=\; N \cdot \prod_{\alpha' \neq \alpha} |\alpha'|.$$

Each row of this $\big(N_s, |\alpha|, C\big)$ tensor is one *line* $\mathbf{X} = (X_1, \ldots, X_{|\alpha|}) \in \mathbb{R}^{|\alpha| \times C}$: the $|\alpha|$ voxels swept along α at one (video, off-axis location) pair, with $X_\beta \in \mathbb{R}^C$ the voxel at axis-position β. Treating the N_s rows as i.i.d. draws of the joint $\mathbf{X}$ — under a working assumption of stationarity along the off-axis dimensions — and flattening the last two dimensions gives the design matrix $\mathbf{F} \in \mathbb{R}^{N_s \times |\alpha| C}$ that the precision pipeline below consumes. For the benchmarked Wan 2.2 latent at 49 frames, 704×1280, $C = 48$, this yields $N_s \in [2.9 \times 10^4,\ 1.8 \times 10^5]$ depending on axis, comfortably $N_s \gg |\alpha| C$.

Block precision and conditional independence. Form the centered sample covariance

$$\widehat{\Sigma} \;=\; \tfrac{1}{N_s - 1}\, \mathbf{F}^\top \mathbf{F} \;\in\; \mathbb{R}^{|\alpha| C \times |\alpha| C},$$

where $\mathbf{F} \in \mathbb{R}^{N_s \times |\alpha| C}$ stacks the N_s flattened lines after per-coordinate standardization. Add a tiny relative ridge $\widehat{\Sigma}_\lambda = \widehat{\Sigma} + \lambda \operatorname{tr}(\widehat{\Sigma})/(|\alpha|C)\, I$ ($\lambda = 10^{-6}$) for positive-definiteness and form the precision $\widehat{\Omega} = \widehat{\Sigma}_\lambda^{-1}$ via numerically stable Cholesky factorization. Reshape $\widehat{\Omega}$ into a $|\alpha| \times |\alpha|$ array of $(C \times C)$ blocks $\{\widehat{\Omega}_{\beta\gamma}\}$ indexed by axis positions $\beta, \gamma \in \{1, \ldots, |\alpha|\}$. The classical Gauss–Markov result [22] extended block-wise reads

$$X_\beta \perp\!\!\!\perp X_\gamma \mid X_{-\beta\gamma} \iff \widehat{\Omega}_{\beta\gamma} = 0_{C \times C}, \tag{25}$$

where $X_{-\beta\gamma} = \{X_r : r \neq \beta, \gamma\}$ is the rest of the line.

Block partial correlation via canonical correlations. The natural unit-free statistic is the block partial correlation

$$\widehat{R}_{\beta\gamma} = -\widehat{\Omega}_{\beta\beta}^{-1/2}\, \widehat{\Omega}_{\beta\gamma}\, \widehat{\Omega}_{\gamma\gamma}^{-1/2} \in \mathbb{R}^{C \times C},$$

whose singular values $\rho_1 \geq \cdots \geq \rho_C \geq 0$ are the canonical partial correlations between X_β and X_γ after regressing out $X_{-\beta\gamma}$. The top canonical correlation $\rho_1(\widehat{R}_{\beta\gamma}) \in [0, 1]$ measures the strongest linear conditional dependence; vanishing of all ρ_i recovers the conditional-independence test of Eq. (25). On the diagonal $\beta = \gamma$, $\widehat{R}_{\beta\beta} = -I$ and all singular values equal 1 (the source of the $|\alpha| \times |\alpha|$ heatmap diagonal in Figure 2).

Decay metric. For each axis we report the cumulative tail mass of the squared top canonical partial correlation,

$$\eta(r) = \frac{\sum_{|\beta - \gamma| > r} \big(\rho_1(\widehat{R}_{\beta\gamma})\big)^2}{\sum_{\beta \neq \gamma} \big(\rho_1(\widehat{R}_{\beta\gamma})\big)^2} \in [0, 1]. \tag{26}$$

$\eta(r)$ is the fraction of squared off-diagonal partial-correlation mass at distance greater than r; sharp locality $\Leftrightarrow \eta(r) \to 0$ for some modest r.

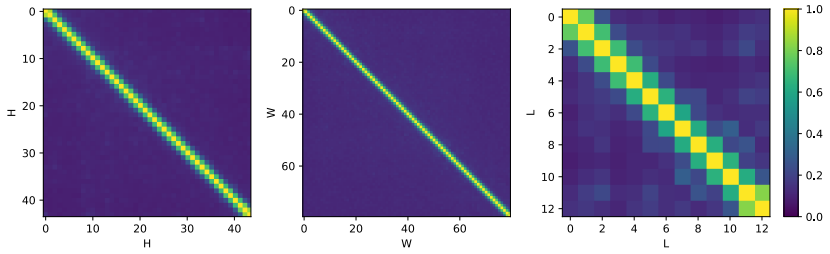
Noise floor. A finite-sample top canonical correlation between two unrelated C-vectors averages to $\mathbb{E}[\rho_1] \approx 2\sqrt{C/N_s}$ rather than zero. For our axes this gives floors of 0.033 (L), 0.061 (H), 0.082 (W); observed far-distance values are 0.07–0.10, within a factor of ~ 2 of the theoretical prediction. Most residual mass at large distances is sampling noise rather than true long-range coupling.

Reading Figure 2. Each axis gives a $|\alpha| \times |\alpha|$ heatmap of $\rho_1(\widehat{R}_{\beta\gamma})$. A clean local axis satisfies (i) $\rho_1 \geq 0.5$ at $|\beta - \gamma| = 1$, (ii) drop to $\leq 1/3$ of that value by $|\beta - \gamma| = 2$, (iii) floor near $2\sqrt{C/N_s}$ for $|\beta - \gamma| \geq 3$. Figure 2 satisfies all three on each of L, H, W for the Wan 2.2 latent, justifying the choice of patch radius ≥ 2 voxels in Section 3.2.

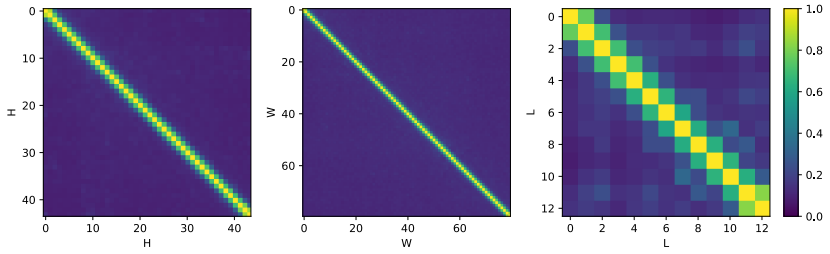
Locality across the noise schedule. The diagnostic above applies to clean VAE-encoded latents. We extend it to the model's clean prediction $\hat{z}_0(z_t, t, c)$ at non-zero noise levels, since the inner-loop perturb-and-redenoise of Eqs. (8)–(9) operates in clean-prediction space at the current noise level σ_t. We generate 200 videos by running the official Wan 2.2 flow-matching sampler from random initial noise $z_1 \sim \mathcal{N}(0, I)$ down to z_0, and record the full sampling trajectory $\{z_t\}$. At every recorded denoising step we evaluate $\hat{z}_0(z_t, t, c) = z_t - \sigma_t u_\theta(z_t, t, c)$ and run the block-precision pipeline of Eq. (25) on the resulting $\hat{z}_0$ tensors at $\sigma_t \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Figure 6 shows the resulting per-axis heatmaps at all five noise levels: the diagonal band stays sharp at every σ_t, with off-diagonal mass dropping into the finite-sample noise floor by distance $|\beta - \gamma| = 2$–3 on each of L, H, W, identical to the clean-latent diagnostic of Figure 2. The locality structure is therefore preserved across the entire sampling trajectory, not just on clean encoded latents, and the block-conditional Gibbs partition of Section 3.2 is valid at every active *GibbsLens* outer step.

E Polyak-averaged readout: compatibility with the FlowMatch Euler step

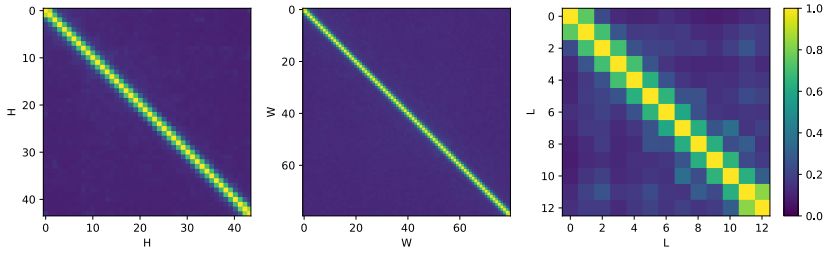
The write-back Eq. (10) consumes the running mean $\bar{z}_0 = J_{\max}^{-1} \sum_{j=1}^{J_{\max}} \hat{z}_0^{(j)}$ of the inner clean-prediction iterates, not the last iterate. The variance reduction argument is the standard MCMC iterate-averaging result [25]: at stationarity the running mean has variance $\sigma_\pi^2\, \tau_{\text{int}}/J + \mathcal{O}(J^{-2})$ vs. the last iterate's σ_π^2, where τ_{int} is the integrated autocorrelation time. We add only the observation



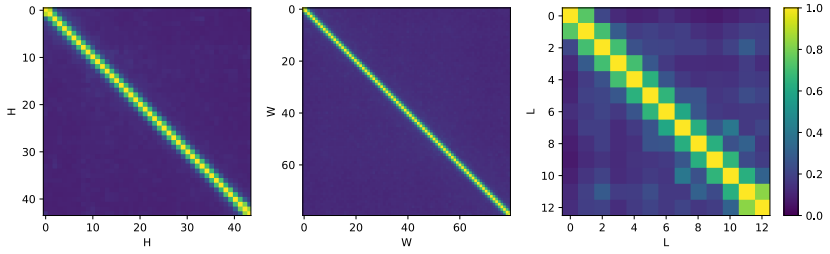
(a) $\sigma_t = 0.1$



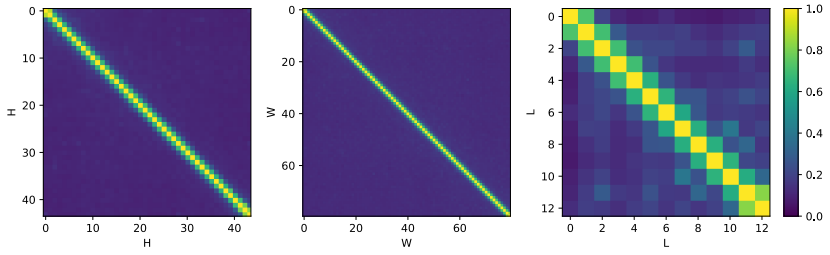
(b) $\sigma_t = 0.3$



(c) $\sigma_t = 0.5$



(d) $\sigma_t = 0.7$



(e) $\sigma_t = 0.9$

Figure 6: Top canonical partial correlation $\rho_1(\widehat{R}_{\beta\gamma})$ on the model's clean prediction $\hat{z}_0(z_t, t, c)$ along the H, W, L axes (left to right) at five noise levels. The diagonal band stays sharp at every σ_t with off-diagonal mass concentrated within $|\beta - \gamma| \leq 2$, confirming that the locality structure of Figure 2 is preserved across the sampling schedule.

that this estimator is the right input to feed into our outer integrator. The FlowMatch Euler update $z_{k+1} = z_k + (\sigma_{k+1} - \sigma_k)\,(z_k - \hat{z}_0^{\text{final}})/\sigma_k$ is *linear* in $\hat{z}_0^{\text{final}}$, so any extra variance in the readout is injected into the latent z_{k+1} without an averaging or projection step to absorb it; conversely, the bias/variance decomposition of the Polyak-averaged readout passes through the linear update unchanged. Consuming the running mean therefore hands the integrator a $\sqrt{\tau_{\text{int}}/J_{\max}}$-times more concentrated estimator of the stationary mean than the last iterate would, at the cost of one in-place accumulator per outer step — a strict improvement under the linearity that FlowMatch already provides.

F Toy-example details

We have provided the code of the toy example for reproducing all the figures in the paper.

Setup. The unconditional density is a 2D 8-square checkerboard: eight unit squares centered at $\{(i+0.5, j+0.5) : i, j \in \{-2, -1, 0, 1\}, (i+j) \bmod 2 = 0\}$, sampled uniform-area within each square. We train a 6-layer flow-matching MLP with sinusoidal time embedding and SiLU activations on this density for 20,000 iterations (batch 512, AdamW with $\text{lr} = 2\times10^{-3}$ and weight decay 10^{-4}, cosine schedule); the trained MLP attains a manifold-hit rate above 99% on unconditional samples, certifying that any conditional shortfall observed below is due to the conditional sampler rather than a poorly trained prior. We then evaluate conditional sampling under a noisy partial observation of the first coordinate $y \sim \mathcal{N}(z_0[1],\, \sigma_y^2)$ with $\sigma_y = 0.2$, anchored at $y_{\text{obs}} \approx 0.5$ where the vertical constraint line intersects exactly two of the eight filled squares, yielding a binary mode-coverage question with two well-separated conditional modes.

Evaluation. We compare *GibbsLens* against DPS [11] (compute-native at $\text{NFE} = 100$, 1 forward + 1 backward per step at 50 outer steps) and TFG-UGD [12] at $\text{NFE} = 250$ (configuration $N_{\text{recur}} = 2$, $N_{\text{iter}} = 1$, $\mu = 0.5$, $\rho = 0.5$, giving 5 calls per step at 50 outer steps). *GibbsLens* with inner budget $J = 4$ also costs 5 calls per step (1 outer forward + 4 inner forwards), so it is compute-matched to TFG-UGD at $\text{NFE} = 250$. For the matched sweep of Figure 1b, we vary $J \in \{0, \ldots, 16\}$ for *GibbsLens* and $N_{\text{recur}} \in \{1, \ldots, 6\}$ for TFG-UGD, plotting total NFE on the horizontal axis.

Each method is evaluated with 10 seeds and 500 samples per seed (5,000 samples total). The posterior-hit rate is the fraction of samples falling *both* inside one of the two filled squares intersected by the constraint line *and* within $|x_1 - y_{\text{obs}}| < 0.5$ of the anchor. A sample on the constraint line but outside the filled squares contributes zero to posterior-hit but may still contribute to manifold-hit (a sanity check that the sampler did not drift entirely off the data manifold).

Results. Figure 1c bins the inner iterations j by noise band and plots $|\Delta_W^{(j)}|$ (averaged over the population of toy chains) on a shared horizontal axis with the posterior-hit rate of Figure 1b; the two curves plateau over the same j-range, the empirical content of the claim that $|\Delta_W^{(j)}| \to 0$ is a reliable ground-truth-free trigger for the early-freeze gate. Disabling the inner loop ($J = 0$) reduces *GibbsLens* to DPS-like behaviour: with the inner Gibbs chain in Eqs. (8)–(9) disabled, only the outer soft pull of Eq. (5) remains, which at the $\sigma_y = 0.2$ used here is the DPS gradient evaluated at the flow-matching Tweedie mean (Appendix B.4). The *GibbsLens* sample cloud then collapses onto the same constraint-line support that DPS produces in Figure 1a, with mode coverage governed entirely by the outer pull — the clean ablation that isolates the inner loop as the source of the mode-coverage gain.

G *GibbsLens* implementation details

G.1 Hyperparameter Settings

Table 5 lists the necessary hyperparameters used in our main experiments with *GibbsLens* on the Wan 2.2 backbone. The discretized FlowMatch sampler runs for 50 outer steps, indexed so that step 0 corresponds to pure noise and step 50 to the clean sample. The camera-control window covers outer steps $t_s = 2$ through $t_e = 5$, the highest-noise interval where the warp constraint is most informative; at each step within this window the inner-loop refinement runs with budget $J_{\max} = 10$, with the Δ-Welford freeze gate (Section 3.2) allowing per-patch early termination so that the realised inner iteration count averages below $J_{\max}$. We use classifier-free guidance with scale 4.0.

Table 5: Hyperparameter settings for the main experiments (*GibbsLens* on the Wan 2.2 backbone).

Hyperparameter	Value	Description
Sampling steps	50	Total discretized FlowMatch outer steps
t_s	2	First outer step of the camera-control window
t_e	5	Last outer step of the camera-control window
$J_{\max}$	10	Inner-loop budget per outer step in the camera-control window
CFG scale	4.0	Classifier-free guidance scale

H Datasets and evaluation protocol

RealEstate10K. We follow the RealEstate10K [57] train/test split. **Sampling.** We sample 200 scenes uniformly at random, with a fixed seed for reproducibility. For each scene we treat the first frame as the I2V conditioning input and the remaining 48 frames as the held-out reference under the source camera trajectory. Reference-based metrics (LPIPS, SSIM) are computed between the generated video and the held-out reference frames after matching frame counts. The target trajectory is the one parsed from the dataset's pose annotations; it is fed to every method that takes camera parameters explicitly, and used to render the warp for every training-free method that takes a warp.

DAVIS controlled-trajectory synthesis We take the DAVIS-2017 challenge set, restricted to the 28 sequences with continuous moderate-to-large object motion. For each scene we synthesize three target trajectories of varying angular extent — approximately $\pm 15^\circ$, $\pm 30^\circ$, and $\pm 45^\circ$ horizontal sweep around the source camera — giving 84 controlled-trajectory videos in total. Each trajectory is realized as a smooth 49-frame sequence of 6-DOF camera poses; they are converted into a warp by the same depth-based lift-and-reproject pipeline used by every training-free baseline (see Appendix G). DAVIS lacks ground-truth novel views under the synthesized trajectories, so we do not report PSNR/LPIPS/SSIM and instead use FVD against the source distribution and the camera-pose-derived trajectory metrics on re-extracted poses (Section H.1).

H.1 Camera trajectory metrics

For every generated video we re-extract the 49-frame camera trajectory with Mega-SAM [42], a recent monocular SLAM tracker, without any tuning. The re-extracted trajectory is aligned to the target trajectory by a single similarity transform (translation, rotation, scale) and then evaluated by:

- **ATE** (Absolute Trajectory Error): per-frame root-mean-square translational error after alignment, in centimeters under the dataset's metric scale.
- **RRE** (Relative Rotation Error): per-pair root-mean-square rotational error between consecutive frames, in degrees.
- **RTE** (Relative Translation Error): per-pair root-mean-square translational error between consecutive frames, in centimeters.

Failures of Mega-SAM (re-extraction divergence on extreme dynamic content) are extremely rare on RealEstate10K and occur on $< 3\%$ of DAVIS trajectories; affected videos are dropped uniformly across all methods to keep the comparison fair, with the drop count reported per row.

H.2 Perceptual metrics

FVD. Standard Fréchet Video Distance [36] computed against a held-out reference distribution per dataset: source DAVIS clips on DAVIS, source RealEstate10K clips on RealEstate10K. Features come from the I3D backbone trained on Kinetics-400; we use the public implementation with default centering.

FVMD. Fréchet Video Motion Distance [38] compares distributions of explicit motion features derived from keypoint tracks. It complements FVD by assessing motion consistency; lower is better.

VisionReward. We report the overall video score (VR-Overall) from VisionReward [39], using the released weighted scoring implementation. Scores lie in $[-0.21645, 0.21645]$, with higher values

indicating stronger alignment with learned human preferences. This automatic metric complements, rather than replaces, human evaluation.

CLIP-v / CLIP-f. CLIP-based video coherence (CLIP-v, mean cosine similarity of frame-CLIP embeddings within the same video) and frame-text alignment (CLIP-f, mean cosine similarity between each frame's CLIP image embedding and the scene caption's CLIP text embedding). These are the two text-aligned consistency metrics commonly used.

LPIPS. Learned Perceptual Image Patch Similarity [58]: a reference-based perceptual metric that compares deep features (we use the standard AlexNet backbone) between two RGB frames, calibrated to human perceptual judgments. We report LPIPS only on RealEstate10K, where each generated frame can be compared to a held-out ground-truth view at the corresponding pose; DAVIS provides no such reference for novel-trajectory views. Lower is better.

SSIM. Structural Similarity Index [59]: a classical reference-based metric that compares the luminance, contrast, and structural components of two RGB frames in a sliding window. As with LPIPS, we report SSIM only on RealEstate10K against the held-out ground-truth view, computed per frame and averaged over the video. Higher is better.

I Additional quantitative results

I.1 Inference speed

Table 6 reports inference speed in frames/min on a single NVIDIA A40 GPU at each method's native operating resolution. *GibbsLens* runs at 7.13 frames/min at the full 720×1280 resolution shared by every training-free baseline—the second-fastest method overall, behind only TTM (12.99), which is bare hard-replacement without any inner-loop refinement and therefore represents the natural runtime ceiling. As quantified in Table 3, the inner-loop pseudo-Gibbs refinement increases runtime by 81.9% relative to TTM, with less than 0.01 GiB of additional peak GPU memory. Nevertheless, *GibbsLens* remains faster than the remaining training-free baselines (WorldForge 3.92, RePaint 6.23, Coarse-Guided 6.97) and faster than every training-based controller, even though the training-based methods operate at strictly lower resolution (ReCamMaster 5.55 at 480×832, TrajectoryCrafter 4.42 at 384×672, TrajectoryAttention 1.36 at 576×1024). Combined with the FVD gains of Tables 1–2, this places *GibbsLens* on the favourable side of the quality–speed Pareto frontier.

Table 6: Inference speed of training-free and training-based camera-control baselines and ours, reported at each method's native operating resolution on a single NVIDIA A40 GPU. † marks training-based methods.

	Resolution	Inference speed (frames/min)	Base video model
WorldForge [15]	720×1280	3.92	Wan 2.2
TTM [14]	720×1280	**12.99**	Wan 2.2
Coarse-Guided [16]	720×1280	6.97	Wan 2.2
RePaint [13]	720×1280	6.23	Wan 2.2
TrajectoryAttention† [34]	576×1024	1.36	Stable Video Diffusion
TrajectoryCrafter† [35]	384×672	4.42	CogVideoX
ReCamMaster† [1]	480×832	5.55	Wan 2.1
***GibbsLens* (Ours)**	720×1280	<u>7.13</u>	Wan 2.2

J Additional qualitative results

J.1 Failure-mode catalogue

Figure 7 compares the warped guidance video (with depth-induced wrong warping), the ground-truth target trajectory, and our *GibbsLens* output. Although the inner-loop refinement still produces visually plausible inpainting on the unobserved complement (the generated frames are seam-free

and natural-looking), the output tracks the wrongly warped guidance rather than the ground-truth trajectory. This is the expected behaviour under our partial-observation framing (Section 3): the warped video supplies the posterior evidence, and once that evidence is geometrically inconsistent the pretrained prior cannot correct it, since the prior encodes what plausible video looks like rather than what the correct camera trajectory should be. Robustness to severe depth misestimation therefore requires either a more reliable depth estimator upstream or a separate trajectory prior beyond the visual prior $p(z_0)$.



Figure 7: **Failure modes.** Severe depth estimation error produces wrong warping, leads to bad camera control.

J.2 RealEstate10K

Figure 8 extends the main-paper qualitative panel (Figure 3) with additional held-out RealEstate10K scenes spanning a broader range of trajectory extents.

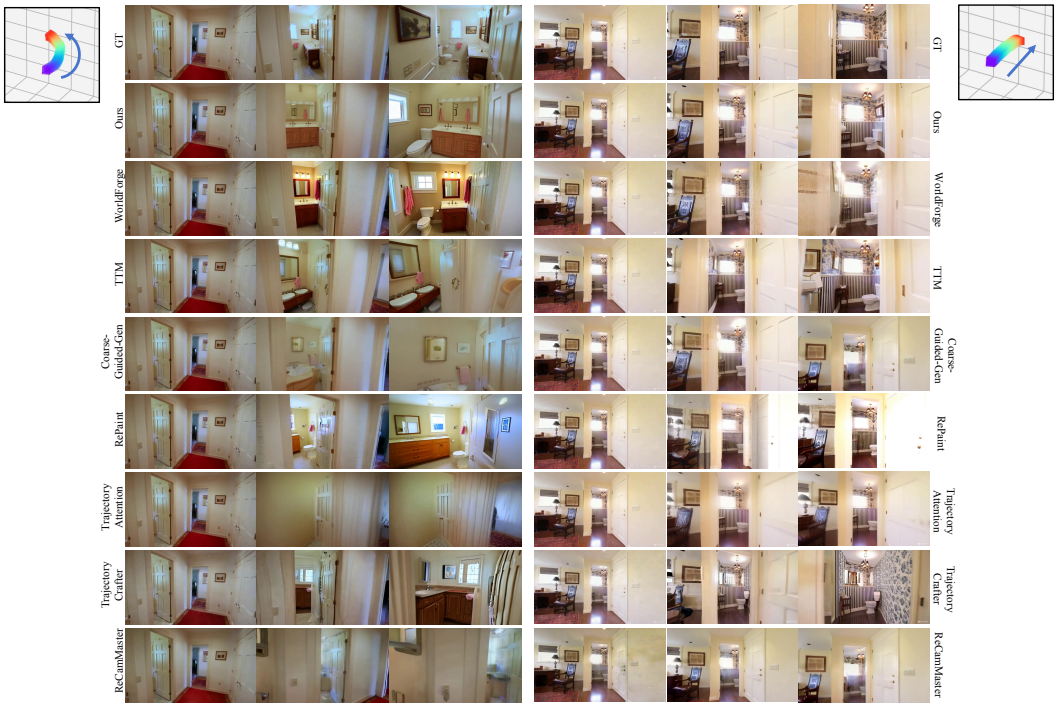


Figure 8: Additional RealEstate10K qualitative comparisons.

J.3 DAVIS

Figure 9 extends the main-paper DAVIS qualitative panel (Figure 4) with additional dynamic scenes under larger trajectory sweeps.

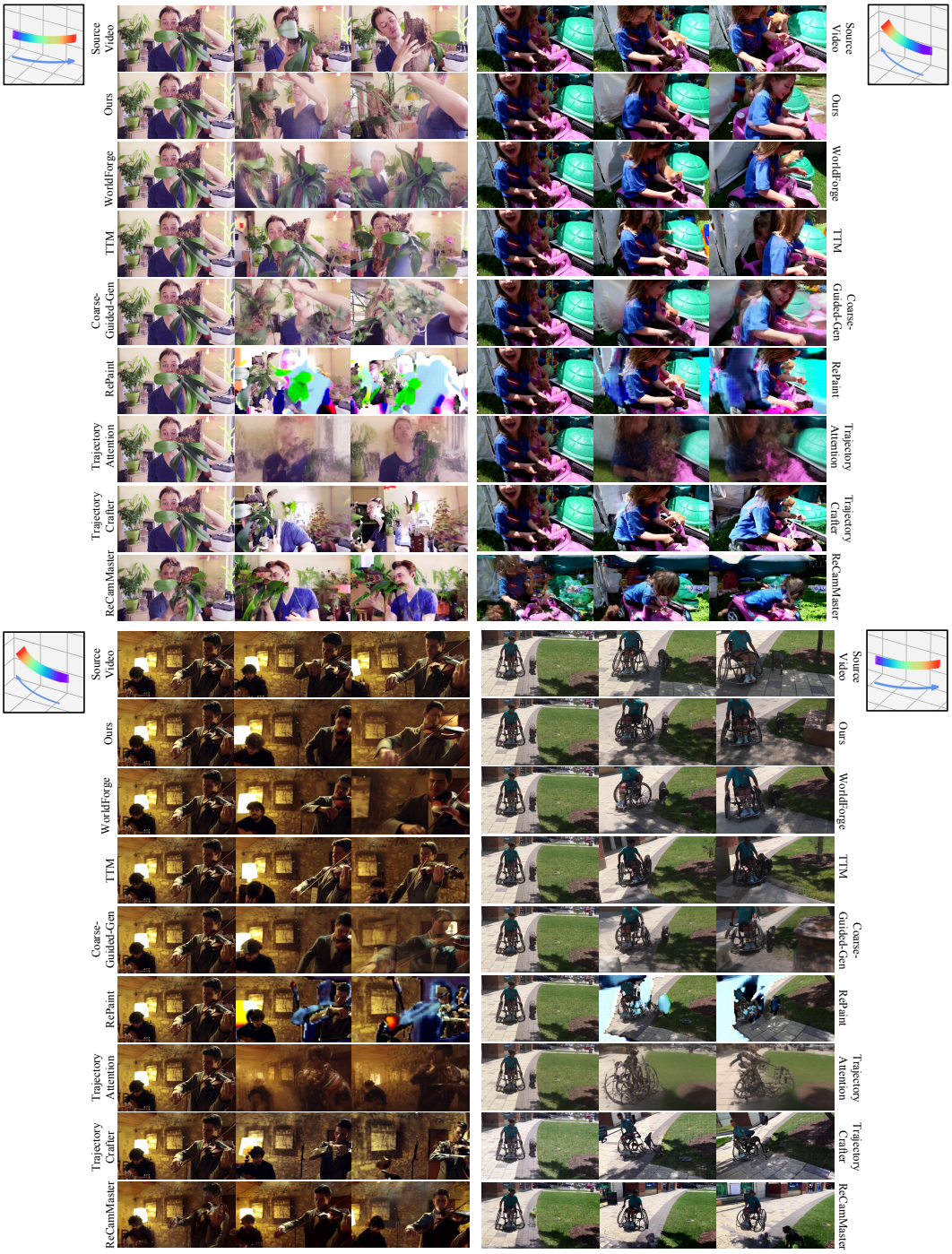


Figure 9: More comparison with state-of-the-art methods on Davis.

K Broader impacts

Positive impacts. Improved camera control for pretrained video generators has constructive applications in filmmaking previsualization, simulation and synthetic data for embodied AI, virtual- and augmented-reality content authoring, and education. Because *GibbsLens* is training-free and adds no new generative capability on top of the base model, it lowers the engineering barrier to high-quality camera-controlled video without requiring new large-scale training, which improves accessibility for users without the compute to fine-tune video generators.

Negative impacts. Higher-fidelity camera control on pretrained video generators can also make synthetic video more convincing, which inherits the well-known misuse profile of modern video generative models—non-consensual or deceptive synthetic media, impersonation, and disinformation. *GibbsLens* itself does not introduce new training data or new generative capacity; it operates as a sampler on top of an already-released backbone, so the risks it raises are the risks of the underlying generator combined with sharper trajectory control.

Mitigations. We rely on the safeguards of the underlying pretrained video generator (content filters, license restrictions, watermarking where applicable), release no new pretrained model or scraped dataset, and use only public benchmarks (RealEstate10K, DAVIS) under their original terms.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Section 1 state the three contributions—the *GibbsLens* sampler (Section 3), the block-conditional pseudo-Gibbs refinement, and the empirical results—which are substantiated by the derivations in Section 3 and the experiments in Section 4. Claims are scoped to training-free camera control on pretrained flow-matching video generators and are not asserted beyond this setting.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6 (Limitations) notes that the order-2 Markov property is calibrated to the Wan 2.2 latent and must be re-measured per backbone, and that the outer step inherits the geometric accuracy of the depth-and-reproject warping pipeline.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Section 3 states the theoretical assumptions, with full derivations and proofs in Appendices B.4 and C, including the conditional extension of Bengio et al. [18]. The paragraph preceding Section 4 explicitly distinguishes the asymptotic result in Proposition 1 from Algorithm 1's finite-budget implementation, for which no finite-step convergence guarantee is claimed.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We release all the experiment code. The method is training-free and built on the public Wan 2.2 (Text-Image-to-Video-5B) backbone with the default Wan 2.2 schedule; the depth-warp pipeline follows TTM and WorldForge; the full *GibbsLens* sampler is given as Algorithm 1. Section 4 specifies datasets (RealEstate10K split, DAVIS controlled-trajectory protocol), baselines, and evaluation metrics.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The experiment code is provided in the supplementary material with run scripts and configurations; the project page also links to the public sampler repository. RealEstate10K, DAVIS, and the Wan 2.2 backbone are publicly available, and Section 4 and Appendix H document the evaluation splits, trajectory-synthesis protocol, and metrics.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide the detailed implementation and hyperparameters of the algorithm in the Appendix. The experimental settings are presented in Section 4. We provide the full experimental code in the supplementary materials.

   Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: For the per-sample reference-based metrics on RealEstate10K (LPIPS, SSIM), Table 1 reports the mean across the 200 evaluation scenes. FVD, FVMD, VR-Overall, CLIP-v, CLIP-f and trajectory metrics aggregated across the benchmark are reported as single dataset-level scores, as is standard in this literature. We do not report error bars, confidence intervals, or statistical significance tests for these comparisons. Following common practice in this research area, we report a single dataset-level score for each metric without standard deviations.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 4 states that all experiments run on NVIDIA A40 GPUs in mixed precision on the Wan 2.2 (5B) backbone with the default Wan 2.2 sampling schedule.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and the research conforms with it. The work uses publicly available datasets and pretrained models, involves no human-subject research, and discusses potential misuse and mitigations in Appendix K.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

Answer: [Yes]

Justification: Appendix K discusses positive impacts on accessibility of camera-controlled video generation for filmmaking, simulation, embodied AI, and content-authoring use cases, as well as risks shared with modern video generative models—non-consensual or deceptive synthetic media, impersonation, and disinformation—and corresponding mitigations.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: We release no new pretrained models or scraped datasets; the contribution is a training-free sampler that runs on top of the publicly released Wan 2.2 backbone, and we rely on that base model's existing release-time safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Section 4 and the appendix credit the datasets, backbone, baselines, and evaluation methods, including FVD, FVMD, VisionReward, LPIPS, and CLIP, which are used under their respective upstream terms. Relevant licenses include Apache-2.0 for Wan 2.2 and the code of TTM, WorldForge, Mega-SAM, FVMD, and VisionReward; BSD-2-Clause for LPIPS; MIT for CLIP; CC-BY-4.0 for RealEstate10K camera-pose metadata and Mega-SAM's non-software materials; and the CogVLM2 license for VisionReward-Video weights, with dataset videos subject to their original terms rather than these code licenses.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The only new asset is the *GibbsLens* sampler code, available in the public repository with an inference script and a README describing dependencies, checkpoint access, and guidance-video preparation. No new pretrained models or datasets are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing or research with human subjects; all evaluation uses automatic metrics on public benchmarks.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper involves no research with human subjects, so no IRB review is required.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not part of the core methodology of this research; any LLM use was limited to writing and editing assistance and did not affect the scientific content, methodology, or originality of the work.

    Guidelines:

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