SMOOTH RESIDUAL ADAPTATION FOR VIDEO DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Improving conditional video generation involves choosing the form of a correction as well as its strength. Conditional smoothness offers a structural bias for this choice, but imposing it on a model's existing response also restricts the conditional behavior it can express. We introduce smooth residual adaptation for video diffusion, which constrains the added correction while retaining the model's nonlinear response. The residual is linear in selected condition coordinates at a fixed sampling state, while its coefficients adapt to the evolving video. Gaussian projection characterizes the approximation cost of excluding higher-order responses, and finite-sampler sensitivity describes how successive corrections affect the output. Experiments across multiple tasks and datasets show that smooth residual adaptation can improve generative performance across different conditioning interfaces.

1 INTRODUCTION

Changing an object's appearance in a video requires an edit to follow that object while remaining compatible with its surroundings. An action-conditioned prediction similarly ties the effect of an input to the current scene. Video diffusion models represent these contextual dependencies through conditional predictions updated during denoising (Ho et al., 2022b; Blattmann et al., 2023). Improving their conditional behavior raises a structural question: how should a learned correction respond to the condition while adapting to the evolving video state?

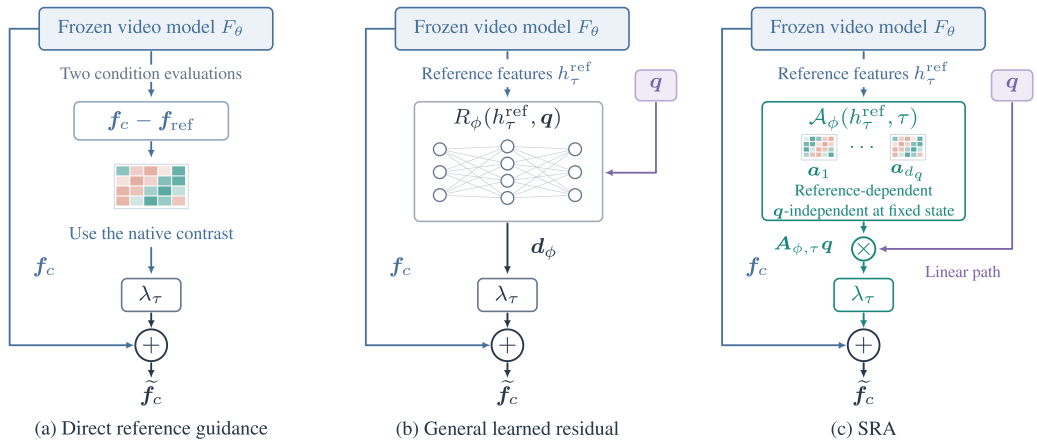
Guidance offers one way to change this response. Classifier-free guidance, perturbed-attention guidance, and autoguidance modify predictions through a scaled contrast, using an unconditional prediction, a perturbed forward pass, or a weaker model as the comparison (Ho & Salimans, 2022; Ahn et al., 2024; Karras et al., 2024). The contrast changes with the current state, while a scalar adjusts its overall strength. Figure 1(a) illustrates this choice with a reference condition. Transforming the components of that contrast allows a richer adjustment than scalar rescaling.

Learned conditioning pathways, such as ControlNet, T2I-Adapter, and IP-Adapter, provide further freedom to modify the response (Zhang et al., 2023; Mou et al., 2024; Ye et al., 2023). Figure 1(b) abstracts a learned correction as a joint function of state and condition; the cited methods realize conditioning through different internal architectures. This raises a choice about where to use nonlinear dependence. A correction can obtain its coefficients from the current video state while applying them linearly to selected condition coordinates. Restricting this added response leaves the model's existing nonlinear prediction available as the base for adaptation.

We introduce *Smooth Residual Adaptation* (SRA), which assigns nonlinear state dependence to a reference-dependent response operator. A reference evaluated at the current sampling state determines its coefficients; selected condition coordinates enter through a linear path. Figure 1(c) shows this separation. Recomputing the operator as sampling proceeds makes the correction responsive to the evolving video, while its dependence on the selected coordinates remains linear at each fixed state and reference. Smoothness refers to the added response's curvature in these coordinates. With prediction-difference coordinates, SRA learns a linear transformation of the contrast; actions, source-video components, and text representations provide other coordinates for the same construction.

The restriction makes the added response amenable to analysis. Under a Gaussian coordinate law, its approximation error separates into a missing mean, an error in the fitted operator, and higher-order response components controlled by curvature. A separate analysis follows the correction through a

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



(a) Direct reference guidance (b) General learned residual (c) SRA

Figure 1: Three conditional corrections at fixed state, time, and reference. (a) Direct guidance scales $f_c - f_{\text{ref}}$. (b) A general residual jointly depends on reference features h_τ^{ref} and condition coordinate q; this is a functional abstraction of a learned correction, not the internal architecture of the cited methods. (c) SRA determines $A_{\phi,\tau}$ from reference features independently of q, then applies $A_{\phi,\tau} q$; its grids illustrate response columns a_i in the output layout. All three retain f_c before native guidance and sampling. With native prediction-difference coordinates, (a) is scalar rescaling and (c) a learned linear transformation.

finite sampler, showing how guidance and subsequent updates transform its effect. Together, these analyses distinguish the cost of restricting a response from the effect of deploying it; generation quality is assessed through the resulting videos.

Experiments demonstrate gains over frozen parent models across video tasks. The base correction improves perceptual and distributional quality in held-out action-conditioned prediction without direction adjustment. Further gains in editing and text- and image-conditioned generation demonstrate the practical reach of the shared construction. Comparisons of reference routes, operator rank, and correction directions examine the contribution of its components. We make three contributions:

- We characterize the approximation error of condition-linear corrections under a Gaussian coordinate law and derive the first-order effect of inserted corrections on the output of a finite sampler.
- We introduce smooth residual adaptation, which combines a reference-dependent response operator with a linear condition path in a shared construction across video conditioning interfaces.
- We demonstrate gains over frozen parent models in prediction, editing, and text- and image-conditioned generation, including a formal held-out test of the base correction, and assess the roles of the reference, operator structure, and correction direction.

2 Related Work

Video modeling must account for both appearance and temporal evolution (Ho et al., 2022b; Singer et al., 2023; Blattmann et al., 2023; Yang et al., 2025; Shen et al., 2026). World models and causal denoising extend video priors to interaction and evolving context (Bruce et al., 2024; Chen et al., 2024); image-to-video adaptation and structure-preserving editing reuse learned representations for new conditions (Wu et al., 2023; Guo et al., 2024; Geyer et al., 2024; Qi et al., 2023; Esser et al., 2023). Our question concerns the conditional function of the added response, rather than a new video backbone or task-specific conditioning modality.

The closest architectural precedents are conditional parameter generation and additive adaptation. HyperNetworks generate network weights from another network (Ha et al., 2017), and FiLM generates feature-wise affine modulation from conditioning information (Perez et al., 2018). Side-Tuning adds a trainable network to a frozen representation (Zhang et al., 2020). Adapters and LoRA restrict the trainable parameterization or update rank (Houlsby et al., 2019; Hu et al., 2022); ControlNet,

T2I-Adapter, and IP-Adapter provide additional condition pathways (Zhang et al., 2023; Mou et al., 2024; Ye et al., 2023). AID adapts image-to-video models for instruction-guided prediction through learned conditioning and spatial–temporal adapters (Xing et al., 2025). SRA instead constrains the terminal correction: reference features determine its coefficients, and the selected coordinate enters linearly.

Guidance is the closest functional comparison. Classifier-free guidance, composition, and posterior sampling modify pretrained fields using conditional contrasts or constraints (Dhariwal & Nichol, 2021; Ho & Salimans, 2022; Liu et al., 2022; Chung et al., 2023). Guidance scheduling and autoguidance change when and from which model a correction is obtained (Kynkäänniemi et al., 2024; Karras et al., 2024). A response-trained SRA head approximates a scaled reference contrast, so a directly calibrated contrast is an essential comparator; replacing the whole conditional prediction with an affine surrogate tests a different question. Diffusion models can also be fine-tuned using rewards assigned to generated samples or pairwise preference data, through reinforcement learning, direct preference optimization, or stochastic optimal control (Black et al., 2024; Wallace et al., 2024; Domingo-Enrich et al., 2025).

Field structure alone does not determine the sampled distribution. Diffusion and flow analyses connect approximation error to generation under assumptions on both the field and the evolving state law (Song et al., 2021b; Lipman et al., 2023; Chen et al., 2023; Benton et al., 2024); guidance analyses likewise distinguish a prediction identity from a distributional guarantee (Du et al., 2023; Bradley & Nakkiran, 2025). Our analysis uses Gaussian projection to describe the restricted response class and sampler sensitivity to follow its deployment. Causal-Adapter and causal video steering instead specify causal models or graph-based constraints on the desired changes (Tong et al., 2025; Spyrou et al., 2025). Appendix G discusses further design context.

3 Theoretical Analysis

The analysis follows the correction from its conditional response to its effect on sampling. At a fixed state, Gaussian projection quantifies the approximation cost of a linear condition path. Along a sampling trajectory, a separate sensitivity calculation describes how guidance and later updates propagate the inserted corrections.

3.1 Video diffusion preliminaries

Write the video latents as a tensor $\mathsf{X}_\tau \in \mathbb{R}^{T_z \times C_z \times H_z \times W_z}$, with frame, channel, height, and width axes, and its fixed vectorization as $\boldsymbol{x}_\tau \in \mathbb{R}^{d_x}$, where $d_x = T_z C_z H_z W_z$; $\boldsymbol{x}_0$ is clean. Bold symbols denote vectors or matrices, with scalar components in ordinary type. Let $k \in \mathcal{K}$ be retained context, $c \in \mathcal{C}$ the varied native condition, and τ the noise time. A Gaussian training path is (Ho et al., 2020; 2022b)

$$\boldsymbol{x}_\tau = a_\tau \boldsymbol{x}_0 + b_\tau \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{1}$$

Here a_τ, b_τ are scalar schedule coefficients, $\boldsymbol{I}$ is the d_x-dimensional identity, and $\boldsymbol{\epsilon}$ is independent of the data. Write $\boldsymbol{f}_{\tau,c} := F_\theta(\boldsymbol{x}_\tau, \tau, k, c) \in \mathbb{R}^{d_{\text{out}}}$, abbreviated $\boldsymbol{f}_c$ at a fixed time, for the conditional prediction on the selected output block, with frozen parameters θ. Let $\boldsymbol{S}_{\text{out}}$ select that block; it is the identity for a full-block prediction. The target is $\boldsymbol{Y} = \boldsymbol{S}_{\text{out}}\boldsymbol{\epsilon}$ for noise regression and $\boldsymbol{Y} = \boldsymbol{S}_{\text{out}}\boldsymbol{x}_0$ for clean prediction. For $a_\tau = 1 - \tau$, $b_\tau = \tau$, it is $\boldsymbol{Y} = \boldsymbol{S}_{\text{out}}(\boldsymbol{\epsilon} - \boldsymbol{x}_0)$ in flow units, with generation proceeding from $\tau = 1$ to 0 (Lipman et al., 2023). A reference condition c_{ref} gives $\boldsymbol{f}_{\text{ref}} := F_\theta(\boldsymbol{x}_\tau, \tau, k, c_{\text{ref}})$ at the same state and time. Appendix C.1 gives the regression objective and distinguishes these outputs from an exact score.

3.2 Conditional curvature and response approximation

Fix $\xi = (\boldsymbol{x}_\tau, \tau, k, c_{\text{ref}})$. A coordinate map $q_\xi : \mathcal{C} \to \mathbb{R}^{d_q}$ gives $\boldsymbol{q} = q_\xi(c)$, with $q_\xi(c_{\text{ref}}) = \mathbf{0}$. The native condition need not itself be continuous. The base correction is $\boldsymbol{d}_\phi = \boldsymbol{A}_{\phi,\tau}\boldsymbol{q} \in \mathbb{R}^{d_{\text{out}}}$, where the operator $\boldsymbol{A}_{\phi,\tau} \in \mathbb{R}^{d_{\text{out}} \times d_q}$ has trainable parameters ϕ and depends on the state, time, and reference, but not on $\boldsymbol{q}$. The corrected prediction is $\widetilde{\boldsymbol{f}}_c = \boldsymbol{f}_c + \lambda_\tau \boldsymbol{d}_\phi$, with scalar strength λ_τ at time τ; we write λ when this time is fixed.

Let $\boldsymbol{Q}$ denote the random coordinate and $\boldsymbol{q}$ its value. The desired response is $T(\boldsymbol{q}) = \mathbb{E}_{\mathsf{P}_{\text{data}}}[\boldsymbol{R} \mid \boldsymbol{Q} = \boldsymbol{q}, \xi]$ for target error $\boldsymbol{R} = \boldsymbol{Y} - \boldsymbol{f}_c$ under the data law; a teacher determined by $\boldsymbol{q}$ defines T directly.

We study approximation by an arbitrary matrix $\boldsymbol{A} \in \mathbb{R}^{d_{\text{out}} \times d_q}$ before considering a fitted operator. For analysis, choose a version or extension of T on $\mathbb{R}^{d_q}$ and a separate measure $\nu_\Sigma = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma} \succ 0$. Under this measure, $\boldsymbol{Q} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}$, where $\boldsymbol{Z} \sim \gamma_{d_q} := \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d_q})$. Inner products and norms are Euclidean; $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm. Appendix A.1 lists object types and derivative conventions.

Lemma 1 (Curvature-controlled response approximation)**.** *Suppose $T_\Sigma(\boldsymbol{z}) = T(\boldsymbol{\Sigma}^{1/2}\boldsymbol{z}) \in W^{2,2}(\gamma_{d_q}; \mathbb{R}^{d_{\text{out}}})$, meaning that T_Σ and its weak derivatives through order two are square-integrable under γ_{d_q}. Set $\boldsymbol{\mu} = \mathbb{E}_{\nu_\Sigma} T(\boldsymbol{Q})$ and $\boldsymbol{A}_* = \mathbb{E}_{\nu_\Sigma}[T(\boldsymbol{Q})\boldsymbol{Q}^\top]\boldsymbol{\Sigma}^{-1}$. For any $\boldsymbol{A} \in \mathbb{R}^{d_{\text{out}} \times d_q}$,*

$$\mathbb{E}_{\nu_\Sigma}\|T(\boldsymbol{Q}) - \boldsymbol{A}\boldsymbol{Q}\|^2 = \|\boldsymbol{\mu}\|^2 + \|(\boldsymbol{A} - \boldsymbol{A}_*)\boldsymbol{\Sigma}^{1/2}\|_{\text{F}}^2 + \mathcal{R}_2, \tag{2}$$

$$\mathcal{R}_2 := \mathbb{E}_{\nu_\Sigma}\|T(\boldsymbol{Q}) - \boldsymbol{\mu} - \boldsymbol{A}_*\boldsymbol{Q}\|^2 \le \tfrac{1}{2}\mathbb{E}_{\gamma_{d_q}}\|D_{\boldsymbol{z}}^2 T_\Sigma(\boldsymbol{Z})\|_{\text{HS}}^2. \tag{3}$$

The squared Hilbert–Schmidt norm sums squared second derivatives over output components and input pairs; the factor $1/2$ is sharp.

The decomposition identifies what a correction anchored at the reference can approximate. The fitted operator controls the slope term, while the missing mean and higher-order components remain even for the optimal linear operator. Curvature bounds the latter under the Gaussian measure. The mean still matters when $T(0) = 0$: in one dimension, $T(q) = q^2$ with $Q \sim \mathcal{N}(0, 1)$ gives $\mathbb{E}T(Q) = 1$. The proof uses a standard Hermite expansion related to the second-order Poincaré inequality of Mossel & Neeman (2015, Lemma 3.3) and is given in Appendix A.2.

Here the Gaussian law is an analytical measure on condition coordinates, distinct from the diffusion noise. It is not an established model of the encoded actions, source videos, or text used in the experiments. The lemma characterizes representational error under that measure; Appendix A.5 relates it to random targets and weighted fitting. It does not compare the estimation errors of linear and nonlinear heads trained on finite data.

If the parent has a single-valued C^2 representation $f_\xi(\boldsymbol{q})$ in this coordinate, adding $\lambda\boldsymbol{A}\boldsymbol{q}$ leaves its conditional Hessian unchanged at fixed state, reference, and strength. The operator still determines the added response's first-order sensitivity. Appendix B.3 collects the implications for the complete model and its nonlinear variants.

3.3 Propagation through a finite sampler

Now fix the condition, random draws, schedule, and insertion strengths. For a sampler with N updates, scale all inserted corrections by $\eta \in [0, \bar{\eta}]$, where $\bar{\eta} > 0$ is the upper limit of the analyzed range, and write

$$\boldsymbol{s}_{j+1}^{(\eta)} = \mathcal{T}_j\big(\boldsymbol{s}_j^{(\eta)}, \eta d_j(\boldsymbol{s}_j^{(\eta)})\big), \quad j = 0, \dots, N-1, \qquad \boldsymbol{y}^{(\eta)} = \psi(\boldsymbol{s}_N^{(\eta)}). \tag{4}$$

Here $\boldsymbol{s}_j^{(\eta)}$ is the state at update j, with common initial state $\boldsymbol{s}_0$; $\eta = 0$ recovers the parent trajectory. The map d_j takes a state to a correction, includes the fixed dose, and is zero at inactive steps. The update $\mathcal{T}_j(\boldsymbol{s}, \boldsymbol{u})$ takes a state $\boldsymbol{s}$ and an injected correction $\boldsymbol{u}$ to the next state, including the parent prediction, output conversion, guidance, and numerical update. These state and correction spaces are finite-dimensional Euclidean spaces of compatible dimensions; the state includes any solver history or generated context. The output map ψ is the identity or a smooth decoder/feature map.

Proposition 1 (Endpoint response of a finite sampler)**.** *Assume $\mathcal{T}_j$, d_j, and ψ are C^2 on trajectory neighborhoods for $0 \le \eta \le \bar{\eta}$, with bounded first and second derivatives. Along the parent path, let $\boldsymbol{J}_j = D_{\boldsymbol{s}}\mathcal{T}_j(\boldsymbol{s}_j^{(0)}, \mathbf{0})$, $\boldsymbol{B}_j = D_{\boldsymbol{u}}\mathcal{T}_j(\boldsymbol{s}_j^{(0)}, \mathbf{0})$, and $\boldsymbol{d}_j^{(0)} = d_j(\boldsymbol{s}_j^{(0)})$. Then*

$$\boldsymbol{P}_j = D\psi(\boldsymbol{s}_N^{(0)})\boldsymbol{J}_{N-1}\cdots\boldsymbol{J}_{j+1}\boldsymbol{B}_j, \qquad \boldsymbol{w} = \sum_{j=0}^{N-1}\boldsymbol{P}_j\boldsymbol{d}_j^{(0)}, \tag{5}$$

$$\boldsymbol{y}^{(\eta)} = \boldsymbol{y}^{(0)} + \eta\boldsymbol{w} + \boldsymbol{r}_\eta, \qquad \|\boldsymbol{r}_\eta\| \le \tfrac{1}{2}C\eta^2, \tag{6}$$

where the Jacobian product is the identity for $j = N - 1$. Appendix A.3 derives the first variation; Appendix A.4 gives an explicit finite C.

The first-order output response is $\boldsymbol{w}$, which includes guidance and every remaining update. With $\boldsymbol{d}^{(0)}$ stacking the insertions $\boldsymbol{d}_j^{(0)}$ and $\boldsymbol{P} = [\boldsymbol{P}_0 \ \cdots \ \boldsymbol{P}_{N-1}]$, its squared size is $(\boldsymbol{d}^{(0)})^\top \boldsymbol{P}^\top \boldsymbol{P} \boldsymbol{d}^{(0)}$. Corrections of equal energy in the predictor's output space can therefore have different effects on the video. Multiple insertions may reinforce one another or cancel. For a paired target $\boldsymbol{y}_*$, the first-order change in squared error is $2\eta\langle \boldsymbol{y}^{(0)} - \boldsymbol{y}_*, \boldsymbol{w}\rangle$; improvement at a finite strength also requires the remainder to be small enough. This makes correction direction and insertion schedule relevant beyond the local fitting loss. Their effects on generation are evaluated after sampling; neither the linear condition path nor local calibration fixes the sign of the endpoint error change.

The calculation applies to smooth residuals generally; related sampler differentiation appears in Domingo-Enrich et al. (2025, Section 5.1.1). Recomputing the reference contributes to the remainder. Its local regularity assumptions and the effects of clipping and normalization are detailed in Appendix A.6. The derivative bounds have not been estimated for the deployed models, so the approximation at the deployed dose $\eta = 1$ remains unverified.

4 METHOD

SRA implements the separation between state dependence and conditional response using a reference evaluated at the current sampling state. The reference determines how the correction acts on the condition; recomputing it during denoising adapts that action to the evolving video. Figure 2 shows the resulting correction within guidance and sampling.

4.1 REFERENCE-DEPENDENT RESPONSE OPERATORS

Using the reference c_{ref} and coordinate $\boldsymbol{q}$ from Section 3.2, evaluate the parent's feature extractor H_θ at the *same current state*:

$$\begin{aligned} h_\tau^{\mathrm{ref}} &= H_\theta(\boldsymbol{x}_\tau, \tau, k, c_{\mathrm{ref}}), \quad & \boldsymbol{A}_{\phi,\tau} &= \mathcal{A}_\phi(h_\tau^{\mathrm{ref}}, \tau), \\ \boldsymbol{d}_\phi &= \boldsymbol{A}_{\phi,\tau}\boldsymbol{q}, & \widetilde{\boldsymbol{f}}_c &= \boldsymbol{f}_c + \lambda_\tau \boldsymbol{d}_\phi. \end{aligned} \tag{7}$$

Here $h_\tau^{\mathrm{ref}} \in \mathcal{H}$ is a collection of reference features; $\mathcal{A}_\phi$ maps that collection and time to the matrix $\boldsymbol{A}_{\phi,\tau}$. The calibrated strength λ_τ can be zero outside the insertion steps. Routing and contraction implement $\boldsymbol{A}_{\phi,\tau}$ without materializing a large dense matrix. The reference features h_τ^{ref} determine all nonlinear gates and attention weights; the condition coordinate enters only through the linear value path. A zero-initialized readout makes the initial model coincide with the parent. After training, $\boldsymbol{q} = \boldsymbol{0}$ still gives zero correction.

Figure 2 includes a learnable mask, giving $\widetilde{\boldsymbol{f}}_c = \boldsymbol{f}_c + \lambda_\tau[\boldsymbol{m}_\tau \odot \boldsymbol{d}_\phi]$, where $\odot$ denotes elementwise multiplication. The mask weights spatial locations or video frames and is broadcast and vectorized to $\boldsymbol{m}_\tau \in \mathbb{R}^{d_{\mathrm{out}}}$. Thus its effective operator is $\mathrm{diag}(\boldsymbol{m}_\tau)\boldsymbol{A}_{\phi,\tau}$. Its weights can be learned from the state, reference, and time, independently of $\boldsymbol{q}$ at a fixed state and reference, preserving the branch's coordinate-linearity. The action and text-token bases use $\boldsymbol{m}_\tau \equiv \boldsymbol{1}$.

4.2 ADAPTING ACROSS CONDITIONING INTERFACES

Table 1 gives two constructions of $\boldsymbol{q}$ for the same residual form in (7). Here E denotes a fixed condition encoding or projection; identity and standardization maps include vector controls. Each construction uses the same state, time, retained context, and reference as the operator.

A text-conditioned model can use differences in its text features or in the resulting predictions. For a video editor, the source can be held fixed while the text varies, or the edit instruction can be retained while a source component varies. Changing backbone changes feature dimensions, the operator architecture, and the output readout; the reference-only dependence of $\boldsymbol{A}_{\phi,\tau}$ and its linear action on $\boldsymbol{q}$ remain the shared construction.

Linearity is with respect to the selected coordinate. Encoded conditions and prediction differences may still depend nonlinearly on raw text. A prediction-difference coordinate is recomputed at every active insertion, with any fixed linear rescaling absorbed into its map. Tensor coordinates enter the analysis through a fixed vectorization, preserving their Frobenius norm. Appendix C.2 specifies the coordinate layouts used by these implementations.

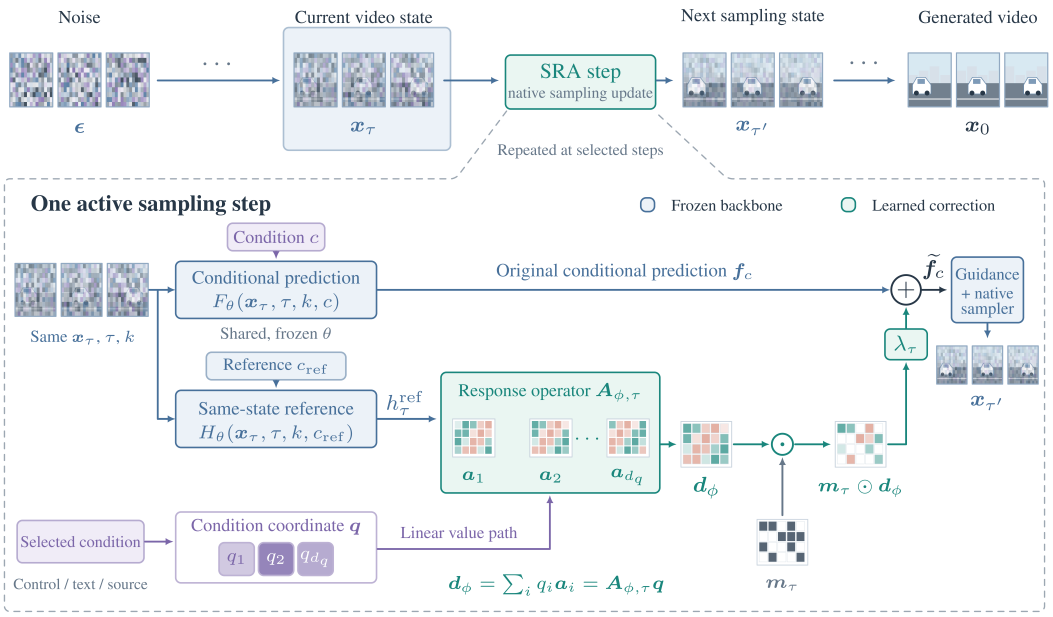


Figure 2: Smooth residual adaptation within video generation. At each active sampling step, the same current state supplies the conditional prediction and reference features. The reference determines the response columns $\boldsymbol{a}_i = \boldsymbol{A}_{\phi,\tau}\boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the ith coordinate basis vector; the condition supplies their linear weights q_i. The grids illustrate a shared output layout; the optional learnable mask $\boldsymbol{m}_\tau$ weights corresponding residual entries before scaling and addition. The circled dot denotes elementwise multiplication. Reference features and the operator are recomputed as sampling proceeds, with trained parameters fixed. Sampling steps run horizontally; the frames within each state represent video time. The image-conditioned variant fixes the first-slice mask; action and text-token bases use $\boldsymbol{m}_\tau \equiv \mathbf{1}$.

Table 1: Two coordinate constructions for smooth residual adaptation. Both vanish at the reference and enter $\boldsymbol{d}_\phi = \mathcal{A}_\phi(h_\tau^{\text{ref}}, \tau)\boldsymbol{q}$. The coordinate is a choice of representation; a backbone can support either construction.

Coordinate space	Construction of $\boldsymbol{q}$	Conditioning examples
Condition representation	$\text{vec}(E(c) - E(c_{\text{ref}}))$	Control vectors, text embeddings, source-video components
Native prediction	$\boldsymbol{f}_c - \boldsymbol{f}_{\text{ref}}$	Output response to any condition supported by the parent

Reference-dependent routing. Vector and source coordinates can be contracted with a learned slope field. For text tokens, take $E = E_\theta$, the parent's fixed embedding transformation, and write $\boldsymbol{U}_c = E_\theta(c) - E_\theta(c_{\text{ref}})$ with $\boldsymbol{q} = \text{vec}(\boldsymbol{U}_c)$. One realization uses reference queries and keys to route each head's values:

$$\begin{aligned} \boldsymbol{P}_{\ell h} &= \text{softmax}_{\text{key}}\left(\boldsymbol{Q}_{\ell h}^{\text{ref}}(\boldsymbol{K}_{\ell h}^{\text{ref}})^\top / \sqrt{d_{\text{head}}}\right), \\ \boldsymbol{M}_\ell &= \text{Concat}_{h=1}^{H_a}\left(\boldsymbol{P}_{\ell h}\boldsymbol{U}_c\boldsymbol{W}_{V,\ell h}\right)\boldsymbol{W}_{O,\ell}. \end{aligned} \tag{8}$$

Here ℓ indexes blocks and h the H_a attention heads of common width d_{head}; concatenation is along the feature axis. In the Wan implementation, E_θ and the value/output projections are frozen parent components. The adapter learns subsequent linear message projections and reference-dependent mixing. Appendix C.2 gives the matrix dimensions. Reference-only routing keeps this realization linear in $\boldsymbol{q}$; other backbones can use different contractions of the same coordinate.

Output coordinates. The readout can predict a correction in the parent's native units or use a fixed linear conversion at each noise level. For example, under $\boldsymbol{x}_\tau = (1-\tau)\boldsymbol{x}_0 + \tau\boldsymbol{\epsilon}$, a clean-latent

correction $\boldsymbol{d}_0$ becomes $-\boldsymbol{d}_0/\tau$ in velocity units for $\tau > 0$. Such a conversion can be absorbed into $\boldsymbol{A}_{\phi,\tau}$ without changing its dependence on $\boldsymbol{q}$. Appendix C.1 distinguishes flow velocity from VP v-prediction. Appendix C specifies the reported implementations.

4.3 Training and calibration

The head can learn from the parent's response or from its prediction error. Response regression fits $\alpha_\tau(\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}})$, where α_τ is a fixed output scaling. Prediction-error regression instead fits the random target $\boldsymbol{R} = \boldsymbol{Y} - \boldsymbol{f}_c$, whose conditional mean is the response $T(\boldsymbol{q})$ analyzed in Section 3.2. The target and prediction share the head's output coordinates; Appendix C specifies their conversion, normalization, and regularization. Exact response fitting gives

$$\widetilde{\boldsymbol{f}}_c = (1 + \lambda_\tau \alpha_\tau)\boldsymbol{f}_c - \lambda_\tau \alpha_\tau \boldsymbol{f}_{\mathrm{ref}}. \tag{9}$$

The learned head approximates this contrast within its operator class. Let $\boldsymbol{f}_{\mathrm{base}}$ denote the CFG baseline prediction, using null or negative conditioning at the same state, time, and output coordinates. When $\boldsymbol{f}_{\mathrm{ref}} = \boldsymbol{f}_{\mathrm{base}}$, exact fitting reduces to a classifier-free guidance (CFG) scale change. General references need not admit that reduction (Appendix B.2). A directly calibrated contrast tests the value of learning the restricted response.

With the head fixed, we calibrate strength against native conditional prediction error, before guidance. For unweighted squared loss, the capped nonnegative minimizer at each calibrated insertion is

$$\widehat{\lambda} = \mathrm{clip}\left(\frac{\sum_i \langle \boldsymbol{y}_i^{\mathrm{tar}} - \boldsymbol{f}_{c,i}, \boldsymbol{d}_i \rangle}{\sum_i \|\boldsymbol{d}_i\|^2}, 0, \Lambda\right), \tag{10}$$

where i indexes calibration states, $\boldsymbol{y}_i^{\mathrm{tar}}$ is the target, $\boldsymbol{d}_i$ the fitted correction, and $\Lambda \geq 0$ the strength cap. The operator clip truncates to $[0, \Lambda]$; zero-energy directions receive zero strength. Calibration can attenuate or reject a direction without reversing it. The exact-teacher case is analyzed in Appendix B.1; LucyEdit uses the corresponding region-weighted quadratic with preservation penalties (Appendix C.8).

This objective is a calibration proxy for generation. Under ordinary CFG with scale $s_{\mathrm{cfg}} > 0$, the deployed increment is $s_{\mathrm{cfg}}\lambda\boldsymbol{d}$ and guided error uses the prediction $\boldsymbol{f}_{\mathrm{CFG}}$, so its fixed-state minimizer generally differs from (10). Appendix B.2 derives that alternative; it is not the calibration used here. Heads and strengths are fitted before evaluation, with configuration choice on development rollouts (Appendix F.1).

4.4 Sampler integration

The native guidance and sampler receive the corrected conditional prediction; the CFG baseline prediction is unchanged. Each active insertion obtains the reference and any state-dependent coordinate at the current sampling state. A reference already evaluated by native guidance can be reused, as in LucyEdit; otherwise it requires an additional evaluation. Thus, in Section 3.3, d_j evaluates this correction, including its fixed strength, at sampler state $\boldsymbol{s}$ and $\mathcal{T}_j$ includes the subsequent coordinate conversion and guidance. Appendix C.6 reports head sizes and reference-evaluation costs; Appendix C.8 details the reused reference branch.

Direction adjustment at fixed energy. The base uses (7) directly. A separate AdaWorld variant decomposes $\boldsymbol{d} = \boldsymbol{p} + \boldsymbol{r}$, where $\boldsymbol{p}$ is its projection onto the difference between the last two parent clean-predicted frames and $\boldsymbol{r}$ is the remaining component. Given a base strength $b \geq 0$ and requested parallel strength $g \geq b$, it caps g to a feasible parallel strength a and solves $\|a\boldsymbol{p} + s\boldsymbol{r}\|^2 = b^2\|\boldsymbol{d}\|^2$ for $s \geq 0$, retaining the numerical cross term. The projection and energy-dependent coefficients make this variant generally nonlinear in $\boldsymbol{q}$. Unchanged and random equal-energy controls compare its directions. Section 5.1 identifies which comparisons use this adjustment. Appendix C.3 specifies this energy-preserving adjustment.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate action-conditioned prediction with AdaWorld (Gao et al., 2025) on LIBERO (Liu et al., 2023a) and the Franka and Jaco collections from Open X-Embodiment (Open X-Embodiment Collaboration et al., 2023); text-guided editing with Lucy-Edit-1.1-Dev (DecartAI Team, 2025) on OpenVE-Bench (He et al., 2025); and generation with CogVideoX-2B (Yang et al., 2025) and Wan2.2-I2V-A14B (Team Wan et al., 2025) on OpenVid-1M subsets (Nan et al., 2025). LIBERO prediction, LucyEdit, and both generation comparisons use the base correction. Franka and Jaco prediction, Jaco action transfer, and the LIBERO direction study use the optional energy-preserving adjustment. The shared construction is the reference-dependent operator and linear condition path; the coordinate, fitting objective, and insertion schedule follow each interface.

Each parent–SRA pair shares inputs, generation noise, and native sampler settings. LPIPS (Zhang et al., 2018), PSNR, SSIM (Wang et al., 2004), and motion error compare predictions with recorded futures; FVD (Unterthiner et al., 2018) measures distributional similarity. Editing uses category-specific instruction and video-quality ratings; generation also uses VBench dimensions (Huang et al., 2024). Appendix F gives data roles, aggregation, and uncertainty. Appendix C.6 reports trainable parameter counts and sampling costs.

5.2 RESULTS ACROSS VIDEO TASKS

Video prediction. SRA lowers LPIPS and FVD across all three prediction collections (Table 2). Held-out LIBERO gains from the base correction extend across task groups (Figure 3(a)): points average 50 clips for each of 30 tasks, vertical marks give group means, and the diamond gives the overall mean with a 95% task-bootstrap interval excluding zero. Appendix D.9 and Figure 5 provide the clip-level distribution and visual examples.

Table 2: Video prediction with AdaWorld and SRA. Better values are bold.

Dataset	Method	LPIPS↓	FVD↓	PSNR↑	SSIM↑	Motion error↓
LIBERO	AdaWorld	0.28174	640.4	**21.754**	0.7981	0.1867
	SRA	**0.28020**	**612.1**	21.753	**0.7987**	**0.1858**
Franka	AdaWorld	0.30313	1357.1	19.525	0.7420	0.3446
	SRA	**0.30219**	**1312.2**	**19.526**	**0.7424**	**0.3429**
Jaco	AdaWorld	0.29347	537.1	22.687	0.7536	0.2456
	SRA	**0.29336**	**503.7**	**22.689**	**0.7541**	**0.2421**

Action transfer. With initial observations fixed and action sequences substituted, SRA improves all three mean action-matching scores over AdaWorld on 109 Jaco episodes (Figure 3(b)). These compare proximity to substituted versus original actions (Appendix F.2). Points and densities show episode gains; diamonds give means with 95% episode-bootstrap intervals, and counts denote improved episodes. Appendix D.7 gives closed-loop Procgen results.

Text-guided editing. Trained on local modifications, SRA attains higher mean scores than LucyEdit in all eight OpenVE-Bench categories. Table 3 highlights four categories with increases in all three component means. Scores average three ratings; Overall covers all 431 inputs, with paired intervals for all categories and Overall in Appendix E.1.

Text- and image-conditioned generation. SRA improves five evaluated VBench dimensions on average with CogVideoX-2B; dynamic degree is unchanged (Figure 3(c)). Bars average eight fits on eight shared development inputs; error bars show sample standard deviations across fits. Points show individual fits (Appendix D.8).

With Wan2.2-I2V-A14B, SRA lowers FVD against the parent and matched random residuals across all eight fits on eight shared development inputs (Figure 3(d)), for both Full (33 frames) and Future (excluding the first frame). Small points denote fits; larger markers and error bars give means and

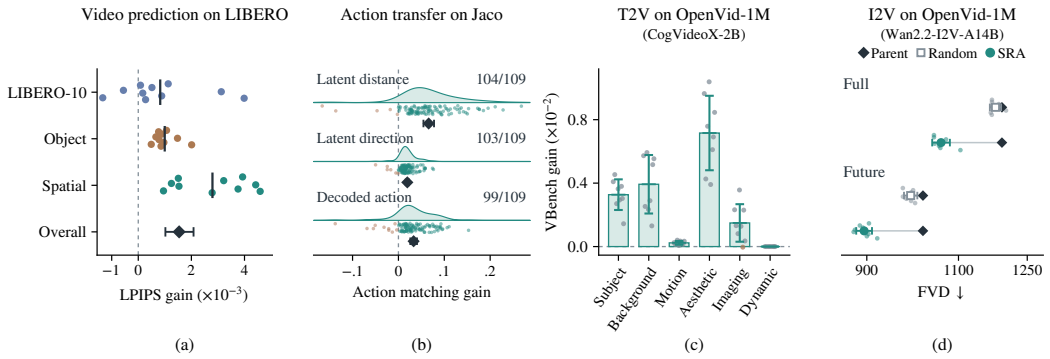


Figure 3: SRA across video tasks: (a) perceptual improvement in LIBERO prediction; (b) action matching after substituting Jaco action sequences; (c) VBench gains in text-to-video generation; (d) FVD in image-to-video generation.

Table 3: Text-guided editing on OpenVE-Bench (1–5; higher is better).

Category	LucyEdit↑	SRA↑
Style transfer	3.236	**3.299**
Local modification / replacement	3.564	**3.631**
Object addition	3.418	**3.502**
Creative editing	3.767	**3.900**
Overall	3.566	**3.616**

Table 4: Direction controls on LIBERO cross-demonstration rollouts. Lower is better.

Configuration	LPIPS↓	FVD↓
AdaWorld	0.315660	653.897
Without direction adjustment	0.314918	625.078
Energy-matched random direction	0.314919	625.216
SRA	**0.314476**	**615.130**

sample standard deviations, and diamonds denote the parent. Subject consistency and aesthetic quality also improve (Appendix D.5); Wan text-to-video results appear in Appendix D.3.

5.3 Ablations

With frozen heads and matched backbone calls, neutralizing both action routes improves LPIPS and FVD over either route alone on LIBERO training tasks; single-route references retain the current action. Direction adjustment also improves over unchanged and random energy-matched corrections (Table 4), supporting the importance of correction direction at equal local magnitude. This comparison averages three paired repetitions on 1,500 cross-demonstration pairs against recorded recipient trajectories (Appendix D.1). A rank-eight projection retains over 99.26% of the operator's squared Frobenius norm before rescaling, with similar LPIPS and FVD but no parameter compression. Direct guidance and nonlinear heads using the same reference information show no stable ordering on development inputs. Appendix D.10 gives the reference, rank, and comparator studies.

6 Conclusion

Smooth residual adaptation combines a linear condition path with a response operator that adapts to the evolving video state, providing a shared construction across conditioning interfaces. Its analysis characterizes approximation cost under a Gaussian coordinate law and propagation through a finite sampler. Experiments demonstrate gains over frozen parent models in prediction, editing, and text- and image-conditioned generation, including held-out prediction gains from the base correction alone. Reference and direction comparisons identify useful choices within the construction.

AI USE STATEMENT

We used generative AI tools to assist with literature search, drafting and polishing the writing, and auxiliary code (e.g., plotting and debugging); the authors reviewed and verified all such output. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work involves no human subjects or new data collection and uses existing pretrained models, datasets, and benchmarks cited in the paper. As with other video generation and editing methods, improved conditional control could be misused to create misleading videos.

REPRODUCIBILITY STATEMENT

Anonymized source code is provided in the supplementary material. Complete proofs are in Appendix A, with calibration and deployment analyses in Appendix B. Appendix C specifies model interfaces, objectives, and training settings. Appendices D–F describe the evaluation protocols, controls, aggregation, and uncertainty estimates.

REFERENCES

Donghoon Ahn, Hyoungwon Cho, Jaewon Min, Wooseok Jang, Jungwoo Kim, SeonHwa Kim, Hyun Hee Park, Kyong Hwan Jin, and Seungryong Kim. Self-Rectifying Diffusion Sampling with Perturbed-Attention Guidance. In *European Conference on Computer Vision*, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/09184.pdf.

Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.15571.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for World Modeling: Visual Details Matter in Atari. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.12399.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=uqQPyWFDhY.

Bernini Team, Chenchen Liu, Junyi Chen, Lei Li, Lu Chi, Mingzhen Sun, Zhuoying Li, Yi Fu, Ruoyu Guo, Yiheng Wu, Ge Bai, and Zehuan Yuan. Bernini: Latent Semantic Planning for Video Diffusion. *arXiv preprint arXiv:2605.22344*, 2026. URL https://arxiv.org/abs/2605.22344.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/14f75513f0f1ca01de1e826b52e6b840-Abstract-Conference.html.

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

Yochai Blau and Tomer Michaeli. The Perception-Distortion Tradeoff. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1711.06077.

Arwen Bradley and Preetum Nakkiran. Classifier-Free Guidance is a Predictor-Corrector. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=zrWNtzSZsf.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2211.09800.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative Interactive Environments. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. URL https://arxiv.org/abs/1705.07750.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2407.01392.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zyLVMgsZ0U_.

Hyungjin Chung, Jeongsol Kim, Michael Thompson McCann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2048–2056. PMLR, 2020. URL https://proceedings.mlr.press/v119/cobbe20a.html.

DecartAI Team. Lucy Edit: Open-Weight Text-Guided Video Editing, 2025. URL https://d2drjpuinn46lb.cloudfront.net/Lucy_Edit__High_Fidelity_Text_Guided_Video_Editing.pdf.

Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine*

Learning Research, pp. 8489–8510, 2023. URL https://proceedings.mlr.press/v202/du23a.html.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023. URL https://arxiv.org/abs/2302.03011.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/pdf/dd5c5803a1a63bd0d148c2be26a9ee612d1615f8.pdf.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. AdaWorld: Learning Adaptable World Models with Latent Actions. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.18938.

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the Content Bias in Fréchet Video Distance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2404.12391.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent Diffusion Features for Consistent Video Editing. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/069274129275ec1c6511abb5f6d96a31-Abstract-Conference.html.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/94894cf9bab20c00f6eaaff3b7f2be70-Abstract-Conference.html.

David Ha and Jürgen Schmidhuber. World Models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/pdf?id=rkpACe1lx.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/pdf/1912.01603.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.02193.

Haoyang He, Jie Wang, Jiangning Zhang, Zhucun Xue, Xingyuan Bu, Qiangpeng Yang, Shilei Wen, and Lei Xie. OpenVE-3M: A large-scale high-quality dataset for instruction-guided video editing. *arXiv preprint arXiv:2512.07826*, 2025. URL https://arxiv.org/abs/2512.07826v1.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/pdf?id=_CDixzkzeyb.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen Video: High Definition Video Generation with Diffusion Models. *arXiv preprint arXiv:2210.02303*, 2022a. URL https://arxiv.org/abs/2210.02303.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. In *Advances in Neural Information Processing Systems*, 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive Benchmark Suite for Video Generative Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.17982.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2206.00364.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a Diffusion Model with a Bad Version of Itself. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.02507.

Sergey Kastryulin, Jamil Zakirov, Denis Prokopenko, and Dmitry V. Dylov. PyTorch Image Quality: Metrics for Image Quality Assessment. *arXiv preprint arXiv:2208.14818*, 2022. URL https://arxiv.org/abs/2208.14818.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2Video-Zero: Text-to-Image Diffusion Models are Zero-Shot Video Generators. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2303.13439.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-Concept Customization of Text-to-Image Diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2212.04488.

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. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2404.07724.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/6940_ECCV_2022_paper.php.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/pdf?id=XVjTT1nw5z.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are Sample-Efficient World Models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.00588.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text Inversion for Editing Real Images using Guided Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2211.09794.

Elchanan Mossel and Joe Neeman. Robust dimension free isoperimetry in Gaussian space. *The Annals of Probability*, 43(3):971–991, 2015. doi: 10.1214/13-AOP860. URL https://arxiv.org/pdf/1202.4124.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 4296–4304, 2024. doi: 10.1609/aaai.v38i5.28226. URL https://ojs.aaai.org/index.php/AAAI/article/view/28226.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A Large-Scale High-Quality Dataset for Text-to-video Generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0396ca5a4c628936609aa819bfbca916-Abstract-Conference.html.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Open X-Embodiment Collaboration et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.09748.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1709.07871.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. FateZero: Fusing Attentions for Zero-shot Text-based Video Editing. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2303.09535.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2112.10752.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2208.12242.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating Oversaturation and Artifacts of High Guidance Scales in Diffusion Models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.02416.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.00512.

Yifan Shen, Peiyuan Zhu, Zijian Li, Shaoan Xie, Namrata Deka, Zongfang Liu, Zeyu Tang, Guangyi Chen, and Kun Zhang. Controllable video generation with provable disentanglement. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OcLNKpcY4J.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-Video Generation without Text-Video Data. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/pdf/89dbebc8608b7115596225380f7f411d9c944eaf.pdf.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *International Conference on Machine Learning*, 2015. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2010.02502.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1907.05600.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021b. URL https://arxiv.org/abs/2011.13456.

Nikos Spyrou, Athanasios Vlontzos, Paraskevas Pegios, Thomas Melistas, Nefeli Gkouti, Yannis Panagakis, Giorgos Papanastasiou, and Sotirios A. Tsaftaris. Causally steered diffusion for automated video counterfactual generation. *arXiv preprint arXiv:2506.14404*, 2025. URL https://arxiv.org/abs/2506.14404v1.

Team Wan et al. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=rimcq1oIFeR.

Lei Tong, Zhihua Liu, Chaochao Lu, Dino Oglic, Tom Diethe, Philip Teare, Sotirios A. Tsaftaris, and Chen Jin. Causal-Adapter: Taming text-to-image diffusion for faithful counterfactual generation. *arXiv preprint arXiv:2509.24798*, 2025. URL https://arxiv.org/abs/2509.24798v1.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2211.12572.

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. URL https://arxiv.org/abs/1812.01717.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wallace_Diffusion_Model_Alignment_Using_Direct_Preference_Optimization_CVPR_2024_paper.html.

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. URL https://ece.uwaterloo.ca/~z70wang/publications/ssim.html.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-A-Video: One-Shot Tuning of Image Diffusion Models for Text-to-Video Generation. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.11565.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive VideoGPTs are Scalable World Models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15223.

Zhen Xing, Qi Dai, Zejia Weng, Zuxuan Wu, and Yu-Gang Jiang. AID: Adapting image2video diffusion models for instruction-guided video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21243–21253, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/papers/Xing_AID_Adapting_Image2Video_Diffusion_Models_for_Instruction-guided_Video_Prediction_ICCV_2025_paper.pdf.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.06072.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. *arXiv preprint arXiv:2308.06721*, 2023. URL https://arxiv.org/abs/2308.06721.

Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-Tuning: A Baseline for Network Adaptation via Additive Side Networks. In *European Conference on Computer Vision*, 2020. URL https://arxiv.org/abs/1912.13503.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2302.05543.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. URL https://arxiv.org/abs/1801.03924.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2302.04867.

A Proofs and Additional Analysis

A.1 Notation and spaces

Vectors and matrices are bold, tensors use sans-serif capitals, and scalar components use ordinary type. Spaces and shapes distinguish random vectors from matrices. Maps such as F_θ, H_θ and $\mathcal{A}_\phi$ are distinct from their evaluated outputs. Hats denote estimates, tildes adapted predictions, and ref a reference condition; (0) denotes a zero-strength trajectory. Lowercase $\boldsymbol{q}$ is a coordinate value; $\boldsymbol{Q}$ denotes its random counterpart. The frame index t, noise time τ, sampler update j, calibration example i, and attention block ℓ have separate roles. Scalar strengths do not index video frames.

Table 5: Principal objects and their types. Dimensions refer to one example; batch axes are omitted.

Symbol	Meaning and space
$\boldsymbol{x}_0, \boldsymbol{x}_\tau, \boldsymbol{\epsilon}$	Clean latents, noisy latents, and Gaussian noise in $\mathbb{R}^{d_x}$
k, c, c_{ref}	Retained context, varied condition, and reference; their native domains depend on the interface
X_τ	Video latent tensor in $\mathbb{R}^{T_z \times C_z \times H_z \times W_z}$, with $\boldsymbol{x}_\tau = \mathrm{vec}(\mathsf{X}_\tau)$
$\boldsymbol{S}_{\mathrm{out}}$	Fixed output-block selector in $\mathbb{R}^{d_{\mathrm{out}} \times d_x}$
$\boldsymbol{q}, \boldsymbol{Q}, \boldsymbol{Z}$	Coordinate value, random coordinate, and standard Gaussian vector in $\mathbb{R}^{d_q}$
$\boldsymbol{f}_c, \boldsymbol{f}_{\mathrm{ref}}, \boldsymbol{f}_{\mathrm{base}}, \boldsymbol{Y}, \boldsymbol{R}$	Conditional, reference, and CFG baseline predictions, target, and error $\boldsymbol{R} = \boldsymbol{Y} - \boldsymbol{f}_c$, all in $\mathbb{R}^{d_{\mathrm{out}}}$
T	Desired mean response $\mathbb{R}^{d_q} \to \mathbb{R}^{d_{\mathrm{out}}}$ at fixed state and context
$\boldsymbol{A}_{\phi,\tau}, \boldsymbol{A}_*, \boldsymbol{A}_\omega$	Fitted, Gaussian-optimal, and weighted-optimal operators in $\mathbb{R}^{d_{\mathrm{out}} \times d_q}$
$\boldsymbol{\Sigma}, \boldsymbol{\mu}$	Gaussian coordinate covariance in $\mathbb{R}^{d_q \times d_q}$ and response mean in $\mathbb{R}^{d_{\mathrm{out}}}$
$h_\tau^{\mathrm{ref}}, H_\theta, \mathcal{A}_\phi$	Reference feature collection in $\mathcal{H}$, its frozen extractor, and its matrix-generating map
θ, ϕ	Frozen parent parameters and learned adapter parameters
$\lambda_\tau, \alpha_\tau, \Lambda$	Residual strength, teacher scaling, and strength cap; scalars
$\boldsymbol{s}_j^{(\eta)}, \boldsymbol{d}_j^{(0)}$	Sampler state in $\mathbb{R}^{n_j}$ and parent-path correction in $\mathbb{R}^{n_{u,j}}$
$\boldsymbol{y}^{(\eta)}, \boldsymbol{w}, \boldsymbol{r}_\eta, \boldsymbol{e}$	Endpoint output, its first variation, remainder, and paired error in $\mathbb{R}^{n_y}$
$\boldsymbol{P}_j$	Propagation from insertion j to the endpoint, in $\mathbb{R}^{n_y \times n_{u,j}}$
$\mathsf{P}_{\mathrm{data}}, \nu_\Sigma$	Data law and the separate Gaussian analysis measure $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$
ξ, q_ξ, T_Σ	Fixed context, its coordinate map, and whitened response $T_\Sigma(\boldsymbol{z}) = T(\boldsymbol{\Sigma}^{1/2}\boldsymbol{z})$
$\boldsymbol{m}_\tau, \mathcal{R}_2$	Broadcast, vectorized mask in $\mathbb{R}^{d_{\mathrm{out}}}$ and scalar higher-order approximation error

The output selector is included in F_θ's definition. During deployment, a correction to that block is inserted back into the parent's full output with zeros elsewhere. This selection is distinct from changing between noise, clean-latent, and velocity units. Tensor layouts and their fixed vectorizations are specified in Appendix C.2.

For $G : \mathbb{R}^a \to \mathbb{R}^b$, DG is a $b \times a$ Jacobian and D^2G is a vector-valued bilinear map, equivalently a $b \times a \times a$ array. Its squared Hilbert–Schmidt norm sums the squared entries. The operator norm instead takes the supremum over unit input directions. The Gaussian lemma uses the former; the sampler derivative bounds use the latter. The matrix square root $\boldsymbol{\Sigma}^{1/2}$ is the symmetric positive-definite root. For each response component T_r, whitening gives

$$D^2_{\boldsymbol{z}} T_{\Sigma,r}(\boldsymbol{z}) = \boldsymbol{\Sigma}^{1/2} D^2_{\boldsymbol{q}} T_r(\boldsymbol{\Sigma}^{1/2}\boldsymbol{z})\boldsymbol{\Sigma}^{1/2}.$$

Thus its curvature bound depends on the chosen coordinate covariance.

In the sampler, $d_j : \mathbb{R}^{n_j} \to \mathbb{R}^{n_{u,j}}$, $\mathcal{T}_j : \mathbb{R}^{n_j} \times \mathbb{R}^{n_{u,j}} \to \mathbb{R}^{n_{j+1}}$, and $\psi : \mathbb{R}^{n_N} \to \mathbb{R}^{n_y}$. Consequently, $\boldsymbol{J}_j \in \mathbb{R}^{n_{j+1} \times n_j}$ and $\boldsymbol{B}_j \in \mathbb{R}^{n_{j+1} \times n_{u,j}}$. The superscript in $\boldsymbol{d}_j^{(0)} = d_j(\boldsymbol{s}_j^{(0)})$ means evaluation on the parent path, not a zero correction. The derivative bounds $L_j, \beta_j, H_j, m_j, k_j, h_j, p_j, v_j, b_j, C$ are nonnegative scalars local to Appendix A.4; b_j bounds the state second derivative and v_j the correction-input first derivative, rather than a velocity prediction.

Paired output error. For $\boldsymbol{e} = \boldsymbol{y}^{(0)} - \boldsymbol{y}_*$, Proposition 1 also yields

$$\|\boldsymbol{y}^{(\eta)} - \boldsymbol{y}_*\|^2 - \|\boldsymbol{y}^{(0)} - \boldsymbol{y}_*\|^2 = 2\eta\langle \boldsymbol{e}, \boldsymbol{w}\rangle + \eta^2\|\boldsymbol{w}\|^2 + \rho_\eta, \tag{11}$$

$$|\rho_\eta| \le C\eta^2(\|\boldsymbol{e}\| + \eta\|\boldsymbol{w}\|) + \tfrac{1}{4}C^2\eta^4. \tag{12}$$

Improvement at a finite dose is certified only when the upper bound on this error change is negative. The loss compares outputs with a specified paired target; unpaired video-quality measures require a separate evaluation.

A.2 Proof of Lemma 1

Use normalized multivariate probabilists' Hermite polynomials H_α, indexed by $\alpha \in \mathbb{N}_0^{d_q}$, where $\mathbb{N}_0 = \{0, 1, \ldots\}$ and $|\alpha| = \sum_{i=1}^{d_q} \alpha_i$. They form an orthonormal basis of the square-integrable functions $L^2(\gamma_{d_q})$. Write $T_\Sigma = \sum_\alpha \boldsymbol{t}_\alpha H_\alpha$, with $\boldsymbol{t}_\alpha = \mathbb{E}_{\gamma_{d_q}}[T_\Sigma(\boldsymbol{Z})H_\alpha(\boldsymbol{Z})] \in \mathbb{R}^{d_{\mathrm{out}}}$. The constant coefficient is $\boldsymbol{t}_0 = \boldsymbol{\mu}$ and the matrix of degree-one coefficients is $\boldsymbol{M} = \mathbb{E}_{\gamma_{d_q}}[T_\Sigma(\boldsymbol{Z})\boldsymbol{Z}^\top] = \boldsymbol{A}_*\boldsymbol{\Sigma}^{1/2}$. Orthogonality gives (2), with $\mathcal{R}_2 = \sum_{|\alpha|\ge 2}\|\boldsymbol{t}_\alpha\|^2$. Let $\boldsymbol{e}_i$ be the multi-index with a one in position i and zeros elsewhere. For $\alpha_i > 0$, $\partial_i H_\alpha = \sqrt{\alpha_i} H_{\alpha - \boldsymbol{e}_i}$; the derivative is zero when $\alpha_i = 0$. Parseval applied to the second derivatives, evaluated at $\boldsymbol{Z}$, gives

$$\mathbb{E}_{\gamma_{d_q}}\|D^2 T_\Sigma\|_{\mathrm{HS}}^2 = \sum_\alpha \left[\sum_i \alpha_i(\alpha_i - 1) + \sum_{i\ne k}\alpha_i\alpha_k\right]\|\boldsymbol{t}_\alpha\|^2 = \sum_\alpha |\alpha|(|\alpha| - 1)\|\boldsymbol{t}_\alpha\|^2. \tag{13}$$

The last weight is at least two for $|\alpha| \ge 2$, proving (3). Polynomial truncation extends the argument to $W^{2,2}$; convergence holds in the derivative-weighted Hermite norm. A nonzero pure degree-two component attains equality, proving sharpness. For smooth T under these integrability conditions, Gaussian integration by parts also gives $\boldsymbol{A}_* = \mathbb{E}_{\nu_\Sigma}[D_{\boldsymbol{q}} T(\boldsymbol{Q})]$. □

A.3 First variation of the implemented update map

Differentiability follows by finite composition. Let $\boldsymbol{\xi}_j = \frac{\mathrm{d}}{\mathrm{d}\eta}\boldsymbol{s}_j^{(\eta)}\Big|_{\eta=0}$. Differentiating (4) gives

$$\boldsymbol{\xi}_0 = \boldsymbol{0}, \qquad \boldsymbol{\xi}_{j+1} = \boldsymbol{J}_j\boldsymbol{\xi}_j + \boldsymbol{B}_j\boldsymbol{d}_j^{(0)}. \tag{14}$$

Unrolling the recursion and applying $D\psi$ yields $\frac{\mathrm{d}}{\mathrm{d}\eta}\boldsymbol{y}^{(\eta)}\Big|_0 = \boldsymbol{w}$. The term Dd_j is absent from this first variation because it is multiplied by η at the parent trajectory. It contributes to the second-order remainder; the reference is not frozen across the perturbed rollout.

A.4 An explicit second-derivative bound

The following constants are uniform on the trajectory neighborhoods for $0 \le \eta \le \bar{\eta}$. Use operator norms for derivatives, with the Euclidean product norm for the joint $(\boldsymbol{s}, \boldsymbol{u})$ input; $\|\cdot\|_{\mathrm{op}}$ denotes the corresponding operator norm. Here $D_{\boldsymbol{s}}$ and $D_{\boldsymbol{u}}$ denote partial derivatives in the state and injected correction, respectively, and D^2 denotes the second derivative as a bilinear map:

$$\begin{aligned}
\|D_{\boldsymbol{s}}\mathcal{T}_j\|_{\mathrm{op}} &\le L_j, & \|D_{\boldsymbol{u}}\mathcal{T}_j\|_{\mathrm{op}} &\le \beta_j, & \|D^2\mathcal{T}_j\|_{\mathrm{op}} &\le H_j,\\
\|d_j\| &\le m_j, & \|Dd_j\|_{\mathrm{op}} &\le k_j, & \|D^2 d_j\|_{\mathrm{op}} &\le h_j,\\
\|D\psi\|_{\mathrm{op}} &\le L_\psi, & \|D^2\psi\|_{\mathrm{op}} &\le H_\psi.
\end{aligned}$$

The bounds on d_j exist after restricting to a compact trajectory tube. We bound the state first derivative by p_j, the correction-input first derivative by v_j, and the state second derivative by b_j, all with respect to η. Define nonnegative $p_0 = b_0 = 0$ and recursively

$$\begin{aligned}
p_{j+1} &= (L_j + \bar{\eta}\beta_j k_j)p_j + \beta_j m_j,\\
v_j &= m_j + \bar{\eta}k_j p_j,\\
b_{j+1} &= (L_j + \bar{\eta}\beta_j k_j)b_j + \beta_j(2k_j p_j + \bar{\eta}h_j p_j^2) + H_j(p_j^2 + v_j^2), &(15)\\
C &= L_\psi b_N + H_\psi p_N^2. &(16)
\end{aligned}$$

The recursion gives conservative bounds; these constants have not been measured for the video models.

To obtain the recursion, put $\boldsymbol{u}_j^{(\eta)} = \eta d_j(\boldsymbol{s}_j^{(\eta)})$, and use primes for derivatives in η. Below, d_j and its derivatives are evaluated at $\boldsymbol{s}_j^{(\eta)}$, and update derivatives at $(\boldsymbol{s}_j^{(\eta)}, \boldsymbol{u}_j^{(\eta)})$. Direct differentiation gives

$$\begin{aligned}
\boldsymbol{u}_j' &= d_j + \eta D d_j \boldsymbol{s}_j', \\
\boldsymbol{u}_j'' &= 2D d_j \boldsymbol{s}_j' + \eta D^2 d_j[\boldsymbol{s}_j', \boldsymbol{s}_j'] + \eta D d_j \boldsymbol{s}_j'', \\
\boldsymbol{s}_{j+1}' &= D_{\boldsymbol{s}} \mathcal{T}_j \boldsymbol{s}_j' + D_{\boldsymbol{u}} \mathcal{T}_j \boldsymbol{u}_j', \\
\boldsymbol{s}_{j+1}'' &= (D_{\boldsymbol{s}} \mathcal{T}_j + \eta D_{\boldsymbol{u}} \mathcal{T}_j D d_j) \boldsymbol{s}_j'' \\
&\qquad + D_{\boldsymbol{u}} \mathcal{T}_j (2D d_j \boldsymbol{s}_j' + \eta D^2 d_j[\boldsymbol{s}_j', \boldsymbol{s}_j']) + D^2 \mathcal{T}_j[(\boldsymbol{s}_j', \boldsymbol{u}_j'), (\boldsymbol{s}_j', \boldsymbol{u}_j')].
\end{aligned}$$

Induction proves $\|\boldsymbol{s}_j'\| \le p_j$, $\|\boldsymbol{u}_j'\| \le v_j$, and $\|\boldsymbol{s}_j''\| \le b_j$. Thus $\|(\boldsymbol{y}^{(\eta)})''\| \le C$. Integral Taylor expansion yields

$$\boldsymbol{r}_\eta = \int_0^\eta (\eta - a)(\boldsymbol{y}^{(a)})'' \, \mathrm{d}a, \qquad \|\boldsymbol{r}_\eta\| \le C\eta^2/2.$$

Expanding the squared target error gives $\rho_\eta = 2\langle \boldsymbol{e} + \eta \boldsymbol{w}, \boldsymbol{r}_\eta \rangle + \|\boldsymbol{r}_\eta\|^2$. Cauchy–Schwarz proves (12) and completes Proposition 1. □

The role of the finite-dose remainder. The smooth scalar example $y^\eta = 1 - \eta + 10\eta^2$, $y_* = 0$, has $ew = -1$, yet the loss increases from 1 to 100 at $\eta = 1$. The negative derivative at zero therefore does not determine the loss at the deployed dose.

A.5 Random targets and the implemented fitting objective

Fix $\xi = (\boldsymbol{x}_\tau, \tau, k, c_{\mathrm{ref}})$ and a matrix $\boldsymbol{A}$, and assume $\boldsymbol{R} = \boldsymbol{Y} - \boldsymbol{f}_c$ and $\boldsymbol{AQ}$ have finite second moments. Expectations in this subsection use the data law $\mathsf{P}_{\mathrm{data}}$ of the random condition coordinate $\boldsymbol{Q}$ and target $\boldsymbol{Y}$ at this fixed state; $T(\boldsymbol{q}) = \mathbb{E}[\boldsymbol{R} \mid \boldsymbol{Q} = \boldsymbol{q}, \xi]$ suppresses the dependence of T on ξ. Orthogonality of conditional expectation gives

$$\mathbb{E}\|\boldsymbol{R} - \boldsymbol{AQ}\|^2 = \mathbb{E}\|\boldsymbol{R} - T(\boldsymbol{Q})\|^2 + \mathbb{E}\|T(\boldsymbol{Q}) - \boldsymbol{AQ}\|^2. \tag{17}$$

Indeed, the cross term vanishes after conditioning on $\boldsymbol{Q}$. The first term is irreducible target variance at this state. If the coordinate marginal under this data law is ν_Σ and the Sobolev assumptions hold, Lemma 1 further decomposes the second term. Otherwise the lemma evaluates a different approximation risk; it cannot be substituted into this data-law identity. The variance term is zero when the teacher is determined by $\boldsymbol{Q}$ and the fixed variables. A teacher deterministic in a richer condition need not have zero variance after a lossy coordinate map. A realized noisy target is not itself the differentiable function to which the curvature lemma applies. Nor does $T(0) = 0$ imply a zero Gaussian mean; for an error target, even $T(0) = 0$ need not hold.

Energy-normalized training changes the projection. For a nonnegative sample weight ω with finite weighted moments and nonsingular $\boldsymbol{G}_\omega = \mathbb{E}[\omega \boldsymbol{Q}\boldsymbol{Q}^\top]$, minimizing the unregularized weighted squared loss gives

$$\boldsymbol{A}_\omega = \mathbb{E}[\omega \boldsymbol{R}\boldsymbol{Q}^\top]\boldsymbol{G}_\omega^{-1}. \tag{18}$$

Expanding the loss and setting its matrix derivative to zero proves the identity. If ω is a function of $\boldsymbol{Q}$, $\boldsymbol{R}$ can be replaced by $T(\boldsymbol{Q})$. If normalization depends on the random target as well, its conditional mean becomes $\mathbb{E}[\omega \boldsymbol{R} \mid \boldsymbol{Q}]/\mathbb{E}[\omega \mid \boldsymbol{Q}]$ wherever the denominator is positive. Thus energy normalization need not preserve the unweighted optimum. Parameter sharing across states, the head architecture, and regularization introduce further differences. Equations (17)–(18) are standard regression identities, not new convergence or generalization results.

The positive-definite $\boldsymbol{\Sigma}$ in the lemma defines an analytical reference measure. It is not an inverse empirical covariance estimated from the training set. For example, Wan's 512×1536 token coordinate is much larger than its number of training examples; a raw empirical covariance there is singular. The implemented routed operator neither constructs nor inverts this matrix. The Gaussian model also requires a response defined on its support, beyond the observed embeddings. A data conditional mean is defined only almost everywhere under its own coordinate law; evaluating it under ν_Σ requires

specifying a version or extension there. For a native-response coordinate $\boldsymbol{q} = \boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}}$ and teacher $T(\boldsymbol{q}) = \alpha\boldsymbol{q}$, the ideal higher-order term is already zero. In that interface, the lemma does not explain a benefit from excluding nonlinear teacher components; architectural restrictions and prediction-error fitting remain separate questions. Establishing that this model approximates the deployed condition distribution would require additional measurements.

A.6 Assumptions and relation to deployment

The Gaussian lemma varies $\boldsymbol{q}$ while holding the state fixed. The sampler proposition holds the external condition fixed while the dose changes the trajectory; a state-dependent coordinate can change with it. To combine them, one would need to control the joint state–condition law along the rollout and make the approximation error uniform there. Standardization, PCA, and token projection alone do not make the coordinates Gaussian; a draw from a Gaussian distribution can also fall outside the support of valid encoded conditions. A further approximation enters when a shared head trained with normalized losses and regularization is used in place of the ideal pointwise matrix $\boldsymbol{A}_*$.

The sampler derivatives are derivatives of the numerical mapping, including the recomputed reference. Detaching reference features in the training graph leaves their numerical dependence on the sampling state intact. Clipping, feasibility caps, and zero-norm branches can violate C^2 regularity, so the proposition is local to neighborhoods where its assumptions hold. A solver that evaluates the predictor several times can be represented by separate updates, retaining intermediate quantities in the state. The energy-adjusted variant can be treated this way only where its update maps have the required regularity; its nonlinear dependence on $\boldsymbol{q}$ remains.

The role of $\boldsymbol{P}^\top\boldsymbol{P}$ is visible even in a simple example. If $\boldsymbol{P} = \mathrm{diag}(2, 1)$, unit coordinate directions reach the output with squared lengths 4 and 1. Two insertions with $\boldsymbol{P}_0 = \boldsymbol{I}$ and $\boldsymbol{P}_1 = -\boldsymbol{I}$ can instead cancel to first order. Norm matching at insertion therefore leaves the final perturbation undetermined. These examples illustrate the mechanism; they do not predict the ordering of the measured direction comparisons.

A distributional guarantee needs more than the response restriction. For example, Benton et al. (2024, Theorem 1) bound Wasserstein error through integrated velocity error on the reference flow and spatial Lipschitz control. Zero curvature in a condition coordinate supplies neither condition. Applying such a result here would also require the native guidance, discretization, and path distribution to be accounted for. Even then, reducing an upper bound would not by itself prove a decrease in the actual distance.

B Calibration and Deployment Scope

We use the fixed-state construction of Section 3.2: $\boldsymbol{d} = \boldsymbol{d}_\phi = \boldsymbol{A}_{\phi,\tau}\boldsymbol{q}$ and $\widetilde{\boldsymbol{f}}_c = \boldsymbol{f}_c + \lambda\boldsymbol{d}$, suppressing the time subscript on λ_τ. The conditional prediction $\boldsymbol{f}_c$, reference prediction $\boldsymbol{f}_{\mathrm{ref}}$, target $\boldsymbol{Y}$, and correction $\boldsymbol{d}$ share the same output coordinates. The head parameters ϕ are fixed during calibration.

B.1 What the calibration rule minimizes

Let i index calibration samples $(\boldsymbol{f}_{c,i}, \boldsymbol{y}_i^{\mathrm{tar}}, \boldsymbol{d}_i)$, where $\boldsymbol{y}_i^{\mathrm{tar}}$ is a realized target, and let $\Lambda \geq 0$ be the strength cap. For lower and upper bounds $l \leq u$, define $\mathrm{clip}(t, l, u) = \min\{u, \max\{l, t\}\}$; calibration uses $l = 0, u = \Lambda$. We assign zero strength to a zero-energy direction.

Equation (10) is the standard least-squares solution for scaling a fixed correction. For a fixed joint law P of $(\boldsymbol{f}_c, \boldsymbol{Y}, \boldsymbol{d})$ with finite second moments, its prediction-error change is

$$\begin{aligned}\Delta_{\mathsf{P}}(\lambda) &:= \mathbb{E}_{\mathsf{P}}\|\boldsymbol{Y} - (\boldsymbol{f}_c + \lambda\boldsymbol{d})\|^2 - \mathbb{E}_{\mathsf{P}}\|\boldsymbol{Y} - \boldsymbol{f}_c\|^2 \\ &= -2\lambda\mathbb{E}_{\mathsf{P}}\langle\boldsymbol{R}, \boldsymbol{d}\rangle + \lambda^2\mathbb{E}_{\mathsf{P}}\|\boldsymbol{d}\|^2, \qquad \boldsymbol{R} = \boldsymbol{Y} - \boldsymbol{f}_c. \end{aligned} \tag{19}$$

On the fixed calibration states, the empirical loss is a scalar convex quadratic. Clipping its minimizer to $[0, \Lambda]$ gives the implemented rule. Because zero is feasible, the calibrated strength cannot raise that empirical squared error. Calibration does not learn a better direction or extend automatically to unseen samples, adapted trajectories, or perceptual metrics; no finite-sample generalization bound is established here.

The nonnegative constraint is a one-sided design choice: calibration can attenuate or reject the learned direction but cannot reverse it. In the exact-teacher limit $\boldsymbol{d} = \alpha\boldsymbol{\Delta}$, where $\boldsymbol{\Delta} = \boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}}$ and $\alpha > 0$ is fixed at this noise level,

$$\widetilde{\boldsymbol{f}}_c = \boldsymbol{f}_{\mathrm{ref}} + (1 + \lambda\alpha)\boldsymbol{\Delta}, \qquad \widehat{\lambda} = \mathrm{clip}\left(\frac{\beta^*_{\mathrm{cal}} - 1}{\alpha}, 0, \Lambda\right), \tag{20}$$

where $\beta^*_{\mathrm{cal}} = \sum_i \langle \boldsymbol{y}_i^{\mathrm{tar}} - \boldsymbol{f}_{\mathrm{ref},i}, \boldsymbol{\Delta}_i\rangle / \sum_i \|\boldsymbol{\Delta}_i\|^2$ for nonzero contrast energy on the same calibration states. Substituting $\boldsymbol{Y} - \boldsymbol{f}_c = (\boldsymbol{Y} - \boldsymbol{f}_{\mathrm{ref}}) - \boldsymbol{\Delta}$ proves the identity. If $\beta^*_{\mathrm{cal}} \le 1$, the calibrated dose is zero. Thus this limit amplifies the parent contrast or leaves it unchanged. The teacher scale fitted on a different training collection is not β^*_{cal}, and an imperfect restricted fit can change the contrast's direction. Direct-contrast comparisons must share the same feasible dose range; signed-dose experiments would test an additional design choice.

Fitting a response teacher accurately is a different matter from correcting the parent. In one dimension, for $q \in \{-1, 1\}$, the teacher $t(q) = q$ is fitted exactly by $d = q$, even if the true scalar error is $Y - f_c = -q$. Every positive strength then raises squared error.

B.2 Guidance and changing sampling states

Let $\boldsymbol{f}_{\mathrm{base}}$ denote the CFG baseline prediction at the same state, time, and output coordinates as $\boldsymbol{f}_c$. Its conditioning can be null or negative; a negative prompt does not define a strictly unconditional distribution. It need not equal the reference prediction $\boldsymbol{f}_{\mathrm{ref}}$. Write $\boldsymbol{f}_{\mathrm{CFG}} = \boldsymbol{f}_{\mathrm{base}} + s_{\mathrm{cfg}}(\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{base}})$ for the guided prediction, where s_{cfg} is the scalar classifier-free guidance (CFG) scale. With $\boldsymbol{e}_g = \boldsymbol{Y} - \boldsymbol{f}_{\mathrm{CFG}}$, a conditional prediction correction $\lambda\boldsymbol{d}$ changes the guided prediction by $s_{\mathrm{cfg}}\lambda\boldsymbol{d}$. For a fixed joint law of $(\boldsymbol{f}_c, \boldsymbol{f}_{\mathrm{base}}, \boldsymbol{Y}, \boldsymbol{d})$ with finite second moments, the guided squared-error change is

$$\begin{aligned}\Delta\mathcal{R}_g(\lambda) &:= \mathbb{E}\|\boldsymbol{Y} - (\boldsymbol{f}_{\mathrm{CFG}} + s_{\mathrm{cfg}}\lambda\boldsymbol{d})\|^2 - \mathbb{E}\|\boldsymbol{Y} - \boldsymbol{f}_{\mathrm{CFG}}\|^2 \\ &= -2s_{\mathrm{cfg}}\lambda\mathbb{E}\langle\boldsymbol{e}_g, \boldsymbol{d}\rangle + s^2_{\mathrm{cfg}}\lambda^2\mathbb{E}\|\boldsymbol{d}\|^2. \end{aligned} \tag{21}$$

The predictions in these expectations are held fixed as λ varies; the law may be the empirical distribution of calibration samples. For $s_{\mathrm{cfg}} > 0$, the capped fixed-state minimizer would instead be

$$\widehat{\lambda}_g = \mathrm{clip}\left(\frac{\sum_i \langle \boldsymbol{y}_i^{\mathrm{tar}} - \boldsymbol{f}_{\mathrm{CFG},i}, \boldsymbol{d}_i\rangle}{s_{\mathrm{cfg}}\sum_i \|\boldsymbol{d}_i\|^2}, 0, \Lambda\right). \tag{22}$$

A zero-energy direction again receives zero strength. This is a diagnostic comparator, not the implemented calibration rule. The implementations fit conditional prediction-error moments; LucyEdit additionally weights edited regions and penalizes corrections elsewhere (Appendix C.8). These objectives do not optimize guided error. The relevant alignment is therefore different even on the same states. Adaptive guidance can change the correction further, and subsequent sampling steps and decoding can amplify different directions by different amounts. Equal injection energy therefore leaves room for different generated videos, which is why the direction controls are evaluated after sampling. Scalar calibration alone gives no endpoint-error or perceptual-quality guarantee.

For exact response fitting $\boldsymbol{d} = \alpha(\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}})$, the adapted guided prediction in general output coordinates is

$$\widetilde{\boldsymbol{f}}_{\mathrm{CFG}} = \boldsymbol{f}_{\mathrm{base}} + s_{\mathrm{cfg}}(\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{base}}) + s_{\mathrm{cfg}}\lambda\alpha(\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}}).$$

If $\boldsymbol{f}_{\mathrm{ref}} = \boldsymbol{f}_{\mathrm{base}}$, the effective guidance scale is $s_{\mathrm{cfg}}(1 + \lambda\alpha)$. More generally, at a fixed prediction, $\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}} = \kappa(\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{base}})$ with scalar proportionality factor κ gives the effective scale $s_{\mathrm{cfg}}(1 + \lambda\alpha\kappa)$ even when $\boldsymbol{f}_{\mathrm{ref}} \neq \boldsymbol{f}_{\mathrm{base}}$. A common proportionality factor yields a common scale across states and conditions when the other scales are fixed. Mean-action, component-removed-source, and mean-text references generally supply neither identity. A learned linear head also only approximates the reference contrast. Consequently, a matched direct-contrast comparison must retain the same reference, output units, insertion times, calibration rule, and guidance. The old replacement rows in Appendix D.1 do not supply this comparison.

B.3 Scope of the response constraint

The base branch is linear in its declared coordinate when the state, reference features, and strength are held fixed. Across scenes, reference-dependent routing can be highly nonlinear. The changing state,

nonlinear encoders, and optional energy reallocation each introduce further derivatives. At fixed ξ, suppose the parent admits a single-valued C^2 representation $f_\xi : \mathcal{U} \to \mathbb{R}^{d_{\text{out}}}$ on an open coordinate domain $\mathcal{U} \subseteq \mathbb{R}^{d_q}$. For example, a specified lifting $c = \chi_\xi(\boldsymbol{q})$ defines $f_\xi(\boldsymbol{q}) = F_\theta(\boldsymbol{x}_\tau, \tau, k, \chi_\xi(\boldsymbol{q}))$. Set $\widetilde{f}_\xi(\boldsymbol{q}) = f_\xi(\boldsymbol{q}) + \lambda \boldsymbol{A}\boldsymbol{q}$ with $\boldsymbol{A} = \boldsymbol{A}_{\phi,\tau}$. Differentiation at fixed state, reference, and strength gives

$$D^2_{\boldsymbol{q}} \widetilde{f}_\xi = D^2_{\boldsymbol{q}} f_\xi. \tag{23}$$

These are vector-valued second derivatives. A lossy condition map alone need not define such a parent function; the identity is not a statement about raw text. It also leaves the full trajectory's curvature and regularity undetermined. The fixed-state branch Jacobian is $\lambda_\tau \boldsymbol{A}_{\phi,\tau}$, whose norm is not bounded by its zero curvature. Energy reallocation adds another source of nonlinearity: even with a fixed projection direction, its coefficients depend on the component energies of $\boldsymbol{d}$. The preserved norm belongs to the inserted correction in the declared output space, before guidance and sampling.

C Implementation Details

C.1 Prediction coordinates

Calibration compares a prediction with a target, so both must be expressed in the same coordinates. Diffusion models learn to reverse a noise-adding process (Sohl-Dickstein et al., 2015), but the quantity predicted during that reversal varies. DDPM predicts noise (Ho et al., 2020); noise-conditional score estimation (Song & Ermon, 2019) and the SDE formulation (Song et al., 2021b) describe related quantities with explicit time-dependent scaling. Sampling introduces further choices: DDIM changes the generation process (Song et al., 2021a), improved DDPM learns reverse variances (Nichol & Dhariwal, 2021), and EDM separates preconditioning from solver design (Karras et al., 2022). These choices also sit within different architectures, including autoencoder latents (Rombach et al., 2022) and transformer denoisers (Peebles & Xie, 2023).

Velocity models use another output convention. Flow matching fits conditional vector fields (Lipman et al., 2023), rectified flow regresses straight-path velocities (Liu et al., 2023b), and stochastic interpolants construct transport fields from interpolating processes (Albergo & Vanden-Eijnden, 2023). In particular, the squared-loss calculations here require a declared output space and the appropriate noise-level-dependent conversion between clean prediction, noise prediction, velocity, and an exact score.

Table 6: Prediction units for $\boldsymbol{x}_\tau = a_\tau \boldsymbol{x}_0 + b_\tau \boldsymbol{\epsilon}$. The target column describes the full latent block; the target used with $\boldsymbol{f}_c$ is its selection by $\boldsymbol{S}_{\text{out}}$. Hats denote learned predictions.

Output convention	Full-block target	Clean reconstruction
Noise	$\boldsymbol{\epsilon}$	$(\boldsymbol{x}_\tau - b_\tau \widehat{\boldsymbol{\epsilon}})/a_\tau,\ a_\tau \neq 0$
Clean latent	$\boldsymbol{x}_0$	$\widehat{\boldsymbol{x}}_0$
Linear flow velocity	$\boldsymbol{\epsilon} - \boldsymbol{x}_0$, for $a_\tau = 1 - \tau,\ b_\tau = \tau$	$\boldsymbol{x}_\tau - \tau \widehat{\boldsymbol{v}}^{\text{flow}}$
VP v-prediction	$a_\tau \boldsymbol{\epsilon} - b_\tau \boldsymbol{x}_0$, for $a_\tau^2 + b_\tau^2 = 1$	$a_\tau \boldsymbol{x}_\tau - b_\tau \widehat{\boldsymbol{v}}^{\text{VP}}$

The two velocity conventions differ. For a general differentiable path, the sample derivative is $\dot{a}_\tau \boldsymbol{x}_0 + \dot{b}_\tau \boldsymbol{\epsilon}$. VP v-prediction is the velocity under an angular parameterization of the variance-preserving path (Salimans & Ho, 2022, Appendix D); it is not generally the derivative with respect to τ. The CogVideoX response interface uses VP v-prediction, whereas the Wan flow examples use the linear-path target. These are output conventions, independent of the coordinate constructions in Table 1. At fixed state, both admit a clean-to-output correction of the form $-\delta \boldsymbol{x}_0 / b_\tau$ when $b_\tau > 0$, with $b_\tau = \tau$ for the linear path; their training targets still differ.

Common prediction objectives for the path in (1) have the form

$$\mathcal{L}_{\text{pred}}(\theta) = \mathbb{E}\left[\omega_\tau \| F_\theta(\boldsymbol{x}_\tau, \tau, k, c) - \boldsymbol{Y}_\tau \|^2\right], \qquad \omega_\tau \geq 0, \tag{24}$$

Here ω_τ is a nonnegative time-dependent loss weight, and the expectation is over data, Gaussian noise, and training times. The subscript in $\boldsymbol{Y}_\tau$ makes explicit the time dependence suppressed in the fixed-time notation $\boldsymbol{Y}$. Table 6 gives the full-block targets; $\boldsymbol{Y}_\tau$ is their selection by $\boldsymbol{S}_{\text{out}}$.

For the Gaussian path in Section 3.1, let $p_\tau(\boldsymbol{x} \mid k, c)$ be the conditional noisy-state density. With $b_\tau > 0$ and finite conditional first moments, its exact score satisfies

$$\begin{aligned} s_\tau(\boldsymbol{x}, k, c) := \nabla_{\boldsymbol{x}} \log p_\tau(\boldsymbol{x} \mid k, c) &= -\frac{\mathbb{E}[\boldsymbol{\epsilon} \mid \boldsymbol{x}_\tau = \boldsymbol{x}, k, c]}{b_\tau} \\ &= \frac{a_\tau \mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_\tau = \boldsymbol{x}, k, c] - \boldsymbol{x}}{b_\tau^2}. \end{aligned} \tag{25}$$

Differentiating the Gaussian convolution gives this identity, which underlies denoising score estimation (Song et al., 2021b). Replacing its conditional means with learned predictions gives score estimates; it does not make a pretrained predictor or an arbitrary added residual an exact score. The score differentiates with respect to the noisy state $\boldsymbol{x}$, whereas our response constraint concerns the condition coordinate $\boldsymbol{q}$ at a fixed state. The Gaussian reference distribution over $\boldsymbol{q}$ in Lemma 1 is a separate assumption from Gaussian corruption noise. Equation (25) concerns the full noisy-state density; a selected output block only estimates its corresponding score coordinates.

For AdaWorld, $\boldsymbol{S}_{\mathrm{out}}$ selects the last-frame clean prediction; the correction is zero outside that frame. Bernini converts a clean correction to velocity as $\delta \boldsymbol{v} = -\delta \boldsymbol{x}_0 / \sigma$ before adaptive projected guidance. Here σ is the native linear-path noise coordinate, so $a = 1 - \sigma$ and $b = \sigma$. At a fixed noise level, converting both the correction and its target error multiplies the squared loss by $1/\sigma^2$ and leaves the least-squares strength unchanged. The subsequent nonlinear guidance does not share this equivalence. Wan instead adds a velocity correction before ordinary CFG.

C.2 Coordinate layouts and averaging

The implementations instantiate the two constructions in Table 1. AdaWorld controls, Bernini source components, and Wan T2V text embeddings use condition representations. The CogVideoX, LucyEdit, and Wan I2V heads use native prediction differences. These are implementation choices: text-feature differences are also admissible for a text-conditioned backbone when paired with a compatible reference-dependent operator and readout. Their use does not require reproducing Wan's attention layout. These alternative interfaces have not all been evaluated here.

All vectorizations use a fixed order of the declared axes. Changing between these orders is a fixed permutation absorbed into $\boldsymbol{A}_{\phi,\tau}$; it preserves the Euclidean/Frobenius norm. Spatial packing is also a fixed linear rearrangement. Neither operation makes the coordinate Gaussian.

For AdaWorld, $\boldsymbol{q} \in \mathbb{R}^{32}$ is shared across the output frame. Each latent pixel has a 4×32 slope; stacking the pixel outputs gives the full $d_{\mathrm{out}} \times 32$ operator. Bernini instead uses the edge-coordinate field $\mathsf{Q}_e \in \mathbb{R}^{8 \times (T_z - 1) \times H_z \times W_z}$, where T_z, H_z, W_z are latent frame and spatial sizes. Its theoretical input is $\boldsymbol{q} = \mathrm{vec}(\mathsf{Q}_e)$. Packing each 2×2 spatial patch produces 32 edge channels. Concatenating the incoming and outgoing packed edges, with zero boundary edges, is a further fixed linear map to 64 channels per frame and patch. This concatenation duplicates interior edges and need not preserve the coordinate norm; its scale is part of the operator. The theoretical coordinate remains the vectorized edge field.

For the token interface, let n_t be the number of text tokens and n_v the number of video query tokens. In (8),

$$\begin{gathered} \boldsymbol{U}_c \in \mathbb{R}^{n_t \times d_e}, \quad \boldsymbol{Q}_{\ell h}^{\mathrm{ref}} \in \mathbb{R}^{n_v \times d_{\mathrm{head}}}, \quad \boldsymbol{K}_{\ell h}^{\mathrm{ref}} \in \mathbb{R}^{n_t \times d_{\mathrm{head}}}, \\ \boldsymbol{P}_{\ell h} \in \mathbb{R}^{n_v \times n_t}, \quad \boldsymbol{W}_{V,\ell h} \in \mathbb{R}^{d_e \times d_{\mathrm{head}}}, \quad \boldsymbol{W}_{O,\ell} \in \mathbb{R}^{H_a d_{\mathrm{head}} \times d_o}, \quad \boldsymbol{M}_\ell \in \mathbb{R}^{n_v \times d_o}. \end{gathered}$$

Here $d_e, d_{\mathrm{head}}, d_o$ are embedding, common per-head, and routed-output widths, and H_a is the head count. Each softmax row sums over text keys. With columnwise vectorization, $\boldsymbol{q} = \mathrm{vec}(\boldsymbol{U}_c) \in \mathbb{R}^{n_t d_e}$; the Wan example has $n_t = 512, d_e = 1536$. The parent value bias cancels in the embedding difference, and the linear response path omits the output bias. Each head has its own reference routing matrix. The adapter's subsequent message projections are distinct learned parameters. A native-response interface instead vectorizes the prediction-difference field, whose shape and units follow the parent output block.

Norms and losses. Squared norms sum squared entries; they do not average them. For tensors U, V of the same shape with n entries, define $\mathrm{MSE}(\mathsf{U}, \mathsf{V}) = n^{-1} \|\mathsf{U} - \mathsf{V}\|_{\mathrm{F}}^2$ and $\mathrm{MSE}(\mathsf{U}) = \mathrm{MSE}(\mathsf{U}, 0)$.

The descriptions below specify whether n counts a complete output or sampled rows. Training normalizes each example or cached record before averaging its loss with the other records in an update. Calibration in (10) instead pools the stated inner-product and energy sums. A common averaging factor cancels from that ratio; different per-example factors would change its weights.

C.3 AdaWorld

AdaWorld uses a vector condition. A reference-dependent slope tensor contracts with the standardized action to correct the last-frame clean prediction. The single-route ablations instead neutralize only cross-attention or only vector conditioning in the conditional branch of the reference forward. The other route retains the requested action; the ordinary conditional parent forward and unconditional branch are unchanged. Reference features then depend on $\boldsymbol{q}$, giving $d_\phi(\boldsymbol{q}) = \mathcal{A}_\phi(h^{\mathrm{ref}}(\boldsymbol{q}), \tau)\boldsymbol{q}$ before direction adjustment. These variants use the direction schedule and lie outside the base's fixed-reference condition-linear family.

The 32-dimensional latent action is standardized using training statistics, with standard deviations floored at 0.02. The reference pass replaces both cross-attention and vector conditioning with the training mean, retaining the image history and current noisy state. The parent is frozen, and the head has 149,304 trainable parameters. Reference features are captured immediately before the final output convolution, with 320 channels per latent pixel. Functional layer normalization (epsilon 10^{-5}, with no learned scale or bias) feeds a 128-wide local $\tanh$ map. The spatial mean and log-RMS descriptors feed a 16-wide global $\tanh$ map; an eight-dimensional noise-step embedding completes the 152-dimensional readout input. Each of five zero-initialized readouts produces a 4×32 slope per pixel, bounded by a scaled $\tanh$. The scale is twice the square root of mean reference-contrast energy divided by mean squared coordinate norm at that noise level. Biases are allowed in this reference-only slope path; the final contraction with $\boldsymbol{q}$ still vanishes at $\boldsymbol{q} = \mathbf{0}$.

Writing $\boldsymbol{\Delta} = \boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}}$, the head produces $\boldsymbol{d} = \alpha_\tau \overline{\boldsymbol{A}}_{\phi,\tau}\boldsymbol{q}$. For the sampled spatial rows $\mathcal{I}$, it minimizes $\mathrm{MSE}(\boldsymbol{d}_\mathcal{I}, \alpha_\tau \boldsymbol{\Delta}_\mathcal{I}) / \max(\mathrm{MSE}(\boldsymbol{\Delta}), 10^{-6})$, plus 10^{-6} times the mean squared deployed slope. The numerator averages sampled rows and channels; the denominator uses the complete cached frame. The fixed teacher scale is $\alpha_\tau = \mathrm{clip}(\sum\langle \boldsymbol{Y} - \boldsymbol{f}_{\mathrm{ref}}, \boldsymbol{\Delta}\rangle / \sum\|\boldsymbol{\Delta}\|^2, 0.05, 1.25)$. The deployed operator in (7) is $\boldsymbol{A}_{\phi,\tau} = \alpha_\tau \overline{\boldsymbol{A}}_{\phi,\tau}$. Training uses AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, gradient-norm cap 1, and 128 sampled spatial rows per update. Each of five folds contains 72 training and 18 held-out tasks. Each epoch visits 720 cached task–demonstration–noise-level examples. One optimizer step samples 128 spatial rows from one such example; eight epochs therefore give 5,760 updates per head and 28,800 across all five heads.

After fitting the head, we collect calibration moments along the released model's training paths and select and confirm candidate corrections on training clips using their own five-step latent trajectories. The deployed base adds the calibrated linear correction in (7), with strength given by (10) and $\Lambda = 4$. Because these moments come from training paths and are followed by candidate checks, the procedure does not provide an independent calibration-sample guarantee.

In Table 2, the optional direction adjustment is enabled for Franka and Jaco. The base inserts the calibrated correction at all five sampler steps. The direction variant uses steps 0 and 1. At the allocation step, the unchanged control has dose 4; the motion construction uses amplitude 6 before exact energy matching. Its direction is the difference between the last two parent clean-predicted frames. A 20-frame rollout records 190 batched backbone forwards for the base and 133 for the direction variant, compared with 95 for the released model. Each forward processes both CFG branches, including in the reference pass. The direction variant's count includes 38 reference forwards.

Franka contains 91 eligible public validation episodes. For two short episodes, frames are selected by the predeclared nearest-neighbor temporal rule. Repeated generation uses the same episodes and should be distinguished from independent training repetitions. The Jaco action predictor is trained on 976 episodes and 69,151 transitions, with 2,071 held-out test transitions and $R^2 \simeq 0.399$. This fit quality is measured on held-out recorded transitions. Its calibration need not carry over to generated videos; Appendix F.2 defines the reported distance contrast.

Energy-preserving direction adjustment. Let $\boldsymbol{d}$ be a correction row and $\boldsymbol{m}$ the corresponding parent-motion row, with base dose $b \geq 0$ and requested parallel dose $g \geq b$. Decompose

$$\boldsymbol{p} = \frac{\langle \boldsymbol{d}, \boldsymbol{m} \rangle}{\|\boldsymbol{m}\|^2} \boldsymbol{m}, \qquad \boldsymbol{r} = \boldsymbol{d} - \boldsymbol{p},$$

setting $\boldsymbol{p} = \boldsymbol{0}$ when $\boldsymbol{m} = \boldsymbol{0}$. Write $E = \|\boldsymbol{d}\|^2$, $E_\parallel = \|\boldsymbol{p}\|^2$, $E_\perp = \|\boldsymbol{r}\|^2$, and $c_\times = \langle \boldsymbol{p}, \boldsymbol{r} \rangle$. The cross term is retained to account for numerical nonorthogonality.

For a positive energy budget $b^2 E$ and nonzero component energies, choose

$$a = \min\left\{ g, b\sqrt{E/E_\parallel} \right\},$$
$$s = \frac{-ac_\times + \sqrt{a^2 c_\times^2 + E_\perp (b^2 E - a^2 E_\parallel)}}{E_\perp}, \quad \boldsymbol{\delta} = a\boldsymbol{p} + s\boldsymbol{r}.$$

The cap on a makes the allocation feasible, and the nonnegative root gives $\|\boldsymbol{\delta}\|^2 = a^2 E_\parallel + s^2 E_\perp + 2asc_\times = b^2 E$. At positive budget, a negligible parallel component leaves $b\boldsymbol{d}$ unchanged; a negligible perpendicular component gives $\boldsymbol{\delta} = b\sqrt{E/E_\parallel}\boldsymbol{p}$. Equal doses $g = b$ return $b\boldsymbol{d}$ directly. The implemented direction-changing branch ($g > b$) does not support a zero energy budget. Energy-dependent doses make this optional rule generally nonlinear in $\boldsymbol{q}$, even with a fixed projection direction.

C.4 BERNINI

In Bernini, removing the lifted coordinate component from the source VAE latents forms the reference; the noisy target state is retained. The coupled head contracts linear coordinate features with reference-dependent slopes. This source coordinate has not been identified with the requested edit.

The Bernini study uses 24 training, eight calibration, and four development videos. Its coupled head has 1,168,640 parameters and receives 512 updates at batch size two. A rank-eight PCA basis is fitted to temporal channel-edge statistics of the source. At each edge location (t, h, w), let $\boldsymbol{\Delta}^{\mathrm{src}}_{t,h,w} \in \mathbb{R}^{16}$ be the source's adjacent-frame difference, $\boldsymbol{\mu}_{\mathrm{src}} \in \mathbb{R}^{16}$ the training channel mean, and $\boldsymbol{B} \in \mathbb{R}^{16\times 8}$ the PCA basis. The coordinate is

$$\boldsymbol{q}_{e,t,h,w} = \mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{src}})^{-1} \boldsymbol{B}^\top (\boldsymbol{\Delta}^{\mathrm{src}}_{t,h,w} - \boldsymbol{\mu}_{\mathrm{src}}), \qquad \boldsymbol{\sigma}_{\mathrm{src}} \in \mathbb{R}^8,$$

where $\boldsymbol{\sigma}_{\mathrm{src}}$ contains square-root eigenvalues floored at 0.02. This channelwise operation forms the edge field Q_e; its vectorization is the theoretical coordinate $\boldsymbol{q}$. Lifting $\boldsymbol{B}\,\mathrm{diag}(\boldsymbol{\sigma}_{\mathrm{src}})\boldsymbol{q}_{e,t,h,w}$ channelwise by cumulative summation and subtracting its temporal mean gives a zero-mean path $\mathcal{L}_{\mathrm{src}}(\mathsf{Q}_e)$. The reference source $\mathsf{X}_{\mathrm{src}} - \mathcal{L}_{\mathrm{src}}(\mathsf{Q}_e)$ retains the temporal mean and complementary edge components; $\mathcal{L}_{\mathrm{src}}$ is the fixed linear lifting map on the edge tensor and $\mathsf{X}_{\mathrm{src}}$ the source latent tensor. Both passes retain the edit text and the same noisy target. Native 2×2 spatial packing maps eight coordinate channels to 32; incoming and outgoing edges are concatenated with zero boundary edges, giving 64 input channels. This coordinate describes source motion; it does not have the same meaning as an action latent or an edit instruction.

Terminal reference features have width 1,536 and use functional layer normalization without learned scale or bias before a width-128 state projection. SiLU activations and a $3 \times 3 \times 3$ convolution then generate a step-specific 64×16 slope field. At frame t and packed spatial position p, the incoming/outgoing coordinate is $\boldsymbol{u}_{t,p} \in \mathbb{R}^{64}$. A learned bias-free projection $\boldsymbol{V}_\phi \in \mathbb{R}^{16\times 64}$ and slope $\boldsymbol{G}_{\phi,t,p} \in \mathbb{R}^{64\times 16}$ give $\boldsymbol{d}_{t,p} = \boldsymbol{G}_{\phi,t,p}\boldsymbol{V}_\phi \boldsymbol{u}_{t,p} \in \mathbb{R}^{64}$. Here 16 is the head rank, and $\boldsymbol{G}_{\phi,t,p}$ depends only on reference features. The slope readout is zero-initialized and uses $\tanh$. The frame variant omits the convolution; the integrated variant additionally averages adjacent output fields and integrates them with zero temporal mean. Thus these ablations change the reference processing or linear output integration, not the source coordinate.

The response loss fits the scaled difference between parent and reference clean estimates. It is normalized by the scaled teacher's mean squared magnitude and adds 10^{-4} times the mean squared correction. The prediction-error control instead fits clean-target error on cached parent trajectories with analogous normalization. The FM control in the ablation tables uses these cached trajectories;

it does not use independently noised flow pairs. The teacher scale is fitted against $\boldsymbol{Y} - \boldsymbol{f}_{\mathrm{ref}}$ and clipped to $[0, 4]$. Each loss averages all time, packed-space, and output-channel entries in one cached record and divides by its own target MSE with floor 10^{-8}; the two record losses are then averaged for an update. Hybrid variants add 0.25 times the clean-error loss; the temporal hybrid adds a further 0.25 times the normalized adjacent-frame error loss. AdamW uses learning rate 10^{-4}, weight decay 0.01, and gradient-norm cap 1. Four cached steps per training video give 96 records. Doses are fitted against clean-target error on the eight calibration videos with $\Lambda = 4$.

At sampler steps $\{0, 1, 8, 20\}$, the calibrated doses are approximately $\{0, 0, 0.270, 4\}$. The direction-allocation step consequently receives zero correction. Changing its direction leaves the video unchanged, explaining the identical scores of the corresponding variants.

C.5 Text-to-video generation

Wan uses (8): E_θ is the parent's frozen text-embedding transformation, $\boldsymbol{U}_c = E_\theta(c) - E_\theta(c_{\mathrm{ref}}) \in \mathbb{R}^{512\times1536}$, and $\boldsymbol{q} = \mathrm{vec}(\boldsymbol{U}_c)$. Value biases cancel. Reference features determine the gates and temporal attention; message projections and the zero-initialized readout are linear. Thus the velocity correction remains linear in $\boldsymbol{q}$ at fixed state and reference. It excludes condition-dependent keys and is not the parent's complete text Jacobian.

Wan uses 512 projected text tokens of width 1,536. We form the reference by averaging raw training text features before the embedding transformation. Averaging the projected embeddings would produce a different reference when that transformation is nonlinear. At blocks 15 and 29, reference queries and keys route a linear value difference. The parent value and output projections remain frozen; the two routed messages then pass through learned bias-free $1536 \to 256$ projections. Layer-normalized terminal reference features pass through a $1536 \to 256$ state projection, a step embedding, and $\tanh$. Reference-only temporal queries and keys have width 64 and use sinusoidal positions; their softmax weights mix linear message values of width 256. Four bias-free, zero-initialized $256 \to 64$ readouts produce packed velocity corrections. No condition-dependent attention weights enter this path. All 512 projected tokens are retained. Other attention architectures require a compatible routing construction that preserves reference-only weights and a linear condition path. The head has 1,344,768 trainable parameters. Training for 512 updates at batch size four entails 2,048 example exposures and 4,096 parent/reference evaluations.

The data comprise 128 training examples, 16 calibration examples divided equally between dose fitting and confirmation, and eight reused development prompts. Generation uses 40 steps of the UniPC scheduler (Zhao et al., 2023), flow shift 8, CFG scale 6, and 33 frames at 480×832 resolution and 16 fps. Inserting corrections at steps 6, 12, 20, and 28 increases parent/reference calls from 80 to 84.

Here $\boldsymbol{f}_c$ and $\boldsymbol{f}_{\mathrm{ref}}$ are predictions in flow units. FM training fits $\boldsymbol{\epsilon} - \boldsymbol{x}_0 - \boldsymbol{f}_c$ by mean squared error; response training fits $\boldsymbol{\Delta} = \boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}}$ with response-energy normalization. Losses average all output entries within each example, then average the four example losses for an update. Both add 0.01 times the mean squared correction and a 0.01-weighted squared hinge encouraging a measured retention ratio of at least 0.5. The retention statistic is $\rho_{\mathrm{ret}} = 1 + \langle \boldsymbol{d}, \boldsymbol{\Delta}\rangle / \max(\|\boldsymbol{\Delta}\|^2, 10^{-6}n)$, where n is the number of entries in that example; the penalty is $[\max(0, 0.5 - \rho_{\mathrm{ret}})]^2$. It measures a scalar projection and imposes no lower bound on singular values. Training uses AdamW with learning rate 10^{-4}, weight decay 0.01, and gradient-norm cap 1. Each example draws fresh flow noise; the second half of calibration confirms the already-fitted strengths without refitting them. Dose fitting uses conditional FM moments with $\Lambda = 1$, so its objective remains conditional prediction error before guided rollout.

C.6 Computational cost

Table 7 distinguishes trainable size from sampling cost. Wan2.1 T2V adds four reference evaluations per rollout; LucyEdit reuses a native branch, while base AdaWorld evaluates a reference at every step. In one Franka timing comparison, adaptation takes approximately 55 seconds per episode versus 42 seconds for the parent. For Wan2.1 T2V-1.3B, elapsed time per video increases from 105.45 seconds for the parent to 110.22 seconds with SRA, an additional 4.77 seconds (approximately 4.5%).

These counts exclude head computation and do not equal wall-clock overhead. The Wan2.2 parameter count includes both experts, only one of which is active per step. AdaWorld's five-head collection

Table 7: Parameters in the head active for one video, and backbone forward calls per video. AdaWorld uses five fitted heads under fixed routing. AdaWorld and CogVideoX count batched CFG forwards; the other interfaces count individual prediction branches.

Interface / variant	Active parameters	Native calls	Adapted calls
AdaWorld / base	149,304	95	190
AdaWorld / direction	149,304	95	133
Bernini / coupled	1,168,640	—	—
Wan2.1 T2V / token	1,344,768	80	84
CogVideoX / response	673,024	50	54
LucyEdit / response	50,179	100	100
Wan2.2 I2V / response	140,928	80	120

contains 746,520 trainable parameters and uses 28,800 fitting updates in total; each clip activates one head and shares the same frozen parent. Matched learned-head comparisons must preserve this routing and total budget. Bernini's total model-call count is unavailable.

C.7 IMAGE-CONDITIONED GENERATION

Wan2.2-I2V-A14B uses the native-prediction coordinate in Table 1. At a fixed sampling state, the conditional and reference calls retain the same input-image channels; only the text context changes to the native negative prompt. Let $\mathsf{U}_\tau \in \mathbb{R}^{16\times T_z\times H_z\times W_z}$ denote their velocity difference, where T_z, H_z, W_z are latent temporal and spatial sizes. The head receives its temporal difference from the first latent slice,

$$(\mathsf{Q}_\tau)_{:,t,:,:} = (\mathsf{U}_\tau)_{:,t,:,:} - (\mathsf{U}_\tau)_{:,1,:,:}, \qquad 1 \le t \le T_z, \qquad \boldsymbol{q} = \mathrm{vec}(\mathsf{Q}_\tau).$$

This subtraction is a fixed linear map of the native response. The added velocity is set to zero at the first latent slice by a fixed mask; the decoded first frame is not replaced. Both operations preserve linearity in the chosen response coordinate at a fixed state and reference.

Two slope generators follow the parent's high- and low-noise experts. Each combines the 16-channel sampling state, 16-channel reference prediction, and 20 retained image-conditioning channels, normalized separately across channels. A width-32 spatial-temporal convolution and a four-component time embedding produce a state-dependent field of three 16×16 matrices per location. Bounded matrix entries contract the previous, current, and next temporal slices of Q_τ, with replicated endpoints. No nonlinear operation in the slope generator receives Q_τ. The two-expert head has 140,928 trainable parameters in total; only one expert is active at each step.

Each of the eight fits uses 128 updates with four examples per update, drawn from 128 fitting videos. Adam uses learning rate 2×10^{-4}. The loss combines conditional flow-matching error, a 0.01-weighted correction-energy penalty, and a 0.01-weighted hinge penalty for response retention below 0.5. Sixteen calibration videos are evaluated at four noise positions: the first eight fit one clipped nonnegative strength per expert, and the remaining eight check the fixed strengths. The strengths are fixed before the eight-source development comparison.

Generation uses UniPC with 40 steps, shift 3, CFG scale 3.5, and 33 frames at 480×832. The first source frame and its text form the input. All methods share inputs and generation noise. The complete deployment applies the learned residual at every step and leaves the native negative branch unchanged, using 120 model evaluations per video versus 80 for the parent. The random comparator has the same first-slice support and is normalized to the learned correction's energy at its own current state. Its trajectory evolves separately from that of the learned residual.

C.8 LUCYEDIT: SOURCE-CONDITIONED RESPONSE COORDINATES

The backbone is Lucy-Edit-1.1-Dev (DecartAI Team, 2025), frozen throughout adaptation. The reference retains the source video and current noisy state while replacing the editing instruction with empty text. The condition coordinate is the difference between the conditional and empty-text velocity predictions, scaled by a fixed per-channel RMS estimated on the fitting set, with a floor of 0.02.

At each latent location, the noisy latent, source latent, and reference velocity provide 144 reference channels. Layer normalization, a learned projection, a step embedding, and $\tanh$ give a 128-dimensional state. Reference-only softmax weights mix the previous, current, and next latent-time responses after a bias-free coordinate projection. A reference-only multiplicative gate and one of four zero-initialized, bias-free readouts produce the velocity correction. Thus the correction remains linear in the chosen response coordinate when the noisy state, source, and reference are fixed. The head contains 50,179 trainable parameters.

Adaptation uses 128 local-modification pairs from OpenVE-3M, with 16 separate calibration pairs and 16 development pairs. The first eight development pairs are used for training-objective and fitted-head selection; the remaining eight are not used for model selection or the reported evaluations. The first eight calibration pairs fit the four correction strengths and the remaining eight check the fixed strengths. Fitting uses 81 frames at 480×832, AdamW with learning rate 10^{-4}, no weight decay, gradient-norm cap 1, effective batch size two, and 128 updates. The flow target is noise minus target latent. A fitting-set estimate of target–source latent differences weights the prediction loss toward changed regions: each channel-averaged squared difference is divided by itself plus 0.24239. The objective adds a correction-energy penalty on the complementary region and a global correction-energy penalty with weight 0.01. These weights are used only during fitting and calibration; inference receives neither a target video nor a target-derived mask.

Calibration retains the same regional weights and preservation penalties. For sample i with n_i output entries, let w_{ij} be the source–target weight defined above, broadcast to entry j. Define $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{w_i} = (n_i^{-1} \sum_j w_{ij} u_j v_j) / \max(n_i^{-1} \sum_j w_{ij}, 10^{-12})$ and $\|\boldsymbol{u}\|_{w_i}^2 = \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{w_i}$. With $\boldsymbol{e}_i = \boldsymbol{y}_i^{\mathrm{tar}} - \boldsymbol{f}_{c,i}$, the region-weighted objective gives

$$\widehat{\lambda} = \mathrm{clip}\left(\frac{\sum_i \langle \boldsymbol{e}_i, \boldsymbol{d}_i \rangle_{w_i}}{\sum_i [\|\boldsymbol{d}_i\|_{w_i}^2 + \|\boldsymbol{d}_i\|_{1-w_i}^2 + 0.01 \|\boldsymbol{d}_i\|^2 / n_i]}, 0, 1 \right).$$

The denominator includes correction energy outside the edited region and the global penalty. The fixed weights calibrate conditional prediction error, not guided prediction error.

Generation retains the native 50-step UniPC sampler and guidance scale 5. Corrections enter at zero-based steps 8, 16, 25, and 35, with calibrated strengths 0.46959, 1, 0.45302, and 1. The native guided velocity is retained, and the scaled correction is added in the same guidance units. Both methods use 100 backbone evaluations per video because the empty-text prediction is already part of native guidance. The head is the lower-development-FVD fit among two fits of this objective, after comparing training objectives on these eight development inputs. Selection precedes the 431-input benchmark evaluation. The complete benchmark uses each source's original duration and frame rate, resized to 480×832; temporary padding for the model's temporal layout is removed on export. Exact and temporal-fingerprint checks found no overlap between the benchmark inputs and the 160 adaptation pairs.

C.9 CogVideoX: reference-conditioned response coordinates

CogVideoX-2B (Yang et al., 2025) uses the native prediction-difference interface in Table 1, with the backbone frozen throughout adaptation. At the current noisy state and timestep, the reference replaces the prompt with empty text. A separate batched CFG call places this reference in the same batch position as the conditional prediction. Their difference is expressed in the parent's native VP v-prediction units and packed into 64 response channels per latent patch. The native negative CFG branch is retained.

Layer normalization of the 1,920-dimensional reference features, a learned projection, a four-position step embedding, and $\tanh$ produce a width-256 state. Reference-only queries and keys of width 64 determine softmax weights over a 3×3 spatial neighborhood within each latent frame; positions outside the grid are excluded. The head equally mixes the original response with this spatially routed response, applies a bias-free $64 \to 256$ projection and a reference-dependent multiplicative gate, and adds temporal attention over the latent frames at each spatial patch. Temporal attention uses reference-only queries and keys with sinusoidal time positions and a linear value projection. Four zero-initialized, bias-free $256 \to 64$ readouts produce the correction at the four insertion positions. All routing weights and gates depend on the reference features, so the correction remains linear in the response coordinate at a fixed state and reference. The head contains 673,024 trainable parameters.

All eight fits in Figure 3(c) use this configuration. Each uses 128 video–text fitting examples and 512 updates with effective batch size four, accumulated from four individual examples. AdamW uses learning rate 10^{-4}, weight decay 0.01, and gradient-norm cap 1. Fresh Gaussian noise constructs VP training states at the four insertion positions. The loss fits the native conditional v-prediction error and adds a 0.01-weighted mean squared correction and a 0.01-weighted squared hinge for response retention below 0.5. Retention is measured by the projection of the correction onto the native conditional–negative prediction contrast, using the normalization in Appendix C.5. Of 16 calibration examples, the first eight fit one strength per insertion in $[0, 1]$ against conditional prediction error, and the remaining eight check the fixed strengths. The eight development inputs used for configuration choice are also used for the reported evaluation.

Generation uses the native DDIM sampler (Song et al., 2021a) with 50 steps, CFG scale 6, and 33 frames at 480×720 and 16 fps. Corrections enter at zero-based steps 8, 15, 25, and 35 in native VP v-prediction units before CFG. Parent and SRA share prompts, generation noise, and sampler settings. The parent makes 50 batched CFG forwards per video; four additional reference forwards bring SRA to 54. Each batch evaluates two prediction branches, giving 100 and 108 individual branch evaluations, respectively.

D Extended Results

D.1 Action-conditioned prediction and ablations

Table 8 extends the LIBERO comparison in Table 2 with the earlier linear-replacement results. The AdaWorld and SRA rows use the same updated means as the main table. The Franka and Jaco prediction rows also follow Table 2, and the LIBERO direction comparison follows Table 4. Separate component studies and independent refits retain their own evaluation settings. ECS is embedding cosine similarity, and Motion is absolute-log motion-ratio error. The compute control makes the additional reference evaluation and discards its output. It checks the effect of that extra computation, not the best use of an equal inference budget. In particular, both old replacement rows evaluate $\boldsymbol{f}_{\mathrm{ref}} + \gamma B_{\mathrm{old}}(h^{\mathrm{ref}})\boldsymbol{q}$; neither adds $\gamma(\boldsymbol{f}_c - \boldsymbol{f}_{\mathrm{ref}})$ to $\boldsymbol{f}_c$. Here B_{old} is the earlier fitted replacement operator, h^{ref} its reference features, and γ its scalar strength. These rows therefore cannot establish an advantage over direct reference guidance, and their earlier fitting and calibration differ from SRA's.

Table 8: LIBERO prediction with residual adaptation and earlier linear replacements. AdaWorld and SRA match Table 2.

Method	LPIPS↓	FVD↓	PSNR↑	SSIM↑	Motion error↓
AdaWorld	0.28174	640.4	**21.754**	0.7981	0.1867
SRA	**0.28020**	**612.1**	21.753	**0.7987**	**0.1858**
Earlier replacement results					
Linear replacement ($\gamma = .2$)	0.31890	848.5	19.577	0.7598	0.3487
Linear replacement ($\gamma = 1$)	0.37770	836.7	18.734	0.7276	0.3980

Base residual without direction adjustment. The base correction is evaluated on 1,500 clips from 30 held-out LIBERO tasks without direction adjustment. Tables 2 and 8 give FVD $640.4 \to 612.1$, a 4.42% reduction, and LPIPS $0.28174 \to 0.28020$. SSIM changes from 0.7981 to 0.7987, motion error from 0.1867 to 0.1858, and PSNR from 21.754 to 21.753. The paired LPIPS analysis in Figure 3(a) improves on 28 of 30 tasks, with task-bootstrap 95% interval $[0.001021, 0.002081]$. That interval describes the figure's paired LPIPS records, rather than uncertainty in the aggregate FVD. The base uses 190 model calls per clip versus 95 for the parent.

The Jaco prediction comparison lowers FVD from 537.1 to 503.7 (6.22%) and LPIPS from 0.29347 to 0.29336. The earlier compute-only control changes LPIPS from 0.29347 to 0.29352, a non-negligible difference relative to the prediction gain. Its cause has not been localized. In the original paired direction study, the LPIPS interval relative to the unchanged correction includes zero, and random reallocation has lower FVD than that study's motion-based correction. Those controls and their uncertainty concern the earlier paired study, rather than a significance test of the updated prediction rows.

Table 9: Jaco prediction on 109 episodes and earlier control measurements. AdaWorld and SRA match Table 2.

Method	LPIPS↓	FVD↓	PSNR↑	SSIM↑	Motion error↓
AdaWorld	0.29347	537.1	22.687	0.7536	0.2456
SRA	**0.29336**	**503.7**	**22.689**	**0.7541**	**0.2421**
Earlier control measurements					
Compute control	0.29352	537.3	22.687	0.7536	0.2456
Unchanged energy	0.29340	534.0	22.685	0.7531	0.2458
Random direction	0.29339	533.4	22.685	0.7531	0.2459

Recipient-trajectory similarity diagnostic. The donor supplies the action and the recipient supplies the initial state, but the reference video remains the recipient's original future. The paired scores in Table 10 reward resemblance to that future, including a hypothetical perfect reconstruction that ignores the donor. FVD compares against the same recipient-future population and does not test pairwise donor response. The scores average three paired generation repetitions on the same 1,500 inputs. The LPIPS reduction relative to the parent is 1.184×10^{-3}, and FVD decreases from 653.897 to 615.130 (5.93%). In Table 4, the same SRA result lowers LPIPS relative to the unchanged and random directions by 0.442×10^{-3} and 0.443×10^{-3}. These are differences between the reported means; no interval from an earlier evaluation is assigned to them. Donor-action correctness requires a valid counterfactual target or a separately validated, phase-aligned action-response measure; additional repetitions of these scores cannot supply it.

Table 10: LIBERO recipient-trajectory similarity. Scores match Table 4; references are the recipient's recorded futures.

Method	LPIPS↓	FVD↓
AdaWorld	0.315660	653.897
SRA	**0.314476**	**615.130**

Table 11: Full metrics for the same reference-route and operator-rank study as Table 22, on 90 training tasks with three source positions per task. SRA includes direction adjustment; single-route references retain the action in the other route.

Method	LPIPS↓	FVD↓	ECS↑	PSNR↑	SSIM↑	Motion err.↓
w/o direction adjustment	0.30677	753.9	0.7897	21.546	0.7801	0.1986
SRA	0.30670	751.3	0.7904	21.543	0.7801	0.1985
Rank-16 projection	0.30671	751.4	0.7904	21.543	0.7801	0.1984
Rank-24 projection	0.30669	751.7	0.7904	21.543	0.7801	0.1984
Rank-8 projection	0.30668	751.1	0.7904	21.544	0.7801	0.1980
Neutralize cross-attention only	0.30760	761.3	0.7883	21.559	0.7798	0.1984
Neutralize global vector only	0.30753	759.1	0.7887	21.558	0.7799	0.1985
AdaWorld (matched calls)	0.30762	764.9	0.7887	21.528	0.7800	0.2060

Table 12: Franka prediction on 91 validation episodes. Scores match Table 2.

Method	LPIPS↓	FVD↓	PSNR↑	SSIM↑	Motion error↓
AdaWorld	0.30313	1357.1	19.525	0.7420	0.3446
SRA	**0.30219**	**1312.2**	**19.526**	**0.7424**	**0.3429**

The Franka comparison lowers FVD by 3.31% and LPIPS by 0.00094.

To assess variation across fitted heads, we independently retrain three five-head collections using the training-task partition and fitting procedure in Appendix C.3. Each collection uses the direction adjustment and is evaluated on the same 91 Franka episodes with a common generation seed and

fixed episode-index routing. The three collections improve LPIPS by 0.819, 1.177, and 0.012 in 10^{-3} units, with 95% intervals $[0.168, 1.465]$, $[0.466, 1.872]$, and $[-1.013, 0.931]$, respectively. The corresponding absolute FVD differences, computed as SRA minus AdaWorld, are $+0.055$, $+1.191$, and -0.968. Thus the first two LPIPS gains accompany slightly worse FVD, while the third fit offers no clear LPIPS advantage.

The earlier attempts to replace the pretrained prediction help explain why we retain it. On a separate 500-clip validation set, an affine replacement gives LPIPS/FVD of $0.35416/709.2$, against AdaWorld's $0.28470/584.1$, and improves none of the ten tasks. In a two-case test, a complete-affine projection reduces measured curvature from 0.0443 to approximately 7.4×10^{-12} but worsens condition utility. Lower curvature alone was therefore insufficient in these tests. Since fitting and calibration also differ between the replacement and residual methods, the full performance difference cannot be attributed to residualization alone.

D.2 Text-guided video editing: paired quality and architecture ablations

These comparisons vary the residual architecture and training objective on Bernini. They are separate from the LucyEdit OpenVE-Bench evaluation in Table 3, which uses a different backbone, dataset, and evaluation criterion.

The coupled response-trained adapter leaves PSNR and SSIM nearly unchanged while raising frame-difference MSE from 0.003795 to 0.003808. All metrics in each row use the same generated videos for one fitted head. LPIPS has not been measured on this editing set. Direction variants with zero calibrated correction at the differing step generate identical videos.

The apparent FVD gain is sensitive to the generated samples. With the same head, one repetition lowers FVD from 1925.9 to 1904.2, while another raises it from 1430.4 to 1434.4. Omitting a single video can reverse the former improvement. Four videos with 400-dimensional features give an empirical covariance of rank at most three, making this sensitivity especially relevant.

The controls expose a further ambiguity in what FVD rewards. Copying the source makes no edit, yet obtains FVD 1243.9, below the parent in one comparison. The integrated FM control with random allocation also scores lower (1030.9) than the response-trained adapter. These observations call for an instruction-compliance measure alongside distributional quality before concluding that an edit has improved.

Table 13: Bernini text-guided video editing on four development videos. PSNR and SSIM compare against the target edit; frame-difference MSE compares adjacent-frame changes with those in the target video.

Method	LPIPS↓	PSNR↑	SSIM↑	Frame-diff. MSE↓	FVD↓
Bernini		15.372	.5327	**.003795**	1114.6
+ SRA		15.372	**.5328**	.003808	**1088.3**

Table 14: Bernini architecture and training-objective ablations. Entries report FVD; the parent model obtains 1114.6.

Head / objective	Base FVD↓	Motion FVD↓	Random FVD↓
Coupled / Teacher	1088.3	1088.3	1088.3
Coupled / Teacher + FM	1149.1	1149.1	1149.1
Coupled / Teacher + temporal	1111.6	1111.6	1111.6
Coupled / FM control	1171.2	1148.7	1136.0
Frame / Teacher	1107.2	1107.2	1107.2
Frame / Teacher + FM	1130.4	1130.4	1130.4
Frame / Teacher + temporal	1102.2	1102.2	1102.2
Frame / FM control	1263.2	1321.6	1289.2
Integrated / Teacher	1155.5	1155.5	1155.5
Integrated / Teacher + FM	1140.5	1140.5	1140.5
Integrated / Teacher + temporal	1134.4	1134.4	1134.4
Integrated / FM control	1112.4	1104.7	1030.9

Table 15: Bernini reconstruction quality under different architectures and training objectives.

Head / objective	PSNR↑	SSIM↑	Frame-diff. MSE↓
Coupled / Teacher	15.372	0.5328	0.003808
Coupled / Teacher + FM	15.371	0.5328	0.003807
Coupled / Teacher + temporal	15.372	0.5329	0.003804
Coupled / FM control	15.349	0.5338	0.003459
Frame / Teacher	15.372	0.5327	0.003816
Frame / Teacher + FM	15.373	0.5327	0.003814
Frame / Teacher + temporal	15.370	0.5329	0.003820
Frame / FM control	15.347	0.5311	0.003608
Integrated / Teacher	15.373	0.5328	0.003794
Integrated / Teacher + FM	15.372	0.5328	0.003794
Integrated / Teacher + temporal	15.372	0.5327	0.003795
Integrated / FM control	15.370	0.5328	0.004061

D.3 TEXT-TO-VIDEO GENERATION: OBJECTIVE AND DIRECTION COMPARISONS

This study uses Wan2.1 T2V-1.3B. Its objective and direction comparisons are distinct from the CogVideoX-2B and Wan2.2-I2V-A14B results in Figure 3(c,d). Table 16 summarizes all eight response-trained fits and their direction controls on the same eight development prompts. SRA attains mean FVD 3456.6 ± 2.6, compared with the parent's 3630.4, a mean reduction of 4.79%. Random and motion-based directions give 3483.8 ± 14.6 and 3535.7 ± 6.3, respectively. Each SRA fit has lower FVD than its matched random and motion controls. The spread describes variation across fits on these prompts.

Table 16: Wan text-to-video FVD across eight fits on eight shared development prompts. Entries are mean $\pm$ sample standard deviation across fits.

Method	FVD↓
Wan	3630.4
SRA	3456.6 ± 2.6
Random direction	3483.8 ± 14.6
Motion direction	3535.7 ± 6.3

The single-fit comparisons use the first fit of each objective, which has the lower development FVD of that objective's first two fits. Among these four fits, the first response-trained fit has the lowest development FVD, 3455.4, close to the eight-fit mean of 3456.6 ± 2.6 in Table 16. The six VBench dimensions (Huang et al., 2024) in Table 17 refer to this fit, not an eight-fit quality average: subject consistency, motion smoothness, and aesthetic quality improve; background consistency and dynamic degree are unchanged, while imaging quality decreases slightly. All metrics in each single-fit row use the same generated videos. The objective comparisons and separate replication results below retain their respective evaluation settings.

Table 17: Wan text-to-video generation for the first response-trained fit on eight development prompts. The six quality measures follow VBench; Smooth. denotes motion smoothness, while dynamic degree measures the amount of motion.

Method	FVD↓	Subj.↑	Backgr.↑	Smooth.↑	Dynamic	Aesth.↑	Imaging↑
Wan	3630.4	.9735	**.9847**	.9839	.2500	.6334	**.6682**
+ SRA	**3455.4**	**.9740**	**.9847**	**.9841**	.2500	**.6341**	.6675

The objective comparison uses the same first fits; the direction controls reuse the first response-trained fit.

Table 18: Wan training-objective and correction-direction comparisons for the first fit of each objective. Dynamic degree is .2500 for all methods.

Method	FVD↓	Subj.↑	Backgr.↑	Smooth.↑	Aesth.↑	Imaging↑
Wan	3630.4	.9735	.9847	.9839	.6334	.6682
SRA (FM)	3538.1	.9787	.9807	.9846	.6430	.6707
SRA (response)	3455.4	.9740	.9847	.9841	.6341	.6675
Motion direction	3542.3	.9755	.9807	.9840	.6378	.6740
Random direction	3469.7	.9748	.9835	.9840	.6359	.6747

A second fit of each objective gives FVD 3626.8 for FM and 3457.7 for response training, with imaging quality $.6688$ and $.6731$, respectively. The FVD gain itself does not survive a separate replication: the adapted models range from 3340.3 to 3588.4, above the parent's 3302.2. In one of these comparisons, the FM adapter also lowers imaging quality from $.6748$ to $.6644$ and aesthetic quality from $.6136$ to $.6104$.

Training longer can reverse an initial gain. An earlier Wan residual improves FVD after 128 updates (3288.7 and 3350.5, versus 3360.5 for the parent), then regresses after 256 updates (3374.2 and 3390.0).

D.4 Ablation summary

The unchanged, random, and motion-direction controls share insertion steps 0 and 1 and correction energy. They isolate direction within that schedule, rather than compare directly with the five-step base. On Franka, the direction variant improves LPIPS over unchanged and random directions by 0.146×10^{-3} and 0.208×10^{-3}, with positive paired intervals. The corresponding recipient-similarity diagnostics are in Appendix D.1.

Neutralizing only one action route in the reference worsens the development scores. The unneutralized route makes the reference features action-dependent, so this ablation also leaves the base's condition-linear family. Truncating the fitted operator to rank eight retains about 99% of its energy with little change in quality, although the stored head still has the same number of parameters. One AdaWorld refit has an inconclusive LPIPS gain, and both Bernini and Wan include regressions. The unsuccessful attempts to replace the full pretrained prediction are also reported above.

D.5 Additional generation results

Figure 4 expands Figure 3(d) using all eight fitted heads on the same eight development inputs sampled from OpenVid-1M. Each fit improves full-video FVD, future-frame FVD, subject consistency, and aesthetic quality over both the parent and its random residual. Background consistency, motion smoothness, and imaging quality improve on average, with individual fits on both sides of the parent. The variation across fits describes repeated adaptation on these inputs; it does not measure variation across a larger input population. The image-conditioned construction used in Figure 4 is specified separately in Appendix C.7.

For CogVideoX-2B, the eight-fit summary in Table 19 corresponds to Figure 3(c). The five quality dimensions improve in mean, while dynamic degree remains 0.5. These text-conditioned scores use a different model from the image-conditioned comparison above. Appendix D.8 specifies their aggregation.

Table 19: CogVideoX-2B on an OpenVid-1M subset. SRA scores are mean $\pm$ sample standard deviation across eight fits, matching Figure 3(c).

Metric	Parent	SRA
Subject consistency↑	0.98290	0.98617 ± 0.00097
Background consistency↑	0.97672	0.98065 ± 0.00184
Motion smoothness↑	0.98658	0.98682 ± 0.00012
Aesthetic quality↑	0.59410	0.60126 ± 0.00235
Imaging quality↑	0.58266	0.58415 ± 0.00118
Dynamic degree	0.50000	0.50000 ± 0.00000

Three fits sampled uniformly without replacement from the same eight CogVideoX fits illustrate individual variation. All eight fits contribute to the aggregate scores and Figure 3(c).

Table 20: CogVideoX-2B scores for three sampled fits, with original fit indices.

Metric	Parent	SRA fit 2	SRA fit 3	SRA fit 5
Subject consistency↑	0.98290	0.98669	0.98560	0.98743
Background consistency↑	0.97672	0.97803	0.98265	0.97925
Motion smoothness↑	0.98658	0.98679	0.98699	0.98671
Aesthetic quality↑	0.59410	0.60021	0.60169	0.59837
Imaging quality↑	0.58266	0.58262	0.58347	0.58398
Dynamic degree	0.50000	0.50000	0.50000	0.50000

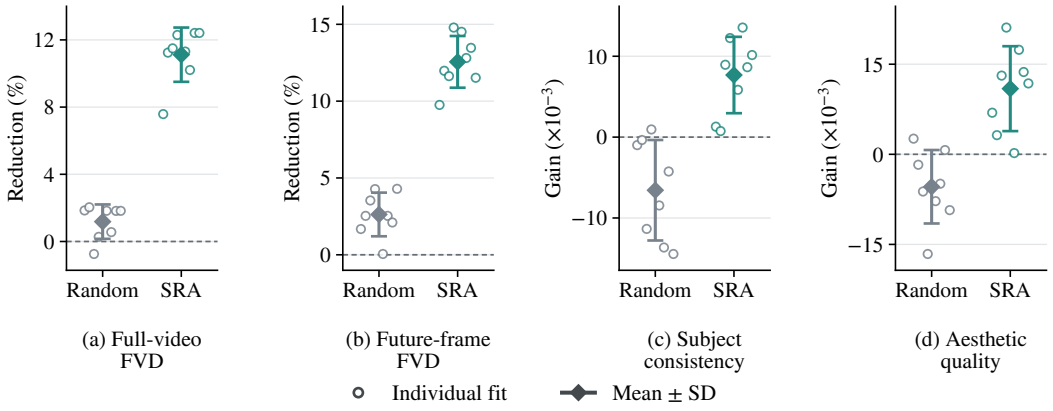


Figure 4: Wan2.2-I2V-A14B gains across eight fits on eight shared OpenVid-1M inputs. Circles show fits; diamonds and bars show mean and sample SD. FVD uses percentage reductions; quality uses absolute differences from the parent. Future excludes the first frame.

D.6 Qualitative action-conditioned video prediction

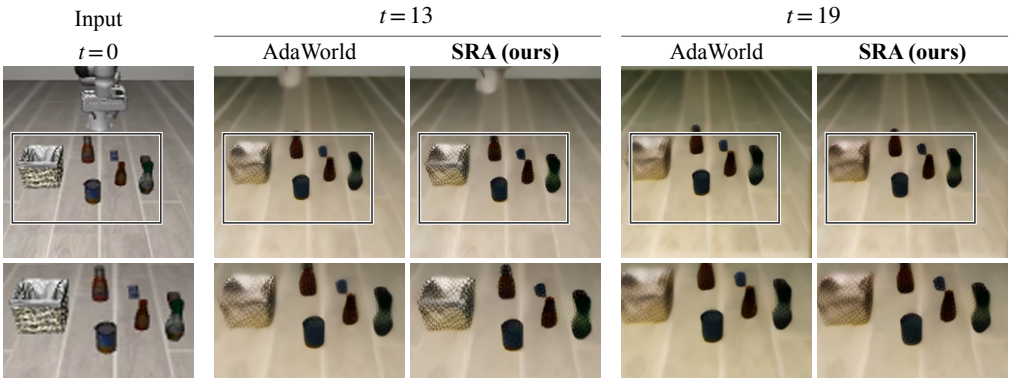


Figure 5: An example of action-conditioned video prediction on LIBERO, on the task of placing a BBQ-sauce bottle in a basket. Columns show the shared input at $t = 0$, followed by AdaWorld and SRA at $t = 13$ and $t = 19$, relative to the sampled clip start. Complete frames appear above, with the boxed regions enlarged below using identical crop coordinates. In the enlarged views, SRA shows more distinct basket texture and object boundaries than AdaWorld, particularly at $t = 13$. Both methods use the same initial observation, action-source demonstration, and generation noise.

Figure 5 compares AdaWorld and SRA on a LIBERO rollout with an initial observation from one demonstration and latent actions from another demonstration of the same task. The enlarged views show more distinct basket and tabletop-object boundaries with SRA at the displayed later frames. SRA uses the energy-preserving direction adjustment described in Appendix C.3. Over all 19 predicted frames in this replay, mean LPIPS decreases from 0.38931 to 0.34800 (10.61%), PSNR increases from 20.151 to 20.710 dB, and SSIM increases from 0.7281 to 0.7555. These scores compare the generated frames with the recorded recipient future, following the recipient-similarity convention in Appendix D.1.

The figure shows frame 13 and the final predicted frame 19. The displayed input is the initial observation, not a future target. The same tabletop region, $x \in [12, 212)$ and $y \in [90, 210)$ in the 256×256 frames, is enlarged in all five columns. Images use the decoded predictions with a common conversion to display pixels, without sharpening, color adjustment, or registration. The reported metrics are those of the displayed replay and describe this sequence rather than an aggregate task result.

D.7 CLOSED-LOOP VISUAL PLANNING ON PROCGEN

Video prediction affects a world model's usefulness through the decisions made from its imagined futures. We examine this connection in goal-reaching episodes from Procgen (Cobbe et al., 2020), using AdaWorld (Gao et al., 2025) to predict the consequences of candidate actions. Table 21 reports native task completion across evaluation cohorts. Figure 6 shows four examples from Jumper, Heist, and CaveFlyer. The displayed frames are observations obtained by executing the planned actions in the simulator.

Planning setup. The planner searches over four discrete actions with the cross-entropy method. At each replanning point, it samples 100 sequences of length 15, retains ten elites, and updates the categorical action distribution for two iterations. A sequence is scored by the maximum cosine similarity between its predicted RGB frames and a fixed goal image, after flattening the images into vectors. The planner executes the first five actions given by the final distribution's per-position modes, then replans from the newly observed history. Each episode ends on success, environment termination, or a budget of 20 executed actions. The environments use Procgen's hard setting. World-model predictions use images resized to 256×256, up to six context frames, five diffusion steps, and guidance scale 1.1. This follows AdaWorld's visual-planning design while fixing the same planner and objective for each paired comparison.

Evaluation protocol. The comparison covers Heist, Jumper, Maze, and CaveFlyer. Each pair uses the same level, initial state, goal image, and planner randomness. Success is the native level-completion event with reward 10, rather than a threshold on predicted image similarity. Each level contributes one paired trial to the cohorts in Table 21; the success rate is the fraction of completed levels. The 40-level cohort contains ten levels per game, selected before comparing the methods by model-independent random action search and verified by native replay. A qualifying solution uses at least five actions within the 20-action budget. Cohorts are reported separately rather than pooled across checkpoints or repeated levels.

Adapted dynamics. Both methods use the same Procgen-adapted AdaWorld checkpoint and its learned four-action embedding table, frozen throughout planning. SRA reuses the five previously fitted residual heads and their calibration parameters (Appendix F.1); the level index modulo five assigns one head to each episode. Neither the goal image nor environment success is used to fit the residual. These examples use the direction variant described in Appendix C.3: the correction is active at diffusion steps 0 and 1, with a requested motion-aligned strength of 6 at step 1 and a base strength of 4. The remaining component preserves the base correction's energy. As in the other direction experiments, this adjustment is generally nonlinear in the condition coordinates. Initial states, goals, backbone weights, action embeddings, and planner random draws are matched within each pair; SRA additionally evaluates the reference branch. Subsequent action sequences can differ because the predicted futures change the search distribution.

Task success. Across five groups containing 100 distinct levels, AdaWorld completes 21 levels and SRA completes 31, increasing success from 21.0% to 31.0%. The 40-level comparison gives 7.5% versus 22.5%. In that cohort, Heist, Jumper, Maze, and CaveFlyer contribute respectively $0/10$, $2/10$, $0/10$, and $1/10$ successes for AdaWorld, compared with $1/10$, $5/10$, $0/10$, and $3/10$ for SRA.

Table 21: Procgen task success. Each row is a separate evaluation cohort with one paired trial per level.

Evaluation cohort	Levels	AdaWorld	SRA
Two level groups	40	$3/40$ (7.5%)	$9/40$ (22.5%)
Five level groups	100	$21/100$ (21.0%)	$31/100$ (31.0%)

A separate three-repetition check on 20 levels gives $9/60$ versus $8/60$; repetitions here do not increase the number of distinct levels. With an alternative change-based planning objective, three completed blocks give $9/60$ versus $10/60$; one of the four planned blocks remains incomplete. These comparisons use their own checkpoints, objectives, or repeated trials and are not combined with the distinct-level cohorts.

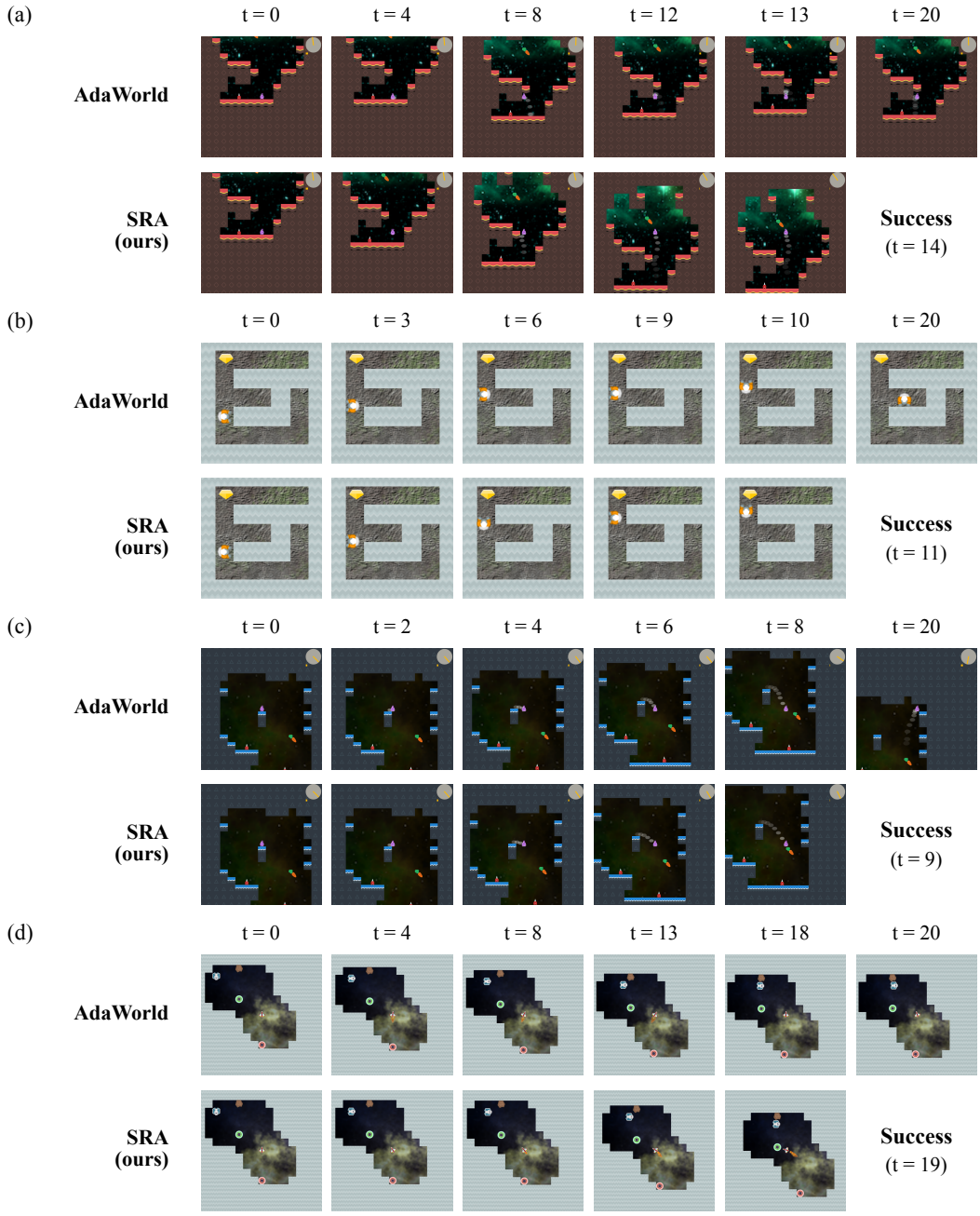


Figure 6: Examples of closed-loop planning on Procgen: (a) Jumper 533, (b) Heist 444, (c) Jumper 658, and (d) CaveFlyer 408. SRA uses the direction variant. Each pair shares the initial state, goal, adapted backbone, planner random draws, and planning objective. Images are actual simulator observations synchronized by executed action count t. SRA completes the levels in 14, 11, 9, and 19 actions; AdaWorld does not complete them within 20 actions. Success is certified by the native level-completion event with reward 10. The last displayed image of each successful rollout precedes its terminal action; a success marker replaces subsequent images. Heist uses the same fixed crop in both rows.

Other planning environments. In a separate evaluation of 35 paired trials from the public VP^2 benchmark (Tian et al., 2023), covering RoboSuite and RoboDesk tasks, SRA matches AdaWorld's observed success rate: both complete $17/26$ RoboSuite trials (65.4%) and $3/9$ RoboDesk trials (33.3%). Both methods use the same adapted backbone, initial states, goals, action interface, and MPPI planner within each pair; SRA adds the direction-adjusted residual. The planner scores imagined trajectories using the goal-image cost for RoboSuite and the image/classifier objective for RoboDesk. Success is evaluated on executed simulator trajectories: minimum native task cost below 0.05 for RoboSuite, and a native task-success event for RoboDesk.

Executed trajectories. SRA completes Jumper level 533, Heist level 444, Jumper level 658, and CaveFlyer level 408 in 14, 11, 9, and 19 actions, respectively. The paired AdaWorld rollouts do not complete these levels within 20 actions. Completion is determined by the native level-completion event with reward 10, independently of the image-similarity objective used for planning. These cases show how a correction to imagined dynamics can affect executed decisions, rather than merely the appearance of predicted frames.

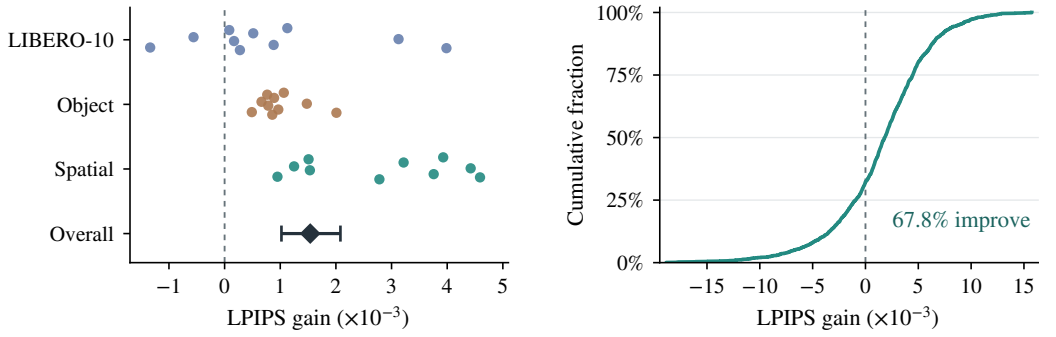
D.8 Text-conditioned generation with CogVideoX

Figure 3(c) reports six VBench dimensions on inputs sampled from OpenVid-1M with CogVideoX-2B. Table 19 gives the corresponding absolute scores. For each dimension, the plotted quantity is the adapted-model score minus the matched parent score. Bars average eight fits with equal weight; error bars show their sample standard deviation, and the points retain all eight values. These fits share the same eight development inputs; repeating adaptation does not enlarge the input set. The displayed axis rescales score differences by 100, so an axis value of 0.4 denotes a raw-score increase of 0.004, with the same scaling applied to the standard deviations. The five quality dimensions have positive mean gains; dynamic degree remains equal to the parent in every measurement. These are dimension-level evaluations on these inputs, rather than VBench Total, Quality, or Semantic aggregate scores.

Appendix C.9 specifies the response coordinate, head architecture, fitting procedure, and sampling settings shared by all eight fits. The error bars in Figure 3(c) describe variation across fits on these inputs; they are not video-bootstrap confidence intervals.

D.9 Complete prediction-gain distributions

Figure 7 expands Figure 3(a) with the empirical distribution of all 1,500 paired clip gains. The 30 task means each average 50 clips, giving every task equal weight. The overall confidence interval resamples tasks and is conditional on the five fitted heads and their fixed routing; it does not include retraining variability. All positive and negative task and clip gains are retained.



(a) All 30 task means

(b) All 1,500 paired clips

Figure 7: LPIPS gains across LIBERO tasks and clips. Positive values indicate lower LPIPS for SRA-base than AdaWorld. (a) Each dot averages 50 paired clips from one task; the diamond shows the overall mean with a 95% task-bootstrap interval, conditional on five fitted heads and fixed routing. (b) The empirical cumulative distribution includes all 1,500 paired clips.

D.10 Component ablations and matched residual comparisons

The component comparisons reuse SRA's fitted heads and change the reference, correction direction, or operator rank. Direct guidance and nonlinear heads instead test competing correction forms. The component study below uses training-task rollouts; the held-out base comparison in Table 2 and the cross-demonstration direction comparison in Table 4 use separate evaluation protocols.

Table 22: Ablations with a frozen AdaWorld backbone and fitted heads on 90 LIBERO training tasks, using three fixed source positions per task. All configurations use 133 backbone evaluations per rollout. LPIPS averages 270 rollouts; FVD averages the three position-level estimates. Reference routes specifies where the action is neutralized for the reference evaluation; the parent prediction is unchanged. SRA includes the optional direction adjustment.

Configuration	Reference routes	Rank	LPIPS↓	FVD↓
AdaWorld (matched calls)	–	–	0.307624	764.948
SRA	Both	32	0.306705	751.318
w/o direction adjustment	Both	32	0.306772	753.945
Single-route reference	Cross-attention	32	0.307604	761.347
Single-route reference	Global vector	32	0.307531	759.125
Rank-8 projection	Both	8	0.306679	751.097
Rank-16 projection	Both	16	0.306705	751.408
Rank-24 projection	Both	24	0.306694	751.681

LIBERO ablation setup. All eight configurations reuse the same frozen five-head collection and its task-to-head routing on 90 LIBERO training tasks. Three fixed source positions (25, 26, and 27) give 270 rollouts per configuration. A distinct noise realization is assigned to each position and shared across configurations. The positions are repeated observations within a task, not independent fits. No head is retrained and no configuration is selected across positions. LPIPS, PSNR, SSIM, and motion error average the 270 rollouts. FVD is computed over the 90 videos at each position and then averaged over the three positions; it is not FVD over a pooled set of 270 videos. Table 11 reports the remaining metrics for these same rollouts.

The correction is active at sampling steps 0 and 1, with a base dose of 4. At step 1, the optional adjustment requests a dose of 6 along the motion direction while preserving the squared correction norm, as specified in Appendix C.3. Every configuration makes 133 backbone evaluations per rollout, including 38 reference evaluations; the native sampler requires 95. The matched-call parent evaluates the reference and discards its output. This controls the additional evaluation itself, without testing the best alternative use of that computational budget. Direction adjustment preserves energy when compared at the same state; trajectories can subsequently diverge, so the comparison does not assert equal accumulated energy along different rollouts. The adjusted correction is generally nonlinear in the condition coordinates and does not inherit the base branch's zero conditional Hessian.

Reference routes and rank projection. The cross-attention-only reference replaces the action with its reference mean in the cross-attention input and leaves the global action vector unchanged; the global-vector-only reference does the converse. Neither operation removes a module from the parent model. Thus the single-route rows measure incomplete neutralization of the reference, rather than a backbone architecture ablation or a test of semantic disentanglement.

For each reference state, the slope is reshaped into a matrix with 32 action-coordinate columns, truncated by singular-value decomposition, and rescaled to its original Frobenius norm. Before rescaling, the minimum retained squared-norm fractions across evaluated states are 99.267%, 99.871%, and 99.981% for ranks 8, 16, and 24, respectively (rounded). All projected variants retain the same 149,304 parameters per active head and add the projection computation; they are not compressed implementations. Their paired intervals in Table 23 neither identify rank 8 as superior nor establish statistical equivalence to rank 32.

Reference construction. AdaWorld receives the action through both cross-attention and a global conditioning vector. Neutralizing both routes in the reference evaluation lowers LPIPS by $0.899\times$

Table 23: Paired LPIPS reductions for Table 22, in 10^{-3} units. Positive values favor the first configuration. Each interval resamples 90 task means after averaging the three positions within each task (20,000 percentile-bootstrap resamples); it excludes refitting variability.

Comparison	Reduction	95% interval
SRA vs. matched-call AdaWorld	0.920	$[0.295, 1.545]$
SRA vs. no direction adjustment	0.067	$[0.025, 0.109]$
SRA vs. cross-attention-only reference	0.899	$[0.737, 1.075]$
SRA vs. global-vector-only reference	0.827	$[0.672, 0.993]$
Rank 8 vs. rank 32	0.025	$[-0.032, 0.082]$
Rank 16 vs. rank 32	-0.001	$[-0.025, 0.025]$
Rank 24 vs. rank 32	0.010	$[-0.017, 0.038]$

10^{-3} and 0.827×10^{-3} relative to neutralizing either route alone, with positive task-bootstrap intervals. FVD also decreases in both comparisons. The parent retains both conditioning routes in every configuration: the change concerns the information used to construct the correction. The single-route references still contain the current action, and recover little of the full reference's perceptual gain.

Correction direction. With both reference routes neutralized, SRA lowers LPIPS by 0.920×10^{-3} and FVD by 13.63 relative to the matched-call parent. Removing the optional direction adjustment increases LPIPS by 0.067×10^{-3}, with a 95% interval of $[0.025, 0.109] \times 10^{-3}$, and raises FVD by 2.63. The adjustment preserves correction energy at a given state, so this comparison tests its allocation rather than a larger local perturbation. A separate Franka comparison also favors the motion-based adjustment over unchanged and random directions in LPIPS (Appendix D.4).

Operator rank. Projecting the state-dependent slope to rank 8 retains more than 99.26% of its squared Frobenius norm before rescaling. Its LPIPS and FVD remain close to the rank-32 result; the paired LPIPS interval includes zero. Ranks 16 and 24 show the same pattern. The learned conditional response thus admits a substantially lower-rank approximation on these trajectories. This is a projection of the fitted operator, without retraining or reducing stored parameters. Table 23 gives all paired intervals. The direct-guidance and nonlinear-head comparisons below do not yet establish a stable advantage attributable to conditional linearity alone.

Direct guidance and nonlinear alternatives. Separate development comparisons examine whether conditional linearity itself explains the gain. On AdaWorld, the linear head is compared with direct reference guidance, a nonlinear transformation of the condition coordinates, and a jointly nonlinear state–condition residual anchored to vanish at the reference. The latter uses 153,912 parameters, compared with 149,304 for the linear head. Learned heads share five-fold routing, 5,760 fitting updates, three learning-rate settings, and five dose policies. Two generation blocks each cover five development tasks. These comparisons use a different insertion schedule from Table 22, with 190 backbone evaluations for corrected rollouts versus 95 for the native parent.

Across the 15 learning-rate–dose combinations, the linear head has lower mean LPIPS than the jointly nonlinear head in five combinations, and in both blocks in four. None of the family-adjusted intervals across the 60 predeclared contrasts establishes an advantage for the linear head; one LPIPS and four embedding-cosine contrasts favor direct guidance. FVD is lower than the parent in both blocks for all five direct-guidance policies, four of 15 linear-head configurations, and three of 15 jointly nonlinear configurations. These reused development tasks do not support a general ordering of the three correction classes.

The corresponding Wan text-to-video development matrix contains two generation blocks on eight prompts, comparing direct guidance and three learned residual forms under four dose policies. At the calibrated dose, the nonlinear condition transformation has lower FVD than the linear head in both blocks (3834.51 versus 3999.19, and 3923.61 versus 3935.16). The calibrated direct correction is zero and reproduces the parent. Increasing the uncalibrated dose to 4 worsens both FVD and imaging quality for all four correction forms in both blocks. These results concern a small development set and six individual VBench dimensions, not the full VBench aggregate.

Conditional calibration and guided error. A separate Wan diagnostic evaluates fixed heads on cached calibration states, without generating new videos. Two fitted heads are evaluated on the same eight source videos excluded from dose fitting, with four correlated states per video. The calibrated linear correction changes mean conditional MSE by -7.79×10^{-5} and -8.49×10^{-5}, while guided MSE increases by 1.286×10^{-3} and 1.270×10^{-3}. Conditional error decreases but guided error increases at 17 of 32 and 16 of 32 states, respectively. This illustrates the distinction between a conditional fitting objective and error after classifier-free guidance. The 32 states are not independent video samples, and these fixed-state measurements do not measure rollout quality or uniquely implicate conditional linearity.

E Benchmark Evaluation

This appendix details the OpenVE-Bench category evaluation and the scope of the VBench dimensions reported in the main text. Sampling and aggregation are specified in Appendix F.

E.1 Editing categories

OpenVE-Bench (He et al., 2025) contains 431 source-video and editing-instruction pairs across eight categories. We compare LucyEdit, SRA, and a no-op that returns the source video unchanged, using Qwen3-VL-32B-Instruct (Bai et al., 2025) with the benchmark's category-specific prompts. Each response gives three ratings between 1 and 5. Their names and criteria differ by category: for example, style transfer measures style fidelity, content preservation, and temporal stability, whereas object addition measures prompt compliance, visual and temporal naturalness, and physical and detail coherence. We average the three unrounded ratings per input, then average inputs within each category or over the complete benchmark. Overall is sample-weighted, not an unweighted average of categories.

Table 24 reports all eight categories and the Overall summary used in Table 3. SRA has a higher mean score in each category. The complete-benchmark mean increases from 3.566 to 3.616, a difference of 0.049 computed before rounding. Overall weights the category means by their input counts across all 431 inputs. Paired intervals are reported for every category and Overall. Table 3 displays the four categories with gains in all three component means. The no-op comparison also shows why preserving source appearance alone is insufficient for assessing an edit. Appendix C.8 gives implementation details, with Bernini results in Appendix D.2.

Table 24: All eight OpenVE-Bench editing categories. Overall covers all 431 inputs. No-op returns the source unchanged; Δ is SRA minus LucyEdit, with paired 95% intervals for category and Overall gains.

Category	N	LucyEdit	SRA	No-op	Δ	95% CI
Style transfer	58	3.236	3.299	2.983	$+0.063$	$[-0.190, 0.322]$
Background replacement	59	3.362	3.373	2.785	$+0.011$	$[-0.107, 0.136]$
Local modification / replacement	65	3.564	3.631	1.805	$+0.067$	$[-0.077, 0.205]$
Object removal	59	3.616	3.633	3.768	$+0.017$	$[-0.158, 0.186]$
Object addition	67	3.418	3.502	1.299	$+0.085$	$[-0.035, 0.209]$
Subtitle editing	50	4.273	4.293	4.360	$+0.020$	$[-0.187, 0.213]$
Creative editing	30	3.767	3.900	1.078	$+0.133$	$[0.000, 0.311]$
Camera editing	43	3.496	3.519	3.667	$+0.023$	$[-0.054, 0.101]$
All-category aggregate (Table 3)						
Overall	431	3.566	3.616	—	$+0.049$	$[-0.012, 0.111]$

The 95% percentile intervals use 10,000 paired source-cluster bootstrap resamples, retaining both methods for each source. They quantify source-sampling uncertainty for one fixed head, one generation repetition, and one judge response per method and input, rather than variability across retraining or judge repetitions. All 1,293 responses from the three methods yield the required three labels. The evaluator prompts ask the second and third ratings not to exceed the first; 268 responses violate this ordering. The scores are retained without clipping or selective removal. Strict label

parsing avoids including unrelated numbers from the evaluator's explanation; the resulting unrounded means can differ slightly from the official display parser's rounded averages. The same evaluator and scoring rule are used for both methods.

E.2 VIDEO GENERATION DIMENSIONS

VBench combines seven video-quality dimensions and nine semantic dimensions with prescribed normalization and weights (Huang et al., 2024). The six custom-input measures in Figure 3(c) omit temporal flickering and all nine semantic dimensions. They describe individual dimensions of video quality; VBench Quality, Semantic, and Total scores cannot be recovered from these six measurements.

F EVALUATION DETAILS

Datasets and sampling settings. The same-context LIBERO result uses five fixed base heads with a fixed clip-to-head routing and a predeclared 1,500-clip test. Cross-context evaluation pairs each recipient demonstration j with donor $(j + 25) \bmod 50$ from the same task. The donor supplies the latent action and the recipient supplies the initial frame. Comparison frames are the recipient's original future, beginning after its first frame; donor and recipient clips are not aligned by physical state or action phase. Thus these scores quantify similarity to the recipient trajectory, not error against a known donor-action outcome. FVD uses the recipient-future collection as its reference as well. Franka contains 91 public validation episodes, whose possible use during parent pretraining is unknown. Bernini uses four development videos; Wan T2V, CogVideoX T2V, and Wan I2V each use eight development inputs for the reported comparisons. Within each comparison, parent and adapter share the sampler, guidance settings, resolution, and video length; Appendix C specifies these settings for each model.

F.1 DATA ROLES AND MODEL ROUTING

Table 25: Data used for fitting, calibration, selection, and evaluation.

Study	Head fitting	Strength / configuration choice	Evaluation
AdaWorld	Five heads, each fit on 72 of 90 training tasks	Strengths: position 0 of its 72 fit tasks; candidate checks on training tasks	30 disjoint test tasks, 50 clips each; Franka and Jaco external collections
Bernini	24 videos, four cached steps each	Eight calibration videos; four development videos for configuration choice	Same four development videos and repeated generation
LucyEdit	128 local-modification pairs	16 calibration pairs; the first eight of 16 development pairs for selection	431 OpenVE-Bench inputs across eight categories
Wan2.1 T2V	128 paired flow examples	Eight dose-fit examples, eight confirmation examples; eight development prompts	Same eight development prompts; eight response fits
CogVideoX T2V	128 video–text examples	Eight dose-fit examples, eight confirmation examples; eight development inputs for configuration choice	Same eight development inputs; eight reported fits
Wan2.2 I2V	128 videos	Eight dose-fit videos, eight confirmation videos	Same eight development inputs across eight fits

In AdaWorld's training population, sorted task index modulo five defines the held-out group for each head: 72 tasks fit the head and 18 do not. Strength fitting also excludes those 18 tasks. Earlier development clips inform candidate choice; the affine-only candidate is subsequently checked on positions 18–21 of all 90 training tasks, using each task's held-out head, for 360 clips. This check

assesses candidate-controlled latent trajectories, not a new video benchmark. The 30 formal test tasks are disjoint from the 90-task population. Each test clip is assigned one already-fitted head by its demonstration index modulo five, so the result is neither an ensemble nor a single-head evaluation. Each task contributes ten clips to each head. Franka and Jaco route by episode index modulo five. Task-level intervals apply to this fixed routing rule. Training-task architecture ablations remain development evidence.

F.2 Metrics and reference targets

Pixel and perceptual measures. AdaWorld evaluates the 19 generated future RGB frames in $[0, 1]$, excluding the initial recipient frame. PSNR and SSIM use the PIQ implementations (Kastryulin et al., 2022) with data range 1; LPIPS uses PIQ's VGG16-based network and learned LPIPS weights with its native input normalization. Framewise losses are averaged within each video before task or episode aggregation.

Motion magnitude. For an RGB video $\mathsf{V} \in [0,1]^{T_v \times C_v \times H_v \times W_v}$ with $T_v \geq 2$, define

$$m(\mathsf{V}) = \frac{1}{(T_v - 1)C_v H_v W_v} \sum_{t=2}^{T_v} \sum_{a=1}^{C_v} \sum_{h=1}^{H_v} \sum_{w=1}^{W_v} |V_{t,a,h,w} - V_{t-1,a,h,w}|.$$

Here $V_{t,a,h,w}$ is a scalar pixel value, t indexes video frames and a the color channel. The reported motion error is $|\log((m(\widehat{\mathsf{V}}) + 10^{-6})/(m(\mathsf{V}) + 10^{-6}))|$. The additive floor is applied to both magnitudes; the ratio is not clipped in this metric. The separate temporal-failure indicator tests a ratio outside $[0.1, 10]$, with an additional static-reference rule. Neither quantity is VBench motion smoothness.

Video features and distribution distance. The AdaWorld evaluator rounds RGB frames to 8-bit values, resizes them bilinearly to 224×224 with no corner alignment, and rescales them to $[-1, 1]$. It uses the 400 output logits of the Kinetics I3D classifier (Carreira & Zisserman, 2017), not a 1,024-dimensional intermediate activation. The classifier uses its native spatiotemporal average pool and averages the output logits over the remaining temporal dimension. FVD compares empirical feature means and unbiased covariances. ECS averages cosine similarity between the generated and designated reference clip features in matched row order; it is not text similarity. The Bernini and Wan FVD records likewise use 400-dimensional I3D features, but their clip lengths and populations differ, so their FVD values are not comparable across tasks. At four or eight videos, covariance rank is at most three or seven. These development distances do not support a large-sample benchmark claim.

Action-matching scores. Figure 3(b) uses three distance contrasts between a generated action sequence, the episode's original action, and the substituted action. At each transition, the contrast is distance to the original action minus distance to the substituted action; it is averaged over the 19 predicted transitions within each episode. Latent distance uses the mean squared coordinate difference in the frozen encoder's 32-dimensional action space, divided coordinatewise by training-set variances. Latent direction uses cosine distance between the unstandardized action vectors. Decoded action uses the standardized physical-action distance below. Thus the first two scores share an encoder but measure magnitude-sensitive and angular discrepancies, respectively. Each plotted gain subtracts the parent's episode score from that of SRA-direction, with positive values indicating stronger relative matching to the substituted action. The 109 target episodes draw on 21 distinct source episodes; the displayed intervals resample target episodes.

A fixed linear predictor maps generated latent actions to standardized physical actions $\widehat{\boldsymbol{a}}$. Write $\boldsymbol{a}_r, \boldsymbol{a}_d, \widehat{\boldsymbol{a}} \in \mathbb{R}^{d_a}$ for the recipient, donor, and predicted actions. The factor $1/d_a$ below averages over action coordinates; the norms remain Euclidean. The margin averages the contrast

$$\begin{aligned} M(\widehat{\boldsymbol{a}}) &= \frac{\|\widehat{\boldsymbol{a}} - \boldsymbol{a}_r\|^2 - \|\widehat{\boldsymbol{a}} - \boldsymbol{a}_d\|^2}{d_a} \\ &= \frac{2\langle \widehat{\boldsymbol{a}}, \boldsymbol{a}_d - \boldsymbol{a}_r \rangle + \|\boldsymbol{a}_r\|^2 - \|\boldsymbol{a}_d\|^2}{d_a}. \end{aligned}$$

over time within an episode and then equally over 109 episodes. A positive value indicates a relative preference for the donor over the recipient. An increase need not reduce donor error. The measured

0.0331 (95% interval $[0.0259, 0.0404]$) is the paired mean margin of SRA-direction minus that of the native parent, with an episode-bootstrap interval. Held-out recorded-transition $R^2 \simeq 0.399$ does not establish calibration on generated videos.

Randomness and uncertainty. The LIBERO direction comparison in Tables 4 and 10, and the Franka prediction comparison, average three paired generation repetitions with fixed clip-to-head routing to the same five fitted heads, whereas each AdaWorld refit retrains the complete five-head collection and uses shared generation noise for comparison. Jaco and the direction controls also share generation noise across methods. Bernini varies generation noise for one head; the Wan objective comparison includes independently trained heads.

For paired confidence intervals, we first average generated samples within each LIBERO task or Franka episode, then resample those units. The percentile bootstrap uses 100,000 resamples of the 30 LIBERO task means in each setting, and 10,000 resamples of the 91 Franka episode means, with replacement. The paired LIBERO analysis in Figure 3(a) gives the 95% LPIPS-improvement interval $[1.021, 2.081] \times 10^{-3}$. Intervals are specific to their underlying paired records and are not transferred to updated aggregate means. They describe sampling uncertainty for fixed fitted models and exclude training variability. For each LIBERO generation run, FVD is computed once from all 1,500 clips produced by the five-head routing; repeated-generation FVD values are then averaged. It is not an average of five head-specific FVDs. The same-context base result is one such complete run. Franka also aggregates the complete episode collection within each run before averaging the repeated-generation FVDs. Each replication uses its own parent baseline; videos from different settings are not pooled. The earlier 500-clip replacement study likewise averages three generation seeds.

Editing evaluation repetitions. The LucyEdit comparison covers all 431 OpenVE-Bench inputs for one fit, one generation repetition, and one judge evaluation per method and input. Its paired intervals are specified in Appendix E.1.

Frame-difference error. For the paired editing evaluation, let $\widehat{\mathsf{V}}_t$ and V_t be the generated and target RGB frame tensors, with pixel values in $[0, 1]$. The frame-difference MSE is

$$\frac{1}{(T_v - 1)C_v H_v W_v} \sum_{t=2}^{T_v} \left\| (\widehat{\mathsf{V}}_t - \widehat{\mathsf{V}}_{t-1}) - (\mathsf{V}_t - \mathsf{V}_{t-1}) \right\|_{\mathrm{F}}^2,$$

where T_v is the number of frames and C_v, H_v, W_v are their channel and spatial dimensions. The evaluator computes this average within each video and then averages over videos. A lower value means closer agreement with the target's adjacent-frame changes. This pixelwise measure requires aligned pairs and does not by itself establish perceptual smoothness or instruction compliance. Its small increase in the Bernini comparison has no reported uncertainty interval, so it cannot alone establish a perceptible or statistically reliable degradation.

G Additional Background and Design Context

Video generators and world models. Video diffusion extends image generation along a temporal axis (Ho et al., 2022b;a; Singer et al., 2023). Latent representations and transformers make larger and more flexible video models possible (Blattmann et al., 2023; Yang et al., 2025; Team Wan et al., 2025), while Tune-A-Video, Text2Video-Zero, and AnimateDiff retain knowledge learned from images (Wu et al., 2023; Khachatryan et al., 2023; Guo et al., 2024). For world models, a plausible video is only part of the objective: predictions are used for interaction, imagined behavior, or planning (Ha & Schmidhuber, 2018; Hafner et al., 2020; 2021; Micheli et al., 2023). Genie and AdaWorld learn latent-action interfaces from video (Bruce et al., 2024; Gao et al., 2025); iVideoGPT, DIAMOND, and Diffusion Forcing study interactive tokenization, diffusion-based dynamics, and causal denoising (Wu et al., 2024; Alonso et al., 2024; Chen et al., 2024). Alongside video-quality evaluation, the Procgen success rates and trajectories in Appendix D.7 examine how adapted predictions affect goal-directed planning. Success is measured by native task completion in the evaluated level cohorts.

Adaptation of a frozen model. Adapters, low-rank updates, and side networks restrict the learned change (Houlsby et al., 2019; Hu et al., 2022; Zhang et al., 2020); HyperNetworks and FiLM

make weights or feature-wise modulation depend on an input (Ha et al., 2017; Perez et al., 2018). ControlNet and T2I-Adapter supply spatial control, and IP-Adapter supplies image prompts (Zhang et al., 2023; Mou et al., 2024; Ye et al., 2023). DreamBooth, Textual Inversion, and Custom Diffusion specialize generation to supplied concepts (Ruiz et al., 2023; Gal et al., 2023; Kumari et al., 2023). Zero initialization also has clear precedents in ControlNet's zero convolutions and LoRA's zero update factor (Zhang et al., 2023; Hu et al., 2022). SRA constrains the conditional function of the terminal output correction; low-rank adaptation constrains an internal weight update.

Editing and preservation. Editing makes the tension between change and preservation explicit. SDEdit balances input fidelity with a generative prior, Prompt-to-Prompt changes cross-attention, and Null-text Inversion optimizes the unconditional embedding (Meng et al., 2022; Hertz et al., 2023; Mokady et al., 2023). InstructPix2Pix learns instruction following; Plug-and-Play Diffusion Features reuses internal spatial features (Brooks et al., 2023; Tumanyan et al., 2023). In video, TokenFlow and FateZero preserve structure through feature correspondence and attention reuse (Geyer et al., 2024; Qi et al., 2023), while Bernini couples semantic planning with diffusion rendering (Bernini Team et al., 2026). Our Bernini adapter acts on a source-video coordinate inside that renderer, with the semantic planner and edit instruction retained.

Guidance and deployment. Classifier and classifier-free guidance modify conditional sampling (Dhariwal & Nichol, 2021; Ho & Salimans, 2022); interval guidance changes when guidance is applied, and autoguidance changes its source (Kynkäänniemi et al., 2024; Karras et al., 2024). Adaptive projected guidance separates and rescales components of the CFG update (Sadat et al., 2025). Our optional direction rule acts on a different quantity: it reallocates the fixed energy of a learned correction along a temporal direction supplied by the parent. CFG analysis also separates an instantaneous vector-field identity from the distribution a sampler produces (Bradley & Nakkiran, 2025). Here, the corresponding practical concern is that guided, composed error differs from the conditional error used for calibration. Appendix B.2 makes this distinction explicit for the present correction.

Evaluation. FVD and VBench measure different aspects of video quality (Unterthiner et al., 2018; Huang et al., 2024), and FVD can underweight temporal defects (Ge et al., 2024). The perception–distortion trade-off in image restoration (Blau & Michaeli, 2018) is a reason to use complementary measurements. It does not prove that our video metrics must trade off, or explain the cause of a particular regression.

H Discussion

SRA makes a specific choice about the form of adaptation: a reference-dependent operator carries state dependence, while the selected condition coordinates enter linearly. The held-out prediction result establishes gains from the base construction, and the editing and generation evaluations demonstrate further benefits across conditioning interfaces. Reference-route comparisons show that the information used to construct the operator matters; direction controls show that its local norm alone does not determine the outcome of sampling.

These findings support the method as a practical adaptation scheme. The Gaussian projection analysis characterizes its representational restriction, and the sampler analysis describes propagation; neither determines which correction class will achieve the best generation quality. The existing direct-guidance and nonlinear-head comparisons do not establish a stable ordering. A finite-data estimation advantage remains a possible explanation to investigate, rather than the premise of the reported performance claims. Training-set-size comparisons would address that additional question. Appendix I states the present scope.

I Limitations

The analysis characterizes restricted response approximation and the local effect of residual insertions along a finite sampler. It does not provide a finite-sample generalization bound or a guarantee of improved perceptual quality. Its Gaussian coordinate law and local regularity assumptions have not

been established for arbitrary pretrained models. Conditional linearity applies to the added branch at a fixed state and reference; it does not imply global smoothness of the generated trajectory.

SRA relies on a choice of reference, condition coordinates, and insertion schedule. A general procedure for learning these choices jointly is outside the present scope. The reference evaluations also add sampling cost despite the small number of trainable parameters.

The planning examples concern short goal-reaching rollouts. They do not establish robustness of the resulting policies across environments or long horizons, where prediction errors can compound through subsequent actions and observations. Systematic evaluation of these feedback effects remains outside the present study.