WHAT CONTROLS MOTION IN VIDEO DIFFUSION MODELS? A MECHANISTIC STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

Motion is central to video generation models, yet what inside the model controls it remains largely unexplored. Some methods control object motion by training an added module on video data. Others require no training and edit the latent or the attention at inference time. These methods change the generated motion, but they do not show what inside the model produces the change. We present a mechanistic study of that question. We train a video diffusion transformer on synthetic videos, where the launch angle of a ball is controllable, and we read that angle in degrees from generated videos. Its launch angle is a training condition we can set exactly, which a pretrained model does not provide. We present three findings. First, we add one vector at every token position at the output of one transformer layer, the standard form of activation steering. The shared vector barely rotates the trajectory. Second, a token-wise field, which adds a separate vector at each token position, rotates the trajectory by the requested angle at the same layer. Third, the field's rotation depends on a small group of conditioning-related attention heads, and holding them at their un-intervened outputs removes most of it. On the pretrained Wan2.1-1.3B, all three findings hold on synthetic clips inverted into the model. Under natural-language prompts, we split the change in activations between two prompts into its average over positions and the remainder; adding the remainder flips the subject's facing, while adding the average does not. Together, these experiments characterise the intervention structure and internal components that mediate motion control in the tested models.

1 INTRODUCTION

Motion is central to what video generation models produce. Many methods have been proposed to control motion in video models. Some take a path or a bounding box, and train an added module on video data (Yin et al., 2023; Wang et al., 2024b;a; Zhang et al., 2025). Others require no training, and edit the latent or the attention at inference time (Xiao et al., 2024; Choo et al., 2026). These methods change the motion, but they do not explore what inside the model carries the change. We ask this at the level of internal activations: what controls the motion of an object in a video diffusion model?

Adding one direction to every position of a transformer's activations is the standard form of activation steering (Li et al., 2023), and it also works in video models. It flips a video encoder's judgment of physical plausibility and changes what a video diffusion transformer generates (Alam, 2026; Kundu et al., 2026). Other results point the other way. In video encoders, a physical quantity can be spread over dozens of directions that a single vector cannot move (Joseph et al., 2026). And in video diffusion transformers, the layers where a probe reads a quantity best need not be the layers that move it most (Esmati et al., 2026). None of these reads the attribute in its own units off the generated pixels.

Two components are needed to answer this question. One is a counterfactual pair, two generations that differ only in one conditioning number (here, the launch angle). The other is a readout that measures the changed attribute. We build both by training a video diffusion transformer on projectile motion in a world we synthesize (Section 3.1). The launch angle input to the model is the ground truth, and the angle in the generated videos is read in degrees. A linear probe reads the launch angle accurately from the activations, so a single direction in activation space already tracks the angle.

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

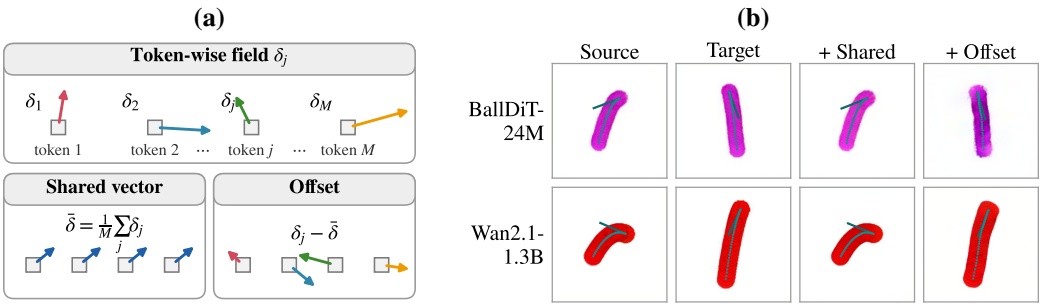


Figure 1: **Splitting a token-wise field into a shared vector and an offset.** (a) The average $\bar{\delta}$ of the vectors δ_j is the shared vector, and what remains at each token is the offset. (b) Source and Target are two generations that differ only in launch angle, by 90°. Each panel overlays the 33 frames of one video; dots mark the ball centre in each frame, and the line starts at the first centre and points along the fitted launch direction. A field that rotates the trajectory by 90° is split into its shared vector and offset, and each is added to the Source generation. In both models the shared vector barely turns the trajectory, while the offset reaches the target.

Adding that kind of direction should therefore move it. However, it does not. A shared vector at our injection site barely rotates the rendered trajectory. Additions that allow the vector to vary across positions, which we call token-wise fields, reach the requested rotation. We split a working field into two parts (Figure 1a). The shared vector is its average over all tokens, and the offset is what remains at each token. The shared vector carries almost none of the rotation, while the offset carries almost all of it (Figure 1b).

Our contributions are threefold:

- We introduce a controlled setting and an analysis procedure for studying motion in video diffusion transformers. A model trained on simulated projectile motion lets us request a launch angle exactly and read it in degrees from the generated pixels; we fit a token-wise field that reaches the request, split it into its shared vector and offset, and pin attention heads to test which components it depends on. We apply the same procedure to Wan2.1-1.3B.
- We show that at one fixed site a shared vector barely rotates the trajectory while a token-wise field does. Splitting a working field shows that its offset carries almost all of the rotation. The rotation reaches the request at a low rank, and image quality needs a higher rank.
- We show that the field acts through a small group of conditioning-related attention heads: pinning five of them removes most of the rotation, while layer-matched random groups do not.

2 Related Work

Steering methods in language and video models. Many methods steer a model by adding one direction, differing in where it comes from. In language models, it is a probe-ranked difference of class means added to every token (Li et al., 2023) or the difference of two prompts' activations (Turner et al., 2023), which Mishra et al. (2026) show reaches states no prompt can. In video, a concept-activation direction flips a video world model's physical-plausibility judgment (Alam, 2026). For video diffusion generation, which is our setting, Kundu et al. (2026) add one supervised-PCA direction at a single layer of nine models, scored by a safety classifier. In their main experiments, Hong et al. (2026) vary a control signal over denoising step and layer, but add it identically to every text token. These studies leave open whether a fixed vector shared across video tokens can steer a specified motion attribute measured in generated pixels.

Probing and causal interventions. In language models, Elazar et al. (2021) remove a property from BERT and find that how well a probe decodes it does not predict how much removing it changes prediction. On a bracket task, Sharma et al. (2026) find that a subspace decoding the count of open brackets can be ablated without changing the model's output. In video, Esmati et al. (2026) probe

three video diffusion models and find the blocks that decode a quantity best are not the ones that perturbing moves most. All three read out through a label, a probe, or perturbation sensitivity, but none adds the decoded direction itself and measures the attribute it moves in the generated pixels.

Intervention and localisation. Other interventions add different things at different positions. In video diffusion, TokenDial keeps one direction shared across tokens and lets only its magnitude vary with position (Liu et al., 2026); in language models, Jin et al. (2026) learn a different direction for each token; and h-space editing in image diffusion varies with position by construction (Kwon et al., 2023; Haas et al., 2024). These works show that position-dependent interventions can work, but they do not ask whether, within a working intervention, the effect comes from the part that varies across positions or from the part shared by all of them. A second line of work finds which components of the model a change passes through. Jung et al. (2026) find motion-related attention heads in video diffusion, and Wang et al. (2026) find a few heads responsible for a concept in text-to-image diffusion. Zhang & Nanda (2024) caution that what such localisation finds depends on how the experiment is set up. We combine the two: we split a working field into its shared and position-dependent parts, and then test which heads it depends on.

3 METHODS

Section 3.1 describes the model we train and how we read attributes such as the launch angle from generated frames. Section 3.2 describes how we add interventions to the activations and how we find the attention heads they act through.

3.1 CONTROLLED SETUP

Model and training objective. We train a diffusion transformer from scratch on videos generated by a two-dimensional projectile simulator, following Kang et al. (2025). We refer to this 23.8M-parameter model as *BallDiT-24M* and use it unless stated otherwise; Table 2 gives the full configuration. Each video is encoded by a frozen VAE into a clean latent x. Given Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$ and noise level $\sigma \in [0, 1]$, we construct

$$z_\sigma = (1 - \sigma)\, x + \sigma\, \varepsilon, \tag{1}$$

and train the transformer f_θ to predict the flow target $\varepsilon - x$ from z_σ, σ, and the conditioning c:

$$\min_\theta \; \mathbb{E}_{x,\varepsilon,\sigma} \left\| f_\theta(z_\sigma, \sigma, c) - (\varepsilon - x) \right\|_2^2. \tag{2}$$

Architecture. A $1 \times 2 \times 2$ patch embedding converts the latent into $M = 576$ tokens of width $D = 384$, denoted $X^{(0)}$. The tokens span nine temporal slices, with each token corresponding to one spatial patch within a slice. The noise level σ and attribute conditioning c are embedded and combined into a vector e. At each transformer block, e is mapped to adaptive-layer-normalisation parameters: a shift β, scale γ, and gate g, with separate parameters for the attention and feed-forward sublayers. Block ℓ then applies

$$X \leftarrow X + g_a \odot \mathrm{Attn}\big(\mathrm{LN}(X)\,(1 + \gamma_a) + \beta_a\big), \tag{3}$$

$$X \leftarrow X + g_f \odot \mathrm{FFN}\big(\mathrm{LN}(X)\,(1 + \gamma_f) + \beta_f\big), \tag{4}$$

where LN denotes layer normalisation without learned affine parameters, Attn is six-head self-attention, and FFN is a two-layer feed-forward network. We denote the block output by $X^{(\ell)}$; our activation interventions are added to these block outputs. Because the conditioning enters through the block-wise modulation parameters β, γ, g, the modulation itself is shared across token positions.

Paired generation. We generate twice from the same starting noise and the same denoising schedule, changing exactly one conditioning attribute between the two runs. Fixing everything else isolates the effect of that attribute on the generated video. The four attributes studied are the ball's launch angle, its initial speed, its hue and the whole-frame brightness.

Readout and success criterion. We measure each controlled attribute directly from the generated pixels and compare it with the simulator ground truth. For each intervention, the measurement records how far the generated video moves toward the requested change, and image quality is tracked separately through three metrics, namely the ball's pixel count, the number of lost frames and the worst-frame saturation. Appendix A gives the exact readout procedure. An intervention *reaches the request* when, over held-out videos, the median difference between the obtained and requested change is within a tenth of the request and fewer than 33 of the 3,300 frames lose the ball. The model follows its conditioning closely: on 100 held-out conditions, it reproduces the requested launch angle and hue with median errors of $0.69°$ and $0.98°$, respectively.

3.2 Interventions

An *intervention* adds one fixed vector per token position to the output of one transformer layer without changing the model weights. We vary two aspects of this intervention: where the vectors come from (*direction*) and how they are distributed across positions (*form*).

3.2.1 The token-wise field

The direction is the vector added to the activations, and we obtain it from three sources: the weight vector of a ridge probe fitted to predict the attribute, a vector found by optimisation, and the difference between activations recorded under two conditions. The form is how that vector is spread across the M tokens of a layer. We call the added array a *token-wise field* $\delta \in \mathbb{R}^{M\times D}$; row δ_j is added at token j. We write it as $\delta = AV$ with $A \in \mathbb{R}^{M\times R}$ and $V \in \mathbb{R}^{R\times D}$: each row of V is one direction in activation space, and each row of A is one token's mixing weights. The two extremes are $R = 1$ with A the all-ones vector, a single vector shared by every token, and $R \geq D$, a free vector per token; a rank-R field lies in between. The rank of δ is at most R, so R is the number of directions the field can use. Our full-rank field is optimised with $R = M = 576$ and has rank $D = 384$. With $R = 1$ and A free, every token's vector lies along the same direction, while its norm varies freely across tokens.

An optimised field is fitted by minimising the squared error between the generated latent and the latent generated under the changed condition, with the field's norm held below a cap κ:

$$\min_{\delta} \frac{1}{N}\sum_{i=1}^{N} \left\| z_i(\delta) - z_i^{\star} \right\|_2^2 \qquad \text{subject to} \qquad \|\delta\|_F \leq \kappa, \tag{5}$$

where N is the number of fitting clips, $z_i(\delta)$ is the latent generated for fitting clip i with the field applied and $z_i^{\star}$ is the target latent, generated for the same clip under the changed condition, which we call the counterfactual latent. The form fixes which entries of δ are free to vary across positions.

Any field decomposes into a shared vector and an offset. Let $\bar{\delta}$ be the average of δ_j over the M tokens:

$$\bar{\delta} \;=\; \frac{1}{M}\sum_{j=1}^{M} \delta_j, \qquad \delta_j \;=\; \bar{\delta} \;+\; (\delta_j - \bar{\delta}). \tag{6}$$

The shared vector $\bar{\delta}$ is added identically at every token; the offset $\delta_j - \bar{\delta}$ is what remains. The two parts are orthogonal, so their squared norms add to the field's. The optimised field is a causal intervention, not an estimate of the model's endogenous motion representation. Our experiments ask which intervention structure changes the generated motion and which internal components that effect depends on. Unless stated otherwise we fit each field on a fitting set of twenty-four clips, read it on one hundred held-out videos, and add it at the output of layer 2, where Section 4.3 identifies conditioning-related attention heads.

3.2.2 Finding the components a change uses

A *head* is one slice of a layer's per-token vector, written as its (ℓ, h) pair. To find which heads a generated change uses, we run the model while substituting the outputs of a chosen set S of heads with stored values v: a head in S takes v, every other head takes its original value o, so the head's

patched output $\tilde{o}_{\ell,h}$ is

$$\tilde{o}_{\ell,h} = \begin{cases} v_{\ell,h}, & (\ell,h) \in S, \\ o_{\ell,h}, & \text{otherwise}, \end{cases} \tag{7}$$

applied at every denoising step. We use two methods, differing in the value v they store:

Counterfactual replacement. We overwrite the chosen group's outputs with the values they produced under the changed condition, and read the change in the generated video. It tests whether replacing this group alone reproduces the requested change.

Pinning. We hold the chosen group at its un-intervened outputs while the token-wise field is applied, and read how much of the field's effect survives. It tests whether the field depends on the group.

Heads are then ranked by the change from counterfactual replacement, measured on the fitting set, and the leading group is taken. We call the heads in this group conditioning-related, because replacing their outputs with the values from the changed condition moves the generated video toward the requested change.

4 What structure of a fixed field enables control?

We study which parts of a fixed additive field produce the requested change. We compare a shared vector with fields that vary across token positions, then test which attention heads the effective field depends on.

4.1 A shared vector does not turn the ball

The attributes are linearly decodable. As in the probing work discussed in Section 2, the attributes of the models we train are linearly decodable from their activations. At layer 2, where we add our interventions, a linear probe reads the attributes with little error (the launch angle to $0.64°$ and the hue to $0.78°$).

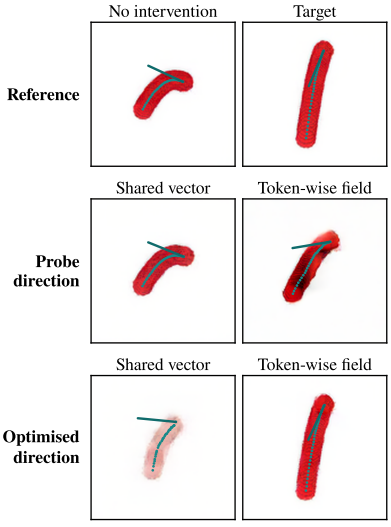


Figure 2: **Direction source and intervention form.** Each panel shows the same held-out clip under a requested $90°$ rotation.

No shared direction at one fixed site moves the trajectory. Figure 2 shows a probe direction and an optimised direction, each applied either as one vector shared across all tokens or as a token-wise field, for a requested rotation of $90°$. As a shared vector, the probe direction turns the trajectory by a median $2.1°$ over held-out clips and the optimised direction by $-1.9°$. As a token-wise field, the probe direction turns it by $35.9°$ and the optimised direction by $90.3°$, and only the latter reaches the request. A third direction, the difference between activations under the two conditions, behaves the same way (Figure 9). In the shared form none of the three moves the trajectory, changing the angle by at most $2.7°$ of the requested $90°$.

Changing the objective or the fitting set does not recover shared-vector control. The interventions above all minimise the distance to the counterfactual latent. To test whether that objective hides a useful shared direction, we also fit the shared vector to a second objective that asks only for the launch angle. The pixel readout has no gradient, so this objective reads the launch angle from the generated latent with a linear probe. Optimised this way, the shared vector turns the rendered trajectory by only $-2.5°$ to $+2.8°$, and the probe can report most of the requested rotation while the ball does not move. A search that uses no gradients, scoring candidate vectors on the rendered pixels, also finds nothing (Appendix C.2).

		Request	Shared vector ActAdd	ITI	REINS	Optimised	Rank-1 field TokenDial	Token-wise field (ours) Rank-16	Full rank	Control Random
BallDiT-24M	Angle	$+30°$	19.1	19.5	-1.5	$-5.5^{\dagger}$	29.5	97.6	**101.4**	1.3
		$+90°$	6.7	4.2	1.0	-2.1	$0.1^{\dagger}$	$82.5^{\dagger}$	**101.9**	1.0
		$+150°$	1.1	0.6	-0.2	$-2.1^{\dagger}$	$-23.8^{\dagger}$	99.1	**99.6**	$1.2^{\ddagger}$
	Speed	$\times 0.67$	59.6	$66.7^{\ddagger}$	38.4	22.3	78.6	98.7	**100.0**	0.7
		$\times 1.5$	59.2	81.0	25.0	29.5	86.7	100.8	**102.1**	-0.8
		$\times 2$	28.4	$70.4^{\ddagger}$	16.1	25.6	$75.9^{\ddagger}$	97.2	**97.6**	-0.3
	Hue	$+90°$	14.6	3.9	4.0	42.0	56.4	100.1	**99.2**	1.2
	Bright.	$0.8 \to 0.5$	$54.5^{\ddagger}$	$54.9^{\ddagger}$	22.6	59.6	$39.3^{\ddagger}$	$64.5^{\ddagger}$	**73.7**	-0.2
Wan2.1-1.3B	Angle	$+90°$	0.6	n/a	0.0	$14.3^{\dagger}$	52.5^{*}	94.9	**97.3**	0.1
	Speed	$\times 1.5$	0.5	n/a	0.0	70.2	71.3	100.0	**101.0**	-1.1
	Hue	$+90°$	32.5	n/a	8.5	48.6	97.0	100.7	**100.2**	1.2
	Bright.	$0.8 \to 0.5$	$122.1^{\S}$	n/a	$92.0^{\S}$	$100.4^{\S}$	$93.3^{\ddagger}$	99.7	**99.9**	0.7

† all balls fade. ‡ ball surface mottled. § ball turns black. $*$ trajectory collapses to a vertical fall.

Table 1: **Existing steering methods versus token-wise fields.** Each entry is the median change over held-out videos, as a percentage of the request. Every intervention in a row starts at the norm of that row's full-rank field; the published methods are then scaled by a factor chosen on the fitting set (Appendix C.4). For $\times 0.67$ and $\times 2$ we use only clips whose target speed stays within the speeds seen in training (1.0 to 3.5); the $+150°$ row is fitted for 1,000 iterations and the others for 300. n/a: Wan has no numeric condition for ITI to probe.

A single shared vector is fitted jointly across all clips, each asked for the same $90°$ rotation but starting from a different launch angle. It might therefore fail simply because one vector cannot serve all those starting points at once, rather than because it is the same at every position. To separate these two explanations, we fit a separate intervention on each clip. Even then the shared vector misses the target and fades the ball, while a token-wise field reaches it with the image intact.

On the held-out videos, the shared vector lowers the counterfactual-latent loss by degrading the ball rather than rotating the trajectory: it loses the ball in dozens of frames and desaturates the rest. The token-wise field reaches the request with no frame lost and a worst-frame saturation of 158, compared with 172 without intervention (Figure 2, bottom row).

Launch angle is the most sensitive to positional variation in the intervention. We next compare the four attributes under one protocol (Table 1). When we set the same norm for shared vectors and token-wise fields, the former reaches only -5.5% to -2.1% of the request across the three angles, while the latter meets it. On speed, hue and whole-frame brightness, the shared vector achieves roughly 20% to 60% of the request, but still falls short of the token-wise field. The shared vector thus fails almost completely only for launch angle, the attribute whose counterfactual change is least uniform across positions. At layer 2, the offset of this change holds 41–68% of its squared norm for launch angle, compared with 3–24% for the other three attributes (Table 19). The published methods in Table 1, ActAdd (Turner et al., 2023), ITI (Li et al., 2023), REINS (Kundu et al., 2026) and TokenDial (Liu et al., 2026), were shown to work on attributes that shift every sample the same way, such as sentiment, truthfulness, safety or the amount of motion; on launch angle in BallDiT-24M none of them reaches 30% of the request at any of the three requested angles.

4.2 What in a working field carries the effect

The offset reproduces almost all of the field-induced rotation. We take the successful token-wise field from Figure 2 and split it into its shared vector and offset using Equation (6), so neither part is obtained by fitting. On the held-out videos, the field rotates the trajectory by a median $90.25°$, the shared vector alone by $-0.46°$, and the offset alone by $89.91°$ (Figure 3a). The offset therefore carries almost all of the effect, and it does so on all three BallDiT-24M models trained with different seeds. None of the three interventions loses a frame. This split is not specific to layer 2. We fit a separate field at layers 1, 2, 4, 6, 8, and 10 and split each field into its shared vector and offset.

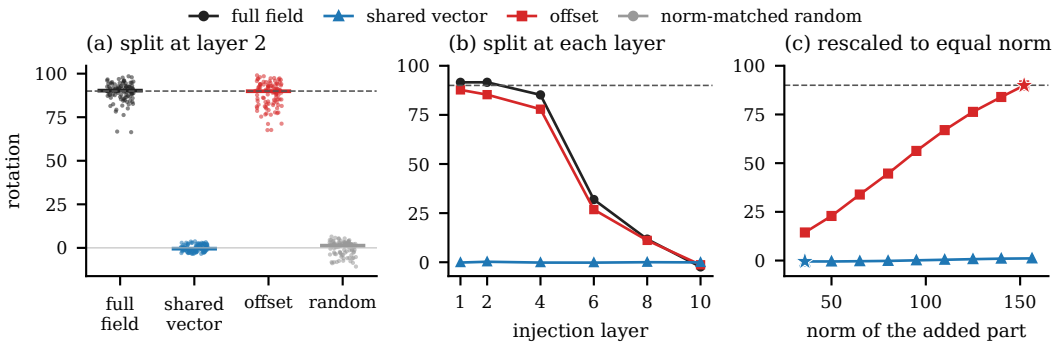


Figure 3: **The offset carries the rotation; the shared vector carries none.** (a) Rotation of each held-out clip when the full field, its shared vector, its offset, or a norm-matched random field is added; bars are medians. (b) A field fitted at each injection layer and split the same way. (c) The shared vector and the offset each rescaled to a common set of norms; stars mark their own norms.

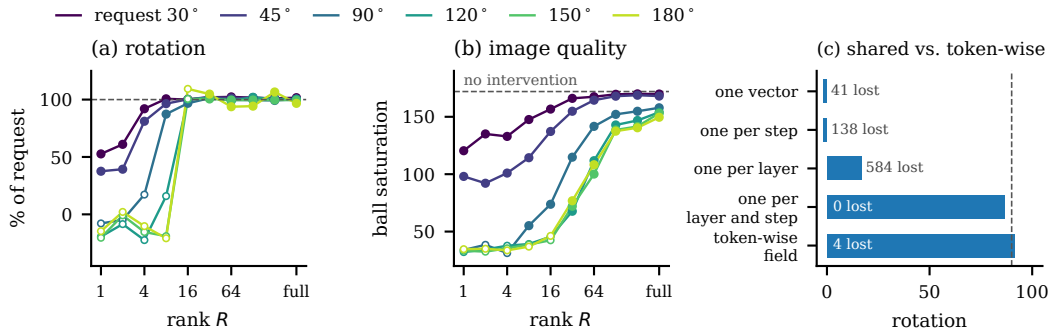


Figure 4: **How much positional freedom control needs on BallDiT-24M.** R is the number of distinct directions a field may use across token positions; each point is a median over held-out clips. (a) Rotation as a percentage of the request, one line per launch-angle request. (b) Saturation of the ball in the frame where it is most faded; the dashed line is the un-intervened value. Hollow markers lose the ball in 33 or more of 3,300 frames. (c) Rotation for a 90° request. The first four bars add one shared vector: fixed at one layer, varying per step, varying per layer, and varying per layer and step. The last bar is a token-wise field at one layer.

Across all six layers, the shared vector turns the trajectory by only $-0.20°$ to $0.24°$, while the full fields at layers 1, 2, and 4 still produce $85.19°$ to $91.59°$ of rotation (Figure 3b).

The shared vector's failure is not explained by its smaller size. When we rescale the shared vector up to the field's norm, it turns the trajectory only $1.1°$, but when we shrink the offset down to the shared vector's norm, it still turns it $14.4°$ (Figure 3c). Even with the norm cap raised eightfold, the optimised shared vector stops at 1.46 times the full-rank norm and still does not turn the trajectory. For launch angle, what the shared vector lacks is therefore more than size.

Low-rank token-wise fields can reproduce control. The split shows which part carries the effect, but we still do not know how many independent directions across positions are needed. Using the rank-R factorisation of Section 3.2, we vary R from one to full rank under a norm cap of 168.7, with the rest of the protocol fixed. At $R = 1$ the field fails to rotate the trajectory and destroys the ball in about a fifth of the frames. For the 90° request, from $R = 16$ the median rotation reaches the request with no frame lost, though image quality is still degraded: only 69 of the 100 videos meet the angle criterion and keep a complete ball, and worst-frame saturation stays well below the un-intervened value. Increasing R further mainly restores image quality rather than changing the rotation (Figure 4a,b). The norm matters as well: under the smaller norm of Table 1 (77.54), rank 16 reaches only 82.5% and fades the ball in every video (Appendix C.6).

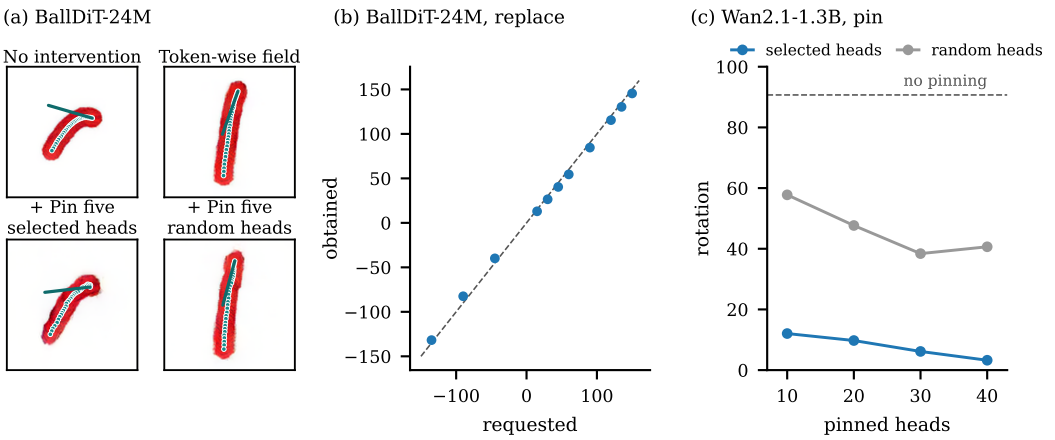


Figure 5: **The field acts through a small group of conditioning-related heads.** (a) BallDiT-24M, one held-out clip. The field alone turns the trajectory by a median $90.25°$; pinning the five selected heads reduces this to $14.66°$, while a layer-matched random five-head group leaves it between $77°$ and $109°$. (b) BallDiT-24M: with no field, the same five heads are replaced with their values from a run given each of 11 requested angles. (c) Wan2.1-1.3B: the field with the top 10 to 40 selected heads pinned, against the median of 30 random groups of each size.

A shared direction that varies over layers and steps. The experiments above all hold the intervention at one site. To test whether positional variation is itself what matters, we instead keep one direction shared across tokens and allow it to vary over layers and denoising steps. This reaches $86.34°$ of the request, using $230{,}400$ adjustable numbers against the token-wise field's $221{,}184$ (Figure 4c). A vector that is held fixed, varies only with step, or varies only with layer does not reach the goal. Sharing a direction across positions is therefore not by itself what limits control. In our fixed-site experiments, control is obtained by varying the intervention across positions; alternatively, a shared direction produces strong control when it varies jointly over layer and denoising step.

4.3 The field depends on conditioning-related components

So far the control has been located in the field's offset. We now identify which internal components the field acts through, using the counterfactual replacement and pinning we defined in Section 3.2.2.

The field's rotation drops only when the selected heads are pinned. The field here is fitted under one request: turn the trajectory by ninety degrees. We rank heads by counterfactual replacement on the fitting set and select the first five. Pinning them, the field's rotation falls from reaching the target to $14.66°$. Thirty random groups of five heads, drawn from the same layers as the selected five, keep the rotation between $77°$ and $109°$ (Figure 5a). Replacing the outputs of those same five heads with their values from a run given the target angle turns the trajectory by 87% to 98% of the request, across eleven requested angles from $-135°$ to $150°$ (Figure 5b).

No single head carries the trajectory, and this holds across formats and weights. We replace heads in five models we train: BallDiT-24M, a second seed, and three that receive the condition through cross-attention, numeric text or an English sentence. In all five, replacing one head turns the trajectory by at most $45.82°$, and replacing eight heads turns it by 73.19 to $87.46°$ (Table 21).

5 The same in a pretrained model

We repeat the study on Wan2.1-1.3B using simulator videos, each inverted to the noise that regenerates it. We test two interventions: an optimised token-wise field (Table 1) and the activation change between two natural-language prompts (Figure 7).

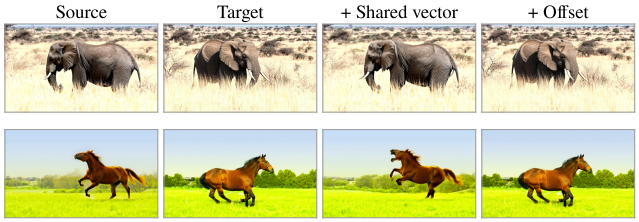


Figure 7: **Splitting the activation change between two prompts into a shared vector and an offset.** Each row is a prompt pair that differs only in facing direction. The last two columns add the shared vector or the offset of the Target–Source change to Source; only the offset flips the facing.

The offset reproduces the rotation. A token-wise field reaches the requested $90°$ rotation on held-out videos: its offset alone produces $86.9°$ and its shared vector $0.1°$. As on BallDiT-24M, the rotation needs a lower rank than the image. $R = 4$ is the smallest rank meeting the criterion, and there the field turns the trajectory by 96.5% of the request, but the ball's saturation in its most faded frame falls to 72.5% of its un-intervened value; from $R = 16$ it stays between 92.9% and 99.2% (Figure 6).

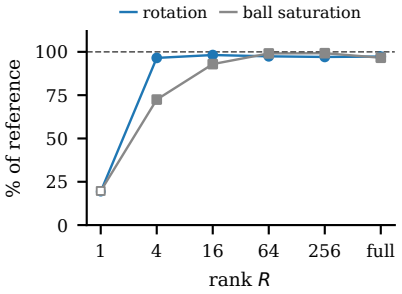


Figure 6: **How much positional freedom control needs on Wan2.1-1.3B.** Fields are fitted for a $90°$ launch-angle request; each point is a median over held-out clips.

The field depends on a selected group of heads. No single head carries the trajectory on its own, the best one reaching at most $1.84°$. Holding ten of the model's 360 heads at their un-intervened outputs while the field is applied drops the rotation from $90.70°$ to $12.05°$. Thirty random groups of ten heads, drawn from the same layers, leave a median rotation of $57.77°$, and the smallest value among them is $15.45°$ (Figure 5c).

The same split holds on natural-language prompts. At each denoising step we split the Target–Source activation change into its shared vector and offset and add one half to the source run (Figure 7). We use five prompt pairs whose two prompts generate a source and a target that face opposite ways, for example an elephant facing left or right. On all five the offset flips the subject's facing, and the shared vector flips none.

6 Conclusion

We study the intervention structure and internal components behind motion control in the tested models. At one fixed injection site, a vector shared across positions barely rotates the trajectory, even when it is the shared vector of an intervention that already works. The offset from that same intervention reproduces almost all of the rotation, and this difference remains after norm matching. Full positional freedom is not required. The rotation reaches the request at a low rank, and image quality needs a higher rank. Fitted fields transfer to held-out videos. The field-induced rotation depends on a small group of conditioning-related attention heads. Pinning them strongly reduces the effect, whereas layer-matched random groups do not. The same pattern holds on Wan2.1-1.3B for synthetic clips inverted into the model. Under natural-language prompts we test only the split, and the offset again flips the subject's facing while the shared vector does not.

Two limitations remain. First, the fitted field works only at the positions in the frame where it was fitted. When we shift a clip so that the ball appears elsewhere in the frame, and shift the field by the same amount, the rotation falls from about $90°$ to $43°$. The field therefore acts on fixed positions in the frame rather than as a rotation that moves with the object. Second, on Wan2.1-1.3B we test the fixed field only on reconstructed clips and on held-out seeds of one prompt, not on new scenes.

REPRODUCIBILITY STATEMENT

The attribute changes we report in the generated video are read from generated pixels against simulator ground truth, and the readout is specified in Appendix A. The synthetic model, its training data generator, the intervention hooks, the optimisation objectives and the per-attribute readouts will be released with the paper. The pretrained-model experiments use the public Wan2.1-T2V-1.3B checkpoint at the layers and step counts stated in Table 3, the Notation paragraph of Appendix B, and Appendix E. The stored token-wise fields used for the shared-vector/offset split and for the pinning experiment are released as well.

THE USE OF LARGE LANGUAGE MODELS

Large language models were used as a general-purpose writing and coding assistant: drafting and editing prose, writing plotting and analysis scripts, and searching for related work. All experiments, measurements and numbers reported in this paper were produced by the authors' own code and checked against the raw outputs. No part of the scientific contribution was delegated to a language model: not the research question, not the experimental design, not the interventions, and not the interpretation of results.

REFERENCES

Nahid Alam. Causal physics steering in video world models via concept activation vectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2026. arXiv:2605.24322.

Kyobin Choo, Youngmin Kim, Hyunkyung Han, Geunrip Park, Chanyoung Kim, Sunyoung Jung, and Seong Jae Hwang. QWERTY: Training-free motion control via query-warped video diffusion transformers. In *European Conference on Computer Vision (ECCV)*, 2026. arXiv:2607.01869.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Parsa Esmati, Somjit Nath, Katja Hofmann, Derek Nowrouzezahrai, Samira Ebrahimi Kahou, and Majid Mirmehdi. The invisible hand of physics: When video diffusion models know more than they show. *arXiv preprint arXiv:2606.05328*, 2026.

René Haas, Inbar Huberman-Spiegelglas, Rotem Mulayoff, Stella Graßhof, Sami S. Brandt, and Tomer Michaeli. Discovering interpretable directions in the semantic latent space of diffusion models. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2024. arXiv:2303.11073.

Jihoon Hong, Alice Chan, Qiyue Dai, Julian Skifstad, and Glen Chou. Activation steering of video generation models via reduced-order linear optimal control. *arXiv preprint arXiv:2606.04775*, 2026.

Zehao Jin, Ruixuan Deng, Junran Wang, Xinjie Shen, and Chao Zhang. Beyond steering vector: Flow-based activation steering for inference-time intervention. *arXiv preprint arXiv:2605.05892*, 2026.

Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Richards, and Mike Rabbat. Interpreting physics in video world models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.07050.

Sunyoung Jung, Jiwoo Park, Yoonseok Choi, Kyobin Choo, Ming-Hsuan Yang, and Seong Jae Hwang. Controlling motion transfer in diffusion transformers via attention heads. In *European Conference on Computer Vision (ECCV)*, 2026. arXiv:2607.11081.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2411.02385.

Rohit Kundu, Arindam Dutta, Sarosij Bose, Athula Balachandran, and Amit K. Roy-Chowdhury. Pulling the REINS: Training-free safety alignment of video diffusion models via representation steering. *arXiv preprint arXiv:2606.17257*, 2026.

Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.10960.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.03341.

Zhixuan Liu, Peter Schaldenbrand, Yijun Li, Long Mai, Aniruddha Mahapatra, Cusuh Ham, Jean Oh, and Jui-Hsien Wang. TokenDial: Continuous attribute control for text-to-video generation in visual dial space. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2026. arXiv:2603.27520.

Aayush Mishra, Daniel Khashabi, and Anqi Liu. Steered LLM activations are non-surjective. In *ICLR 2026 Workshop on Scientific Methods for Understanding Deep Learning (Sci4DL)*, 2026. arXiv:2604.09839.

Aryan Sharma, Cutter Dawes, and Shivam Raval. Dissociating decodability and causal use in bracket-sequence transformers. *arXiv preprint arXiv:2604.22128*, 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Binxu Wang, Jingxuan Fan, and Xu Pan. Circuit mechanisms for spatial relation generation in diffusion transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2601.06338.

Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. In *International Conference on Machine Learning (ICML)*, 2024a. arXiv:2402.01566.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024b. arXiv:2312.03641.

Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.14864.

Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. DragNUWA: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.16042.

Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2407.21705.

CONTENTS OF THE APPENDIX

A Measurement protocol **12**

A.1 Pixel-level attribute readout 13

A.2 Image-quality metrics 13

A.3 Success criterion 13

A.4 Robustness of the criterion 13

B Model, training, and interventions **13**

B.1 Model and training 13

B.2 The interventions 14

C The token-wise field **15**

C.1 Orthogonal decomposition of the field 15

C.2 Control comparisons and image quality 16

C.3 Intervals for the main table 21

C.4 Scale sweep for the published methods 21

C.5 The shared-vector result across injection layers 22

C.6 Rank and intervention structure 22

C.7 Scope across attributes 25

D Conditioning-related components **26**

D.1 Conditioning-induced rotation is concentrated in early layers 27

D.2 A small head group reproduces most conditioning-induced rotation 27

D.3 The field depends on the selected heads 28

E Generalisation to a pretrained model **30**

E.1 Token-wise fields control Wan2.1-1.3B 30

E.2 The asymmetry persists across targets and natural prompts 31

E.3 Wan fields share structure but do not reliably transfer 33

E.4 The head-level dependence generalises to Wan 35

A MEASUREMENT PROTOCOL

The measurement protocol is fixed: it measures each attribute from the generated frames, judges whether an intervention has damaged the image, and decides when a request is met. All 300 clips are drawn once, with speed, launch angle and radius sampled independently and uniformly. Fields are fitted on clips 0 to 23 and read on clips 28 to 127, and each clip has its own starting noise, drawn once and reused for that clip's un-intervened run, its target and every intervention on it. For the $\times 0.67$ and $\times 2$ rows of Table 1, the 24 fitting and 100 held-out clips are instead drawn from all 300, keeping only clips whose target speed stays within the training range of 1.0 to 3.5; the two sets do not overlap.

A.1 PIXEL-LEVEL ATTRIBUTE READOUT

Every attribute reported in this paper is measured from the generated frames. Per frame, a pixel is part of the ball if $\max_c I_c - \min_c I_c > 40$ (on a 0–255 scale); we take the largest connected component of that mask and its centroid, converted to world coordinates. A frame whose mask has fewer than 20 pixels is recorded as *lost* and excluded. Launch angle and initial speed come from a least-squares fit of the resulting track against a projectile with the gravity used to build the data. With t the frame time, we solve

$$x(t) = x_0 + v_x t, \qquad y(t) - \tfrac{1}{2} g t^2 = y_0 + v_y t,$$

where (x_0, y_0) is the initial position, (v_x, v_y) the initial velocity and g the simulator's gravity, and report $\mathrm{atan2}(v_y, v_x)$ and $\|(v_x, v_y)\|$. Because BallDiT-24M is trained on data with a known gravity, this fit treats the gravity as given. The same fit reads the Wan2.1-1.3B field fitted on all fitting clips (Table 1, Figure 6 and Appendix E.1), whose clips are simulator videos inverted into the model. For the matched clip pairs of Appendix E.2, where Wan renders a new clip whose ball need not obey that gravity, we instead read the launch direction from the derivative at $t = 0$ of a free quadratic fit to $x(t)$ and $y(t)$ over the first twelve frames. Hue is the circular mean of the ball pixels' hue. On batches that change colour the ball is found by its distance from the white background, $255 - \min_c I_c > 40$, rather than by $\max_c I_c - \min_c I_c$, which loses a ball that has been turned grey.

A.2 IMAGE-QUALITY METRICS

Image quality is reported as three numbers: ball pixels, the number of lost frames, and the saturation of the *worst* frame. Here a frame's saturation is the mean over the ball's pixels of $\max_c I_c - \min_c I_c$, the same quantity used to find the ball above, on the 0–255 channel scale. We take the worst frame because fading typically ruins only the first or last few frames and a median hides it.

A.3 SUCCESS CRITERION

Every video yields an angle: the fit uses whatever frames survive the twenty-pixel rule, and no video is dropped. For each video, we first compute the obtained rotation as the change in read launch angle relative to its un-intervened run. Its distance to the target is the absolute difference between this obtained rotation and the requested rotation, and the median distance is the median of these distances over the videos. A request counts as met when the median distance to the target is within a tenth of the requested rotation and fewer than 33 of the 3,300 frames are lost; an intervention that loses 33 or more frames is drawn with a hollow marker and does not count as meeting a request, though its angle is still listed. Intervals on a median are bootstrap intervals resampled over videos, not over frames: 10,000 resamples, reported as the 2.5th and 97.5th percentiles.

A.4 ROBUSTNESS OF THE CRITERION

The norm cap comes from the unconstrained full-rank run on the fitting set, and the injection layer from Section 4.3. The criterion was chosen after inspecting the 100 evaluation clips, which are held out from field fitting but not from criterion selection. We therefore test its sensitivity on a further independent set. Varying the lost-frame allowance between 16 and 66, or widening the angular tolerance from a tenth of the request to a fifth, does not change the smallest rank that meets the criterion for nearly all requests. A stricter tolerance of a twentieth shifts some borderline requests; on the independent set, eight of the nine requests keep the same smallest rank.

Lost frames alone do not reveal every failure; Figure 8 shows two. In one the ball loses its trajectory: it shrinks to a blob near the start and the arc is gone. In the second the ball still follows the arc, but coloured speckle grows around the ball and the ball shrinks.

B MODEL, TRAINING, AND INTERVENTIONS

B.1 MODEL AND TRAINING

Training BallDiT-24M. BallDiT-24M is a 12-layer diffusion transformer (hidden width 384; six heads of dimension 64 per layer; feed-forward ratio 4.0; 23.8M parameters), trained from scratch

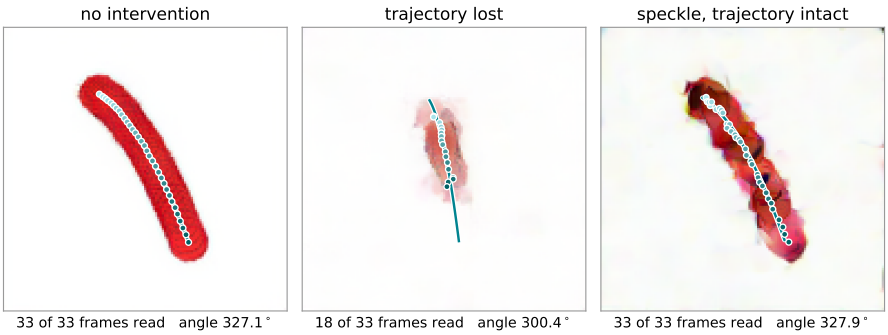


Figure 8: **What the pixel measurement sees.** One held-out clip with no intervention, and under two interventions that break the image in different ways. Each panel collapses the 33 frames by taking the minimum over time. Dots are the ball centre read from each frame, light to dark with time; the line is the projectile track fitted to them. The middle panel reads the ball in only 18 of 33 frames and the fitted track no longer follows the arc. The right panel keeps every frame and reads an angle within $0.8°$ of no intervention, while the ball is visibly speckled.

for 30,000 steps following Kang et al. (2025). We use AdamW ($\beta = (0.9, 0.999)$, weight decay 0.01, no warm-up), a learning rate of 10^{-4} under cosine decay, a global batch of 256, an exponential moving average of 0.9999, and seed 150. BallDiT-24M denoises in latent space rather than in pixels: each $33 \times 128 \times 128$ clip is encoded by the VAE of Wan2.1-T2V-1.3B into a latent that flattens to the $(576, 384)$ tokens of Table 2, and pixels are decoded back only to compute pixel-based metrics.

model	12-layer diffusion transformer, hidden width 384, 23.8M parameters, trained from scratch
data	30k synthetic videos: one ball on a white background following a projectile trajectory
video	33 frames at 128×128
conditioning	launch angle, initial speed, hue; three scalars, adaptive layer normalisation
tokens per layer	$(576, 384)$: 9 temporal slices $\times$ 8×8 positions; slice 0 carries frame 0, slice k frames $4k - 3$ to $4k$
world units	frame 10 across, ball diameter 1.4

Table 2: Setup. Distances are quoted in the simulator's own units throughout; batches that draw fresh conditions say so where they are used.

Notation. A clip occupies $X \in \mathbb{R}^{576 \times 384}$ at the output of each of the 12 blocks, and generation runs for K steps of an explicit Euler integration of the flow-matching velocity field with σ falling linearly from 1 to 0. K is 50 for every BallDiT-24M intervention, and 100 for ranking its heads by counterfactual replacement. On Wan2.1-1.3B, K is 50 for the field fitted on inverted simulator clips (Appendix E.1), the red-to-blue prompt pairs and the facing experiment of Figure 7, and 20 for the 28 matched clip pairs, the head ranking and the transfer to unseen seeds. Where a parameter count is per denoising step, it is per step of that batch's K. We write $X^{(k,\ell)}$ for the tensor at step k and layer ℓ. A *counterfactual pair* is two generations from the same starting noise whose conditioning differs in exactly one scalar; we call them the Source and the Target. Some results below use a fixed *six-clip batch*, reported as a median over the six; the main results use the held-out videos.

B.2 The interventions

This subsection states how the interventions are computed and when one does not count as control.

Linear probes. Activations are collected from real videos with noise added at the same σ schedule. From 5000 separate probe samples we fit on 2500, choose the ridge coefficient from $\{10^{-2}, \ldots, 10^{5}\}$ on a further 1000, and report on the remaining 1500, which is used once. At layer 2, with activations averaged over tokens at $\sigma = 0.5$, the mean circular error on these 1500 clips is $0.64°$ for the launch angle and $0.78°$ for the hue. Ridge is solved in sample space: with $F \in \mathbb{R}^{n \times D}$ the centred training features, Y the centred training targets, $\bar{f}$ and $\bar{y}$ their means and α the ridge

weights	parameters	how the condition arrives	K	used for
BallDiT-24M	23.8M	three scalars, adaLN	50, 100	everywhere
BallDiT-24M, second seed	23.8M	three scalars, adaLN	50	the split, rank sweep, head ranking, pinning
BallDiT-24M, third seed	23.8M	three scalars, adaLN	50	the split, rank sweep
BallDiT-24M, numeric text	31.3M	three scalars as numeric text, Wan's encoder	50	head ranking
BallDiT-24M, brightness	23.8M	three scalars, adaLN, brightness in place of hue	50	Table 1, Appendix C.7
BallDiT-24M, cross-attention	29.8M	three scalars, cross-attention	50	head ranking, pinning
BallDiT-24M, sentence	31.3M	an English sentence, Wan's text encoder	50	head ranking
model-M	94.9M	three scalars, adaLN	50	the split (Appendix C)
Wan2.1-1.3B	1.3B	text prompt (full prompt or one-word change)	20, 50	Table 1, Appendix E
BallDiT-24M, two-ball world	24.0M	four scalars (angle and speed per ball), adaLN, two-ball clips	50	Appendix C.7

Table 3: Every set of weights used in this paper (adaLN: adaptive layer normalisation). The K column lists every number of denoising steps that row is run at, and Appendix B says which batch uses which. Wan2.1-1.3B is the one we did not train; the rest we trained ourselves.

coefficient, the prediction for an activation $q \in \mathbb{R}^D$ is $(q - \bar{f})F^\top(FF^\top + \alpha I_n)^{-1}Y + \bar{y}$. Circular attributes are regressed as a $(\cos, \sin)$ pair and read back with $\mathrm{atan2}$. When we steer along a probe direction, we add the *shortest* vector that makes the probe read the target value, so the readout is at the target by construction. These probes are fitted on real videos with noise added, while the interventions run on the model's own generation.

Token-wise probes. A separate probe is fitted at each of the 576 token positions, with the attribute regressed at each position as a $(\cos, \sin)$ pair and the readout taken as the mean over positions. Steering along it adds a different vector at each token, so the change affects the offset as well as the shared vector.

Optimised fields. A tensor δ is added to one block's output at every generation step: as a shared vector $\delta \in \mathbb{R}^{1\times 384}$, added to all 576 tokens; as a token-wise field $\delta \in \mathbb{R}^{576\times 384}$, one vector per token. It is optimised with Adam through the full generation, learning rate 3×10^{-3} with cosine decay, 300 iterations (1,000 for the $+150°$ row of Table 1), with gradient checkpointing. In the rank sweep of Section 4.2, the full-rank field is optimised as $\delta = AV$ with $R = 576$; the field it reaches has rank 384, the largest a $(576, 384)$ array can have. Two objectives appear. The *latent* objective is the mean squared error to the latent of the true counterfactual clip. The *trajectory* objective targets the launch angle alone: the pixel measurement has no gradient at its colour threshold, so this objective reads the launch angle from a linear probe on the latents, fitted on generated videos against pixel-measured ball centres. This probe target can be satisfied while the ball stays where it started (Appendix C.2), so all reported attribute values are measured from the pixels.

Shared vector and offset. The split of Equation 6 is an identity at the injection point; later nonlinear layers can turn a uniform shift into one that varies across tokens. Our shared vector is the average over token positions within a single intervention.

Copying activations from the target run into the source run (counterfactual replacement, Section 3.2.2) is what Zhang & Nanda (2024) call denoising patching.

C THE TOKEN-WISE FIELD

This appendix holds the proof that the split of Equation (6) is orthogonal, the controls and measurements behind Section 4, the intervals and scale sweep for Table 1, and the rank and attribute analyses.

C.1 ORTHOGONAL DECOMPOSITION OF THE FIELD

This section proves the identity used in Section 3.2.1: the split into a shared vector and an offset is an orthogonal decomposition.

The split is an orthogonal projection (proof of Equation 6). Let $\mathcal{S} = \{\mathbf{1}v^\top : v \in \mathbb{R}^D\}$, where $\mathbf{1} \in \mathbb{R}^M$ is the all-ones vector. For any $\delta \in \mathbb{R}^{M\times D}$ and any v, $\langle \delta - \mathbf{1}\bar{\delta}^\top, \mathbf{1}v^\top\rangle_F = \sum_j (\delta_j - \bar{\delta})^\top v = \big(\sum_j \delta_j - M\bar{\delta}\big)^\top v = 0$, so $\delta - \mathbf{1}\bar{\delta}^\top$ is orthogonal to every element of $\mathcal{S}$. Hence $\mathbf{1}\bar{\delta}^\top$ is the orthogonal

projection of δ onto $\mathcal{S}$, the unique minimiser of $\|\delta - s\|_F$ over $s \in \mathcal{S}$. Since $\|\mathbf{1}\bar{\delta}^\top\|_F^2 = M\|\bar{\delta}\|_2^2$, Pythagoras gives

$$\|\delta\|_F^2 \;=\; M\,\|\bar{\delta}\|_2^2 \;+\; \|\delta - \mathbf{1}\bar{\delta}^\top\|_F^2. \tag{8}$$

The split is unique and energy-additive; the effects of its two parts are tested in Section 4.2.

C.2 Control comparisons and image quality

This section holds the controls behind Section 4. Each asks whether the shared vector's failure has a simpler cause than the form itself: too little norm, the wrong optimiser or objective, one vector serving many clips, a metric fooled by a damaged image, the wrong injection site, and positional structure the shared vector might pick up in later layers. The split on held-out videos that the body cites opens the section.

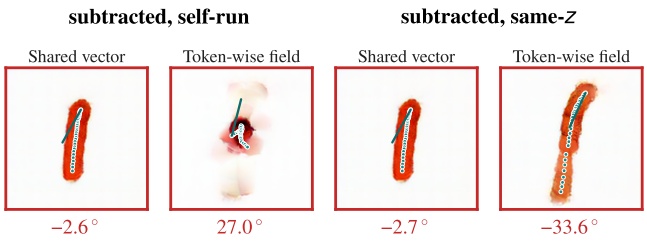


Figure 9: **The activation-difference direction.** The difference between target and source activations, taken in two ways and added as a shared vector or as a token-wise field, on one held-out clip. Self-run: each run follows its own denoising trajectory and the difference is taken step by step. Same-z: one trajectory is kept and the same state is fed to both conditions at every step. Numbers are median rotations over the held-out videos at a requested 90° (Table 7). The self-run subtraction applied as a token-wise field turns the trajectory 27° but breaks the ball.

The split on held-out videos. We test whether the split still holds on the held-out conditions. We split the same stored field again and inject its shared vector and its offset separately, re-optimising nothing (Table 4).

	rotation (°)			ball	worst-frame
addition	median	interquartile	bootstrap 95%	pixels	saturation
baseline	—	—	—	262	172
full	90.25	87.26–92.70	[89.52, 91.32]	262	158
shared vector	−0.46	−1.78–1.70	[−1.18, 0.43]	264	183
offset	89.91	82.94–93.55	[87.75, 91.09]	258	144
random	1.28	−2.73–2.42	[0.60, 1.88]	272	113

Table 4: **Shared vector and offset on held-out videos.** The fitted token-wise field is split using Equation (6); each part, the full field, and a norm-matched random field are added to the held-out videos with a target rotation of 90°. Every intervention keeps the ball in all 3,300 frames (Appendix A).

The shared-vector and offset intervals do not overlap. No intervention loses a frame, and the shared vector keeps the worst-frame saturation at 183 and the offset at 144, against 172 with no intervention.

The shared vector at the field's own norm. The shared vector is small, so its failure could be one of size. Its share of the field's squared norm runs 2.2% to 7.7% across the six launch-angle requests and 2.5% to 3.5% across the three hue requests, which puts it 3.6 to 6.8 times below the field's own norm. Scaled up to that norm and injected on the same held-out videos, the launch-angle shared vectors turn the ball 0.7, 0.2, 1.1, 0.2, 0.4 and 1.2° at requests of 30 to 180°, with no frame lost and a worst-frame saturation of 141.8 to 182.0 against 172 with no intervention. Matching the field norm does not make the shared vector effective for launch-angle control. At a requested 90° the reverse also holds: shrinking the offset to the shared vector's norm still turns the trajectory a median 14.4°

$[12.96, 16.87]$ with no frame lost, above the $1.1°$ the shared vector reaches at over four times that norm, so the asymmetry persists at matched energy. Optimising the shared vector directly with its norm cap raised to two, four and eight times the norm of the full-rank field in the $+90°$ launch-angle row of Table 1 (77.54) gives the same solution under all three caps: it stops at norm 113.0, reaches -2.2% of the launch-angle request, and the ball fades on every held-out clip. The same relaxation brings hue to 86.1% and speed to 103.3% of their requests. For hue, norm is part of the gap: scaled up the same way, the shared vector turns $18.4°$ of a requested $30°$, $23.1°$ of a requested $90°$ and $7.4°$ of a requested $180°$, still short of every request but well past the 0.9 to $4.2°$ it reaches at its natural norm.

The projection comparison against direct optimisation at a fixed norm. Six of the seven parameterisations were also optimised directly against the counterfactual latent, the same objective that produces the working token-wise field, with the field projected back into a ball of Frobenius norm 168.704 after every update. That is the norm the unconstrained optimisation settles on when the field is fitted on the fitting set. The cap applies to the full $(576, 384)$ array, so a shared vector at the cap has, at each token, a norm of $168.704/24 = 7.03$.

The results are in the last two columns of Table 18. Three of the six directly-optimised interventions lose 41 to 701 frames of 3,300 and are listed without counting as control: they lower the loss by degrading the ball rather than by reaching the target latent. Of the three that keep the ball, rank-8 turns the trajectory $78.62°$, one vector per spatial position (64 directions) $70.44°$ and the token-wise field $91.59°$; only the token-wise field reaches the request.

The objective falls while the pixels stand still. Every shared-form intervention is driven by a differentiable objective, so a failure could be the optimiser stopping early rather than the form falling short. Figure 10 follows one shared vector and one token-wise field through 300 Adam steps against the counterfactual latent, reading the launch angle from the pixels every 20 steps. The shared vector's objective falls monotonically from 0.085 to 0.056 and has flattened by step 200, while the median rotation over the six clips stays between -3.5 and $0°$; the ball fades instead, and 34 of 198 frames are gone by the end. Neither intervention in this figure is capped, so these two frame counts are not comparable with the capped ones elsewhere. The token-wise field's objective falls to 0.008 and its rotation rises with it to a median $87.9°$ (80.0 to $91.5°$ over the six clips), with 8 frames lost. On the shared form the objective stops decreasing by step 200, and the solution the optimiser reaches moves the latent toward the target without turning the trajectory; that this run converges does not rule out a better shared vector under a different objective or optimiser.

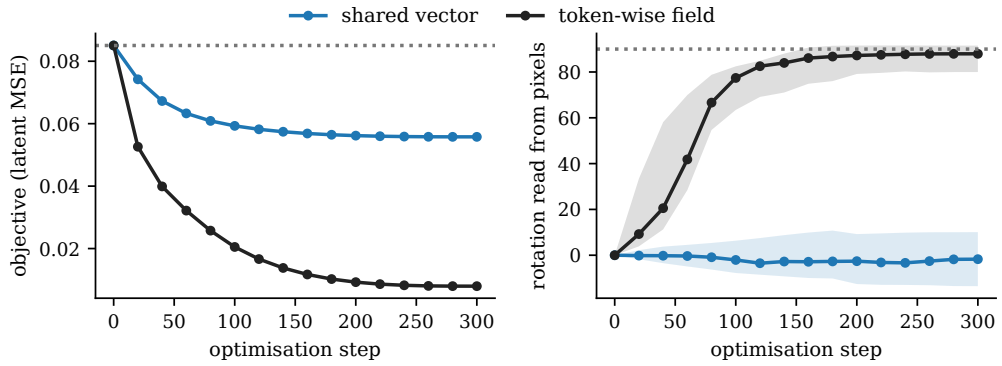


Figure 10: Objective and pixel readout along the optimisation, six clips, a requested $90°$. Left: the differentiable objective, the mean squared distance of the generated latent from the counterfactual latent (dotted: no intervention). Right: the launch-angle rotation read from the generated frames, median with the range over the six clips (dotted: the request). No norm cap in either intervention.

Alternative optimisers and objectives give the same qualitative result. A gradient-free evolution strategy provides an optimiser-independent check, searching the shared vector at layer 2 directly on the pixel readout. Started from zero, no candidate in 120 generations beats zero: every change hurts the image first. Started from the gradient solution, the objective falls only by repairing the

image, and the trajectory stays 72 to 101° from the target. Optimising directly against the launch-angle objective gives the same qualitative result: the shared vector keeps the image intact but moves the six clips only -2.5 to $+2.8^\circ$, with inconsistent sign. The probe can be satisfied while the ball stays put: scaled up, the probe reads 82.5° of a requested 90° while the pixels move under 1°. A second probe refitted on fresh clips over-reports rotations that did not happen and under-reports one that did, so every reading in this paper is taken from the pixels. Frozen and injected on a fresh set of held-out conditions, the best shared vector still moves the trajectory only 0.67°.

One shared vector per clip. (Table 5) The shared vector of Table 7 is one vector for every clip it is fitted on. Optimising a separate shared vector for each of six clips, against that clip's counterfactual latent at layer 2, and a separate token-wise field for each clip the same way, gives the results below; a requested 90° throughout.

form	to target ($^\circ$)	lost of 33	ball px	worst sat.
shared	19.34 to 58.77	0 to 25	0 to 253	33.9 to 113.8
token-wise	0.01 to 0.18	0	254 to 274	150.9 to 173.0

Table 5: One intervention optimised per clip, six clips, a requested 90°. "to target" is the distance of the obtained launch angle from the target; "ball px" the median ball pixel count over frames (un-intervened 254 to 274) and "worst sat." the lowest per-frame ball saturation. The per-clip shared vectors turn the ball 31 to 71°, none of them reaching the target, and they fade it: on one clip the ball is gone from 25 of the 33 frames and no ball pixels remain. The per-clip fields reach the target on every clip with the image intact.

The token-wise field changes the ball, not the background. We measure how much of the token-wise field's improvement falls on the ball rather than the background. With no intervention the ball region differs from the target by 93.78 on a 0 to 255 scale, and the background by 0.09. With the token-wise field the ball region falls to 28.39 while the background stays at 0.12. The shared vector reduces the ball-region error only to 89.70 and is the only form to disturb the background (0.40). The ball region, the pixels where either the generated or the target clip shows the ball, is 1.7 to 2.8% of the pixels, and the whole improvement falls on it.

Degradation alone does not move the reading. (Table 6) The interventions that rotate the trajectory also lower the saturation, so we checked whether image degradation alone can change the measured launch angle. Each un-intervened held-out video is degraded in turn by seven operations, with no intervention applied, and read with the same launch-angle fit. Three of the seven are matched frame by frame to an earlier token-wise field on these videos (median rotation 72.83°, 55 frames lost): blending toward white until the ball's saturation in each frame matched it, eroding the ball until its pixel count matched, or both. Whitening blanks the 48 frames in which that field leaves no measurable ball colour; eroding matches the field's pixel count, so it loses the ball wherever the field does. The other four go past it: blanking the first and last frame, shrinking the first and last three frames below the measurement's 20-pixel floor, greying the end frames to a saturation of 9, and smearing the ball along a fixed random direction with a length that grows toward the ends of the clip, so that the centroid shift varies in time. Table 6 gives, over the held-out videos, the range of the change in the read angle, the number of videos whose angle moves by more than 1°, and the frames lost; the token-wise field to which the first three are matched, on the same videos, is listed for comparison.

Sweeping the probe direction's amplitude. Table 8 sweeps both probe-direction interventions over strength on the held-out videos. Pushing the shared probe vector further, on the six-clip batch at layer 2 against a requested 90°: at $\times 5$ the ball is fading, 12 frames of 198 are gone and the per-clip values no longer agree in sign; from $\times 10$ on the image is destroyed, with 102 frames lost at $\times 10$ and every frame a coloured lattice at $\times 50$. A random direction of the same length at $\times 50$ moves the reading 46.54° against the probe direction's 10.72°, so at that amplitude the movement is not a property of the direction. Increasing the amplitude does not produce controlled rotation before the image degrades.

degradation	change in read angle (°)	videos moving more than 1°	lost frames of 3,300
whitening	-0.22 to $+0.23$	0	50
ball shrinking	-0.21 to $+0.17$	0	61
both	-0.17 to $+0.23$	0	65
blank end frames	-2.01 to $+1.80$	6	202
shrink end frames	-0.29 to $+0.83$	0	600
grey end frames	-0.82 to $+0.76$	0	200
directional smear	-0.07 to $+0.08$	0	0
token-wise field (matched)	13.9 to 101.8 (median 72.83)	100	55

Table 6: Seven image degradations applied to the held-out videos without any intervention, against the token-wise field to which the first three degradations are matched, on the same videos.

form	direction	turn (°)	ball px	lost
shared vector	probe	2.1	254	0
	optimised	-1.9	192	41
	subtracted, self-run	-2.6	288	0
	subtracted, same-z	-2.7	287	0
token-wise field	probe	35.9	256	0
	optimised	**90.3**	262	0
	subtracted, self-run	27.0	312	0
	subtracted, same-z	-33.6	268	0

Table 7: The direction source crossed with the intervention form: launch angle, layer 2, a requested $+90°$. *form* is one vector added to all 576 tokens or one vector per token; *direction* is read from a probe, found by optimisation, or taken as the difference between two sets of activations. Generation runs for fifty steps and the two runs diverge after the first, so the difference is taken in two ways: *self-run* lets each run follow its own trajectory and subtracts step by step; *same-z* stays on one trajectory and feeds the same state to both conditions at every step. Either way the per-step differences are averaged into one array, which is then scaled to the norm of the working field (156.3) — at their own norm, 15 to 22 times smaller, the token-wise subtracted fields move the trajectory by a median of less than 3° ($-2.6°$ self-run, $-2.3°$ same-z). *turn* is the median rotation obtained, *ball px* the median ball pixel count (262 with no intervention) and *lost* the frames of 3,300 that lose the ball. Figure 2 shows the probe and optimised rows as pictures.

form	strength	turn (°)	lost	worst sat.
token-wise field	$\times 0.5$	14.3	0	161.7
	$\times 0.6$	18.1	0	155.1
	$\times 0.8$	26.2	0	134.7
	$\times 1$	35.9	0	113.7
	$\times 1.5$	58.6	13	63.7
	$\times 2$	75.1	101	25.8
shared vector	$\times 0.5$	1.0	0	—
	$\times 1$	2.1	0	—
	$\times 2$	4.5	0	—
random	$\times 0.5$–$\times 2$	-0.3 to 0.0	0	—
condition changed	—	90.2	0	—

Table 8: Both probe-direction interventions swept over strength, on the held-out videos at a requested $+90°$. Strength $\times 1$ makes the probe read exactly the target (the probe is linear). *turn* is the median rotation obtained, *lost* the frames of 3,300 that lose the ball, *worst sat.* the median worst-frame ball saturation (172.3 with no intervention).

A sweep of shared vectors in the plane of the shared vector and the probe direction. The plane spanned by a successful field's shared vector and the probe's shortest vector (cosine 0.09 between the two) is swept at 24 angles and 7 norms from 5 to 150, 168 shared vectors, each read on twelve held-out videos. At norm 5 the ball is already pink and speckled and the largest rotation is 10.95°;

at norm 10 the ball breaks up, keeping about half its pixels, at $17.41°$; from norm 20 on the frames are a woven texture with no ball, and the $30°$ to $92°$ the pixel measurement then reports are read off the texture. Within this plane no shared vector that keeps the ball's colour intact turns it $10°$.

The findings hold at a second injection site. We repeat the rank test at a different site, the adaLN shift, which lies on the conditioning pathway. We optimise a shift increment there, capped at that site's full-rank norm, at rank 1, 64 and full: it turns the trajectory $-8.31°$, $88.96°$ and $90.20°$. As at the block output ($-7.12°$, $89.71°$ and $90.25°$; Table 14), the effect grows with rank, so the two sites give the same ordering. A single shared vector optimised for all six clips turns the trajectory by $-3.33°$ when learned at the adaLN shift and by $5.01°$ when learned at the adaLN scale; one shared vector per layer on the conditioning path turns two to four of the six clips most of the way; none turns all six. The two rank sweeps are read separately, each capped at its own site's full-rank norm (284.4 and 168.7), since no single norm is natural for both.

For each of six clips we take the change the target condition makes to the layer-2 adaLN shift, and add these vectors at both sites at the same norm; this separates the sites. Replacing each clip's modulation shift with its target turns the trajectory $28.21°$. Broadcasting the same vectors across all 576 positions at the block output, at the same norm, turns it $9.81°$, and scaling that down to 91.64 and 46.07 gives $6.50°$ and $2.74°$, so the effect follows the norm. At matched norm, the shared vector produces more rotation at the modulation shift than at the block output, because the modulation is where the model's own conditioning enters and is already shared across positions (Section 3.1); a vector added at the block output is not on that pathway.

site	norm	6 clips	100 clips
adaLN shift, layer 2	123.03	$\mathbf{28.21}°$	$\mathbf{25.30}°$
block output, layer 2	123.03	$\mathbf{9.81}°$	$\mathbf{8.66}°$

Table 9: The same vectors (one per clip), the change the requested condition makes to the layer-2 adaLN shift, added back at two sites at the same norm; a requested $90°$, no frame lost.

Two more variants of the injection site. Under the same protocol (Table 9), adding the same vectors to layer 2's adaLN *scale* instead of its shift turns the trajectory $2.52°$, and replacing all six modulation components turns it $34.97°$.

One shared vector at three sites of every layer. Under the same objective, cap and budget, a single shared vector is optimised at the block output, at the attention output before its gate, and at the MLP output before its gate, in each of the twelve layers. The thirty-six shared vectors, including the layer-2 block-output vector of Figure 4c ($-1.93°$), turn the ball by a median between $-8°$ and $+4°$; in layers 0 to 8 the ball fades, and in layers 9 to 11 it stays intact and does not move.

Downstream mixing creates only limited positional variation. A shared vector is uniform only at the injection point. Downstream nonlinear layers mix tokens and can therefore introduce changes that vary across tokens after injection. We measure how much positional variation develops as the intervention propagates through the network. At the final layer, over the held-out videos, the share of the induced change that varies across tokens, the offset share, is 4.5 to 6.6% for a norm-matched random shared vector, 5.6 to 9.5% for the shared vector of a field that works, and 7.0 to 13.8% for the best shared vector an optimiser finds, against 80.3 to 89.5% for the token-wise field. Each range is over nine measurements, injection at layers 0, 2 and 3 at three noise levels of 0.9, 0.5 and 0.1. Because injections at different depths have travelled different numbers of layers by the final layer, we re-read the same recordings at a fixed number of layers travelled, which removes that confound. At the injection layer itself every shared vector reads 0% and the token-wise field 94.7%, and Table 10 shows the offset share growing with distance: the best shared vector found goes from 3.0 to 5.6% after three layers to 7.0 to 13.8% after eight. At no matched distance does any shared form reach the token-wise field, which stays between 83.8 and 94.0% throughout. The downstream layers do produce a change that varies across tokens from a shared one, but six to thirteen times smaller in share than the one the token-wise field produces.

layers travelled	random shared	the field's shared vector	best shared found	the field itself
3	2.3–3.3	3.5–6.6	3.0–5.6	**90.4–93.8**
5	3.8–4.8	5.5–10.5	5.2–10.8	**87.9–94.0**
8	4.5–7.2	5.6–14.1	7.0–13.8	**83.8–93.4**

Table 10: Offset share at matched numbers of layers travelled.

C.3 Intervals for the main table

Table 11 gives the bootstrap interval of every entry of Table 1.

		Request	Shared vector ActAdd	Shared vector ITI	Shared vector REINS	Shared vector Optimised	Rank-1 field TokenDial	Token-wise field (ours) Rank-16	Token-wise field (ours) Full rank	Control Random
BallDiT-24M	Angle	$+30^\circ$	[17.9, 20.4]	[17.6, 22.5]	[−4.2, −0.2]	[−8.2, −1.1]	[26.8, 32.7]	[94.9, 100.6]	[99.9, 103.4]	[0.4, 2.1]
		$+90^\circ$	[5.4, 8.7]	[3.4, 5.0]	[−0.2, 1.7]	[−5.1, −0.3]	[−11.6, 8.2]	[77.8, 86.3]	[98.0, 103.5]	[0.6, 1.3]
		$+150^\circ$	[0.2, 1.8]	[0.0, 1.0]	[−0.8, 0.7]	[−4.4, −0.2]	[−32.1, −7.9]	[98.1, 99.9]	[99.0, 100.1]	[0.9, 1.4]
	Speed	$\times 0.67$	[58.1, 61.5]	[61.7, 69.4]	[35.2, 40.4]	[21.8, 23.6]	[75.9, 80.8]	[97.4, 100.8]	[99.4, 100.5]	[0.4, 0.9]
		$\times 1.5$	[55.8, 64.4]	[75.6, 86.3]	[23.0, 27.6]	[28.0, 30.5]	[80.3, 91.9]	[98.6, 103.0]	[100.5, 103.4]	[−1.0, −0.5]
		$\times 2$	[27.3, 30.1]	[64.1, 75.3]	[14.9, 17.0]	[24.4, 26.4]	[73.6, 81.0]	[96.0, 99.2]	[96.3, 99.5]	[−0.5, −0.2]
	Hue	$+90^\circ$	[14.1, 14.8]	[3.7, 4.0]	[3.8, 4.2]	[41.7, 42.5]	[53.9, 57.2]	[99.7, 100.6]	[98.9, 99.6]	[1.0, 1.4]
	Bright.	$0.8 \to 0.5$	[54.4, 54.6]	[54.8, 55.0]	[22.5, 22.7]	[59.5, 59.8]	[39.2, 39.4]	[64.4, 64.7]	[73.6, 73.8]	[−0.2, −0.1]
Wan2.1 1.3B	Angle	$+90^\circ$	[0.4, 0.9]	n/a	[−0.1, 0.0]	[9.5, 17.1]	[46.6, 59.3]	[92.3, 97.9]	[93.7, 99.9]	[−0.1, 0.5]
	Speed	$\times 1.5$	[0.1, 1.0]	n/a	[−0.1, 0.1]	[64.6, 76.4]	[67.2, 79.6]	[92.5, 108.8]	[95.8, 107.3]	[−1.6, −0.4]
	Hue	$+90^\circ$	[32.2, 32.9]	n/a	[8.3, 8.6]	[47.5, 50.4]	[96.6, 97.2]	[99.7, 101.0]	[99.8, 100.5]	[1.0, 1.5]
	Bright.	$0.8 \to 0.5$	[121.8, 122.3]	n/a	[91.8, 92.4]	[100.1, 100.6]	[92.7, 93.5]	[99.5, 99.9]	[99.8, 100.0]	[0.7, 0.7]

Table 11: Bootstrap 95% intervals for the medians in Table 1, in percent of the request: held-out videos are resampled 10,000 times and each interval spans the 2.5th to 97.5th percentile of the resampled medians. Rows and columns as in Table 1. n/a: Wan has no numeric condition for ITI to probe.

C.4 Scale sweep for the published methods

Table 12 gives every scale and λ tried for ActAdd, ITI and REINS.

How the published methods are applied. Every method is applied at the output of layer 2 of BallDiT-24M at every denoising step unless noted. *ActAdd:* the difference between layer-2 activations under the target and the source condition, computed on the same noisy input at every step and averaged over tokens and steps on the fitting clips. It is added identically at every token; the original method adds a prompt-length difference at the first positions of a text prompt, which has no counterpart among video tokens. *ITI:* the smallest shift that moves the ridge probe of Appendix B by the requested change, added at every token at the block output rather than to selected heads. For launch angle, both directions depend on the clip's starting angle and are fitted on the fitting clips only. Both are matched to the norm of the row's full-rank field before scaling. *REINS:* the leading supervised-PCA direction separating source from target activations, averaged over tokens and the first 25 steps. It is added at every token during those 25 steps, after which each channel's norm across tokens is restored; λ is chosen on the fitting clips. *TokenDial:* one direction whose magnitude varies freely across tokens (a rank-1 field with A free), fitted with the same objective and norm cap as the full-rank field. On Wan2.1-1.3B, ActAdd and REINS are built the same way at layer 22; ITI is not applicable because Wan has no numeric condition for a probe to read.

		Request	Method	$\times 0.25$	$\times 0.5$	$\times 1$	$\times 2$	$\times 4$	$\times 8$
BallDiT-24M	Angle	$+30^\circ$	ActAdd	4.2	8.8	**19.1**	42.8	82.8	103.2
			ITI	4.7	9.6	**19.5**	40.3	78.8	132.8
		$+90^\circ$	ActAdd	1.7	3.4	**6.7**	15.0	17.7	37.7
			ITI	2.1	**4.2**	8.8	21.1	51.4	26.0
		$+150^\circ$	ActAdd	0.7	**1.1**	0.8	-0.3	24.6	11.7
			ITI	**0.6**	1.3	2.5	1.9	10.5	9.0
	Speed	$\times 0.67$	ActAdd	3.8	7.6	15.3	30.9	**59.6**	100.6
			ITI	4.5	8.8	17.6	34.6	**66.7**	117.4
		$\times 1.5$	ActAdd	4.1	8.1	16.4	32.3	**59.2**	94.4
			ITI	4.9	9.7	19.9	40.7	**81.0**	147.7
		$\times 2$	ActAdd	3.7	7.4	14.6	**28.4**	48.7	71.6
			ITI	4.2	8.6	17.4	35.5	**70.4**	126.0
	Hue	$+90^\circ$	ActAdd	2.8	6.2	**14.6**	39.7	133.3	169.3
			ITI	1.7	**3.9**	10.2	33.8	120.2	-4.7
	Bright.	$0.8 \to 0.5$	ActAdd	4.6	9.4	21.2	**54.5**	130.4	193.4
			ITI	9.0	17.9	33.4	**54.9**	83.2	118.5
Wan2.1 1.3B	Angle	$+90^\circ$	ActAdd	**0.6**	0.8	1.7	5.7	2.2	4.7
	Speed	$\times 1.5$	ActAdd	**0.5**	0.9	-0.8	-5.7	7.3	-0.5
	Hue	$+90^\circ$	ActAdd	**32.5**	76.2	160.4	151.8	147.7	147.3
	Bright.	$0.8 \to 0.5$	ActAdd	**122.1**	201.0	243.9	255.8	257.1	258.0

		Request	$\lambda = 1$	2	5	10	20	50	100	200
BallDiT-24M	Angle	$+30^\circ$	-0.6	-1.0	$\mathbf{-1.5}$	-1.1	1.8	2.5	4.9	8.5
		$+90^\circ$	-0.1	-0.2	-0.1	0.2	**1.0**	1.6	1.3	2.1
		$+150^\circ$	0.0	-0.1	0.0	0.0	$\mathbf{-0.2}$	0.9	0.2	1.4
	Speed	$\times 0.67$	4.9	9.8	22.9	**38.4**	51.7	30.1	16.1	18.8
		$\times 1.5$	3.4	6.8	16.5	**25.0**	14.3	-16.4	-39.9	-40.4
		$\times 2$	2.0	4.0	10.0	**16.1**	11.5	-11.2	-25.8	-24.9
	Hue	$+90^\circ$	1.3	2.2	3.5	**4.0**	4.0	3.8	-8.1	-15.6
	Bright.	$0.8 \to 0.5$	1.4	3.2	8.2	13.8	18.7	**22.6**	20.6	19.5
Wan2.1 1.3B	Angle	$+90^\circ$	**0.0**	0.0	-0.1	0.0	-3.2	24.8	33.5	41.0
	Speed	$\times 1.5$	**0.0**	0.1	0.7	1.3	-7.1	-19.8	-12.5	16.1
	Hue	$+90^\circ$	0.9	**8.5**	19.9	31.4	74.8	107.2	117.5	123.6
	Bright.	$0.8 \to 0.5$	9.1	27.5	67.6	**92.0**	127.4	174.0	182.7	188.6

Table 12: Scale sweep for the published methods in Table 1. Top: ActAdd and ITI. The factor multiplies a vector that starts at the norm of the full-rank field (for $+150^\circ$, the field fitted for $1{,}000$ iterations). Bottom: REINS at each λ. Entries are the median change over held-out videos, as a percentage of the requested change. Bold marks the setting chosen on the fitting set by latent error, which is the value in Table 1. Grey entries damage the image: the ball fades, is mottled or turns black, or the whole frame breaks, judged by eye on held-out clips. A grey number is therefore not a controlled change. At $+90^\circ$ and $+150^\circ$, every setting that turns the launch angle by more than 10% of the request damages the image.

C.5 The shared-vector result across injection layers

This section backs the cross-layer statement in Section 4.2. At each of six layers we independently fit a token-wise field to the fitting set under the same objective, norm cap, and optimisation budget, split it with Equation (6), and evaluate the full field, its shared vector, and its offset on the same held-out videos (Table 13). Because each layer is fitted from scratch, the layer-2 row is a separate run from the field split in Table 4 and its numbers need not match it. The shared vector turns the trajectory by only -0.20° to 0.24° at every layer. The full field turns it 85.19° to 91.59° at layers 1, 2, and 4, and less at layers 6, 8, and 10, where even the full field starts to fail. The negative shared-vector result is therefore not a property of layer 2: it holds at every layer where the full field works, so it cannot be explained by an injection site chosen to disfavour the shared form.

C.6 Rank and intervention structure

This section backs Section 4.2: a small number of learned directions can suffice, whereas a rank-1 field, one direction whose length varies across tokens, does not under our optimisation. It examines how control changes with rank, whether the observed threshold is stable across training seeds, and whether rank alone explains performance.

	rotation (°)			self-run
layer	full field	shared vector	offset	offset share
1	91.57	−0.16	87.73	0.8356
2	91.59	0.24	85.30	0.8301
4	85.19	−0.18	77.90	0.9378
6	31.91	−0.20	26.80	0.9795
8	11.75	0.01	11.12	0.9869
10	−2.37	−0.08	−1.30	0.9850

Table 13: **Layer sweep for the shared-vector result.** Each row independently fits a token-wise field at the indicated layer and evaluates the full field, its shared vector, and its offset on the same held-out videos. Rotations are in degrees for a requested 90° change. The last column reports the positional offset share of the self-run activation difference averaged across denoising steps; it is distinct from the same-z offset share reported in Table 19. Among the six tested layers, layer 2 has the lowest self-run offset share.

Control depends on both rank and magnitude. Figure 11 plots rotation against worst-frame ball saturation at three norm caps. We repeat the sweep at caps of 20 and 40, on the same fitting set and held-out videos. At cap 20 no rank rotates the trajectory, and no frame is lost. At cap 40 the rotation grows, but the image breaks first: 234 frames are lost and the saturation drops to 46, and the request is still missed. Only at the cap of 168.7 does a high rank reach the request with the ball intact. These sweeps show that control depends on both rank and intervention magnitude. Under the smaller norm of the +90° launch-angle row of Table 1 (77.54, the norm of that row's full-rank field), rank 16 reaches only 82.5% of the request and fades the ball in every video. Table 14 lists, for each rank, the free parameters, the final loss and the image-quality metrics.

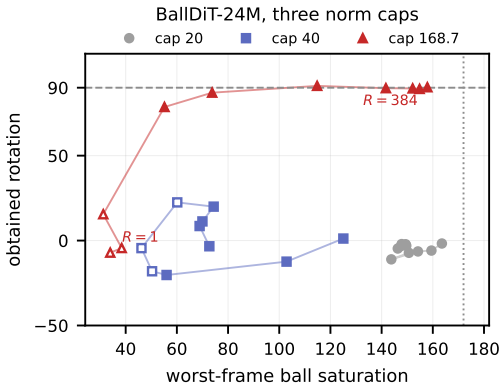


Figure 11: **BallDiT-24M rank against image quality at three norm caps.** Launch-angle rotation versus worst-frame ball saturation at caps 20, 40 and 168.7; each point is one tested rank, and the vertical reference at 172 marks un-intervened saturation.

Rank behaves the same way across requests. (Table 15) We run the same sweep at six launch-angle rotations and three hue rotations, on the same fitting set and held-out videos. The smallest rank that works for each request is read from this table. Reaching the target and keeping the ball are two separate things. At a 120° request the target is met at rank 32, but the saturation recovers only near rank 128. Turning the launch direction by +180° or −180° lands it in the same place, so the sign of the reading is meaningless at a 180° request.

The rank threshold is stable across training seeds but depends on optimisation. (Table 16) We repeat the rank sweep on two more sets of weights, trained from different seeds. Each field is fitted on that model's own fitting set and read on the held-out videos. At rank 4 and below the image breaks on every set of weights. From rank 16 on the median stays within 3° of the requested 90°. Across all three independently trained sets of weights, the smallest rank meeting the criterion is $R = 16$.

directions across positions (R)	free parameters	final loss	rotation (median, °)	lost frames of 3,300	worst-frame saturation
1	960	0.0400	−7.12	703	34.0
2	1,920	0.0354	−4.42	699	38.4
4	3,840	0.0321	15.50	895	31.2
8	7,680	0.0147	78.63	32	55.2
16	15,360	0.0074	**87.08**	**0**	73.8
32	30,720	0.0034	91.07	0	114.8
64	61,440	0.0018	89.71	0	141.7
128	122,880	0.0011	89.56	0	152.1
256	245,760	0.0007	89.27	0	154.8
384	552,960	0.0006	90.25	0	157.9
no intervention	—	—	0	0	172

Table 14: The values of R of Figure 4 with their free parameters, final loss and image-quality metrics, at a requested 90° read on the held-out videos. Sixteen directions reach the request with no frame lost, but the ball is heavily desaturated there; recovering the saturation needs more directions, which barely change the rotation. The largest value, 384, is full rank: a $(576, 384)$ field can hold no more independent directions than that. It is optimised as $\delta = AV$ with $R = 576$, which is where its parameter count comes from.

attribute	request	R=1	R=2	R=4	R=8	R=16	R=32	R=64	R=128	R=256	full
launch angle	30°	15.8 [0]	18.3 [0]	27.6 [0]	30.2 [0]	30.0 [0]	30.6 [0]	30.7 [0]	30.5 [0]	30.3 [0]	30.5 [0]
	45°	16.9 [0]	17.7 [3]	36.5 [0]	43.4 [0]	45.1 [0]	46.0 [0]	45.6 [0]	45.7 [0]	45.4 [0]	45.2 [0]
	90°	−7.1 [703]	−4.4 [699]	15.5 [895]	78.6 [32]	87.1 [0]	91.1 [0]	89.7 [0]	89.6 [0]	89.3 [0]	90.3 [0]
	120°	−23.3 [994]	−10.2 [981]	−26.9 [984]	19.1 [1197]	119.8 [243]	122.1 [16]	118.8 [2]	122.4 [0]	120.4 [0]	120.3 [0]
	150°	−30.6 [1239]	−1.3 [1120]	−23.1 [991]	−28.5 [1175]	150.9 [416]	150.8 [5]	149.4 [0]	149.4 [0]	149.5 [0]	149.1 [0]
	180°	−26.4 [1276]	3.8 [1098]	−18.5 [932]	−37.5 [1166]	−163.2 [262]	−171.4 [2]	168.8 [0]	169.7 [0]	−168.1 [0]	174.1 [0]
hue	30°	30.6 [0]	30.6 [0]	30.5 [0]	31.5 [0]	31.5 [0]	31.5 [0]	31.7 [0]	31.7 [0]	31.6 [0]	31.6 [0]
	90°	70.4 [0]	80.1 [0]	85.3 [0]	88.7 [0]	89.1 [0]	89.4 [0]	89.8 [0]	89.8 [0]	89.7 [0]	89.7 [0]
	180°	−170.8 [188]	−174.6 [0]	−174.0 [0]	−176.3 [0]	−176.9 [0]	−176.6 [0]	−176.5 [0]	−176.7 [0]	−176.8 [0]	−177.0 [0]

Table 15: Request size against attribute in the rank sweep, read on the held-out videos: median rotation obtained (°) [frames lost of 3,300]. Both attributes are swept at ten values of R; the last column is the full-rank field, optimised with $R = 576$ and of rank 384.

R	BallDiT-24M	seed 2	seed 3
1	−7.12° / 703	5.06° / 684	0.76° / 570
2	−4.42° / 699	8.01° / 764	−9.51° / 1291
4	15.50° / 895	51.19° / 381	14.61° / 1098
8	78.63° / 32	80.97° / 4	76.53° / 71
16	**87.08° / 0**	**90.22° / 0**	**88.19° / 1**
32	91.07° / 0	89.00° / 0	89.95° / 0
64	89.71° / 0	88.63° / 0	87.38° / 0
128	89.56° / 0	89.39° / 0	87.89° / 0
256	89.27° / 0	88.62° / 0	89.16° / 0
384 (full)	90.25° / 0	89.31° / 0	87.83° / 0

Table 16: The rank sweep on three sets of weights under the matched protocol: each field is fitted on that model's own fitting set and read on held-out videos, at a requested 90°. Entries are the median rotation obtained and the frames lost of 3,300. Bold marks the smallest R at which the median distance to the target falls under a tenth of the request with fewer than 33 frames lost; it is $R = 16$ on all three.

The threshold does depend on where the search starts. From a zero start, the field first meets the criterion at rank 16. Started instead from a truncated SVD of a field that already works, rank 4 meets the angle criterion at all nine requests. But the image pays for it: at the three largest requests the worst-frame saturation falls to 57 to 81, against 173 for the un-intervened clips of this batch. We therefore read the sweep as a capacity check, not as the intrinsic dimension of motion control.

The split replicates across seeds, layers and a larger model. (Table 17) We repeat the shared-vector-and-offset split on the two further seeds. Each field is optimised on its own clips, frozen, split in place, and injected on the held-out videos. It also holds on model-M, a larger separately trained

model: there the shared vector moves the ball at most $1.5°$ at every layer, while the offset reaches the target at the deeper layers.

weights	full	shared vector	offset	norm-matched random
BallDiT-24M (Table 4)	$90.25°$	$-0.46°$	$89.91°$	$1.28°$
BallDiT-24M, second seed	$89.31°$	$0.18°$	$88.55°$	$-0.65°$
BallDiT-24M, third seed	$87.83°$	$-0.60°$	$82.46°$	$1.21°$

Table 17: The split on three sets of weights. No intervention on any of the three loses a frame.

Parameter count, number of directions, and retained energy are distinct quantities. The forms in Table 18 tie their parameters in different ways, so the number of directions is read off the rank sweep instead (Table 14), where only R changes. Table 18 shows that the three quantities do not move together.

On the held-out videos, the number of videos that meet the criterion rises with rank: none up to rank 4, then 28 at rank 8 and 69 at rank 16, where the median passes.

Joint spatial and temporal variation carries most of the rotation. The offset splits, with no optimisation, into three parts: one that changes only over time, one only over space, and one over both. Added on their own, they turn the trajectory $2.2°$, $11.4°$ and $57.1°$ (medians over six clips). No frame is lost. Most of the effect comes from the part that changes over both space and time.

The positional structure is concentrated along the object's trajectory. The offset of a working field is not distributed uniformly over the 576 tokens. Its ten highest-energy tokens contain 9.14% of the total energy, against 1.74% under a uniform distribution. Across the fitting set, each token's energy correlates with how often the ball occupies each token at Pearson $r = 0.42$, and none of $2{,}000$ token permutations reaches this correlation. The 49 tokens the ball passes through contain 23.47% of the energy, against a median of 8.31% over the $2{,}000$ permutations. Keeping only these 49 tokens produces $52.06°$ of rotation, whereas keeping the remaining 527 tokens, which contain 76.53% of the energy, produces only $26.14°$; the full offset produces $89.91°$. The control effect is therefore disproportionately concentrated at token positions along the object's trajectory.

Layer and denoising-step variation act jointly in the successful shared intervention. A shared vector that varies jointly over layer and denoising step (384×12 layers $\times$ 50 steps $= 230{,}400$ numbers) produces $86.34°$ of rotation (Figure 4c), whereas varying only over layer or only over denoising step does not produce comparable control. We further decompose the successful layer-by-step intervention into a step-invariant component and a step-varying remainder. The step-invariant component, one vector per layer held across all denoising steps, causes the ball to be lost on all held-out videos. The step-varying remainder produces only $2.89°$ of rotation. The successful intervention therefore cannot be attributed to either component alone; the full layer-by-step structure matters for the observed control.

Rank matters, but rank alone does not explain control: the directions, and where they are placed across space and time, matter too.

C.7 Scope across attributes

This section supplements Section 4 and the Conclusion. It reports which attributes a shared vector fails on.

The shared-vector limitation depends on the attribute. A shared vector does not fail on every attribute. We trained a set of weights whose condition is a brightness multiplier applied to every pixel, and asked for a factor $0.8 \rightarrow 0.5$. A shared vector reaches 59.6% of this request, against 73.7% for a token-wise field and -0.2% for a norm-matched random vector, with no frame lost. The shared vector moves the launch angle by a median $1.9°$ from un-intervened (at most $12.3°$), against $0.6°$ for the token-wise field. It is much weaker on launch angle and hue, where a token-wise field works far better, but it handles most of a global brightness change; on speed it reaches 22 to 30% of the request across the three speed requests of Table 1 (Table 19).

parameterisation	free parameters	directions across positions	energy of the solution kept	projected natural	projected equal norm	optimised directly	lost frames of 3,300, direct
one vector, shared	384	1	5.3%	$-0.46°$	$1.10°$	$-1.92°^{\dagger}$	41
rank-1	960	1	9.2%	$-3.12°$	$-5.46°$	$-5.97°^{\dagger}$	701
rank-2	1,920	2	15.5%	$-2.45°$	$3.59°$	$-4.41°^{\dagger}$	699
one per temporal slice	3,456	9	8.0%	$-0.53°$	$0.03°$	—	—
rank-8	7,680	8	37.8%	$39.34°$	$50.49°$	$78.62°$	32
one per spatial position	24,576	64	20.5%	$18.67°$	$37.84°$	$70.44°$	5
token-wise field	221,184	384	100%	$90.25°$	—	$91.59°$	4

Table 18: Seven parameterisations at one site, a requested $90°$ throughout. The projected columns take the whole field, of norm 156.30, and project it onto each form, once at the projection's own norm and once rescaled to 156.30; the equal-norm cell is empty for the token-wise field because that form is the field itself, already at that norm. The optimised-directly column optimises in each form directly, capped at Frobenius norm 168.704, the norm the unconstrained AV ($R = 576$) optimisation settles on; the temporal-slice form has no directly optimised reading. The lost-frame column belongs to the directly optimised column: the projected fields lose no frame at their own norm, and five and two frames at the equal norm for rank-2 and rank-8. A dagger marks the three forms whose direct optimisation loses 33 or more of the 3,300 frames; their angles are listed but do not count as control. The directly optimised column is a separate fitting run from Table 14 under the same protocol; the two agree within $0.01°$ at rank-2 and rank-8, while rank-1, whose ball is mostly lost, reads $-5.97°$ here and $-7.12°$ there. Its token-wise-field row optimises the whole $(576, 384)$ array directly and stops at norm 77.53, a different solution from the field optimised as AV with $R = 576$ in the natural column.

attribute	request	offset share	shared vector	token-wise field
launch angle	$+90°$	0.41–0.68	-2.1%	101.9%
speed	$\times 1.5$	0.17–0.24	29.5%	102.1%
hue	$+90°$	0.06–0.08	42.0%	99.2%
brightness, whole frame	$0.8 \to 0.5$	0.03	59.6%	73.7%

Table 19: Four attributes: how non-uniform the model's own response is, and how far each form gets. Offset share is the fraction of squared norm in the offset of the layer-2 activation change when the condition changes at the same latent state; ranges are over six clips. Brightness is on the weights trained with that conditioning. The percentages are medians, with the shared vector at the same Frobenius norm as the token-wise field; they report the change as a percentage of the request, as in Table 1, obtained by changing the condition directly, values above 100% indicating overshoot.

The field is tied to positions in the frame. We shift twelve held-out clips (28 to 39) 18 pixels to the right, about 1.1 token columns, fill the vacated columns with white, invert each shifted clip to noise with 100 steps and regenerate it with 50. On the unshifted clips, the working field of Figure 3 turns the trajectory by a median $89.23°$. Moved by the measured ball shift with bilinear interpolation, the same field turns the shifted clips by only $43.31°$ (9.3 to $64.3°$); left in place it turns them by $58.8°$, and moved the opposite way by $-12.6°$. Each rotation is read against the same layout without the field.

The split holds with a second ball in the frame. We repeat the split on a model trained on two balls in straight-line motion, changing only the red ball's launch angle by $90°$ and leaving the blue ball's condition unchanged. On the held-out videos, the offset turns the red ball by a median $72.75°$, its shared vector by $0.23°$, and a norm-matched random field by $0.55°$. Under every intervention the blue ball stays on its original trajectory, shifting by about 0.04 in a frame of width 10, as it does under the random field. The red ball fades after frame 21 under the offset, losing 4 to 12 frames in 56 of the 100 clips.

D CONDITIONING-RELATED COMPONENTS

This appendix finds the layers and heads that reproduce the conditioning-induced rotation when their outputs are replaced. It then tests whether the field's effect depends on them.

D.1 CONDITIONING-INDUCED ROTATION IS CONCENTRATED IN EARLY LAYERS

Contiguous layer windows block more than scattered ones. (Table 20) We first replace the conditioning at all twelve layers, which rotates the trajectory $89.95°$. We then pin selected MLP layers to their un-intervened outputs and measure the rotation that remains. We compare contiguous windows against scattered sets of the same size, drawn so no two chosen layers are adjacent, on 100 conditions drawn afresh for this experiment.

layers pinned	rotation left, contiguous $\{3, \dots\}$	lost frames	rotation left, scattered draws (range)	their lost frames	lowest scattered group whose image holds (lost)
3	$38.04°$ $[37.04, 39.33]$	0	36.45–$87.09°$	0–89	$56.81°$ (5)
4	$15.96°$ $[15.25, 17.13]$	0	56.34–$87.28°$	0–152	$65.05°$ (0)
5	$9.89°$ $[9.27, 10.45]$	0	37.55–$80.94°$	0–177	$61.02°$ (0)
6	$8.44°$ $[7.86, 9.27]$	0	19.72–$64.78°$	6–206	$64.78°$ (6)

Table 20: Contiguous MLP windows against scattered sets of the same size. Replacing the conditioning at all twelve layers turns the trajectory $89.95°$ with nothing pinned; each entry is how much of that is left once the named MLPs are held at their un-intervened outputs. Medians over 100 conditions drawn afresh for this table, with bootstrap intervals on the median; lost frames are out of 3,300.

For windows of four or more layers, a contiguous window blocks more rotation than any scattered set of the same size, and it loses no frame, while scattered draws lose up to 206 frames and dull the ball. At three layers one scattered draw ($\{3, 5, 7\}$) reaches $36.45°$ against the contiguous $38.04°$, but it loses 23 frames where the contiguous window loses none. Image quality separates the two forms at every size. Spread-out layers block less and damage the image more, so the effect depends on which layers are pinned, not only on how many.

The tested later-layer groups contribute little conditioning-induced rotation. Replacing the modulation one layer at a time leaves layers 6 to 11 at 2% or less each. This alone would not rule out a group of them acting together, so we replace them in groups. Layers 6–8 together move the trajectory $1.25°$, layers 9–11 move $0.20°$, and all six move $1.48°$, on the six-clip batch, against $84.37°$ for layers 2, 3 and 5 together, and $90.25°$ for all twelve. No frame is lost. Most of the conditioning-induced rotation is concentrated in the early layers; the tested groups of layers 6 to 11 contribute little.

D.2 A SMALL HEAD GROUP REPRODUCES MOST CONDITIONING-INDUCED ROTATION

Counterfactual replacement of a small head group reproduces most of the conditioning-induced rotation. We rank the 72 heads on the fitting set. For each head we replace its output with the counterfactual output, the counterfactual replacement of Section 3.2.2, and read how far the launch angle moves. The five highest-ranked heads, written as (layer, head), are $(4, 2)$, $(2, 2)$, $(5, 4)$, $(3, 3)$ and $(5, 3)$. Replacing each of them on its own moves the trajectory: the top head by $42.31°$ and the next four each by 14 to $17°$, while the median over all 72 heads is $-0.01°$. On the held-out videos, which take no part in the ranking, replacing the five heads together moves the launch angle $84.63°$, against $87.54°$ for an eight-head group ($87.28°$ in Table 21, which uses another draw of starting noises) and $-13.91°$ to $+1.89°$ for thirty layer-matched random five-head groups. The five-head group therefore reproduces most of the conditioning-induced rotation, and the body uses these five.

The head-level result holds across conditioning formats. (Table 21) We next test whether the head-level concentration depends on the conditioning format. On models we train, we compare adaptive layer normalisation with cross-attention, with numeric text, and with an English sentence encoded by Wan's frozen text encoder. In each case we write the target condition, so the counterfactual stays exactly defined. The head-level concentration holds for all three alternative formats.

Table 21 gives, for each format and set of weights, the rotation the ranked head group produces. In all thirty cells none of ten layer-matched random groups of the same size reaches the ranked group, and the ranked groups lose at most 2 of 3,300 frames. From eight heads on, the random groups

heads replaced	1	2	4	8	16	32
adaLN	41.68	54.38	83.20	87.28	89.14	89.20
adaLN, second seed	45.82	71.43	79.66	87.46	88.69	89.43
cross-attention	30.34	52.26	68.93	78.74	84.70	86.28
numeric text	41.67	63.53	72.19	78.90	84.44	86.31
sentence	38.59	55.71	65.44	73.19	83.35	85.71

Table 21: Head-level concentration across conditioning formats. Rows name how the condition arrives: three scalars through adaptive layer normalisation (two seeds), three scalars through cross-attention, the three numbers written as numeric text, and an English sentence, the last two through Wan's frozen text encoder. Each model's heads are ranked on its own fitting clips (0–23); entries are median rotations over held-out clips 28–127 at a requested $90°$.

weights	condition replaced at all twelve layers	+ own five heads pinned	+ layer-matched random five pinned
BallDiT-24M	$89.85°$	$\mathbf{5.72}°$	$97.88°$
second seed	$90.16°$	$\mathbf{7.94}°$	$75.35°$
cross-attention	$90.30°$	$\mathbf{24.05}°$	$90.03°$

Table 22: Pinning the group of five, on three separately trained sets of weights. Entries are medians at a requested $90°$; no intervention loses a frame and the ball's pixel count stays near its un-intervened value. Pinning the five heads after the condition is replaced removes most of the rotation on BallDiT-24M and the second seed ($5.72°$ and $7.94°$) and much of it on the cross-attention set ($24.05°$), whereas layer-matched random groups of five leave the rotation in place. Rotations are on held-out clips 28–127.

requested (°)	15	30	45	60	90	120	135	150	−45	−90	−135
obtained (°)	13.0	26.5	40.4	54.5	84.6	115.5	130.4	145.5	−39.9	−82.5	−131.8
obtained / requested	0.87	0.88	0.90	0.91	0.94	0.96	0.97	0.97	0.89	0.92	0.98

Table 23: Injecting at the group of five heads, with the requested rotation swept. Each entry is a median; the third row is the second divided by the first. No setting loses a frame. Requests of $\pm 180°$ are omitted: for a periodic attribute the direction of a half-turn is undefined.

themselves damage the image, so their rotations are read on broken frames. On the cross-attention weights, replacing one to four ranked heads roughens the ball's edge in about half of the clips while the trajectory still turns. [1]

Pinning the same five heads also removes most of the rotation produced by replacing the condition itself, on three sets of weights (Table 22).

Table 23 gives the eleven requested rotations of Figure 5b.

Hue is associated with a distinct head group. We run the same ranking and sweep for hue. Its group is four heads in layers 8 and 9, disjoint from the five launch-angle heads. Over eleven settings its rotation is signed and monotone, at 0.54 to 0.64 of the request, against 0.87 to 0.94 on launch angle for requests up to $90°$; no frame is lost. Launch angle and hue are associated with distinct head groups under the same procedure.

D.3 The field depends on the selected heads

Pinning the five heads removes most of the field's effect. The field is not re-optimised. The field alone turns the trajectory $90.25°$. When the five heads are held at their un-intervened outputs, the rotation drops to $14.66°$, with no frame lost. Thirty layer-matched random groups of five heads keep the rotation between $77.13°$ and $108.64°$. Pinning alone neither harms the image nor moves

[1] The step from three scalars to a sentence changes several things at once: natural language, 24 conditioning tokens instead of 3, width 4096 instead of 16, a frozen pretrained encoder, and an angle spelled across subword tokens. This shows that the head-level concentration holds under the whole bundle of changes; it does not isolate which individual change matters.

the trajectory. The drop is therefore specific to these five heads. The result replicates on a second, independently trained set of weights (Table 25).

Re-optimisation under pinning does not distinguish the selected heads. With the selected five heads pinned, a newly optimised field reaches $56.56°$ on held-out videos. Repeating the same procedure with four layer-matched random five-head groups gives held-out rotations of $49°$ to $69°$, with a median of $58.18°$ across groups. The loss of held-out control under re-optimisation is therefore not specific to the selected heads. The selected-head condition incurs a substantially larger image-quality cost, losing 233 frames, whereas the random-head runs lose none.

The heads are attribute-specific. (Table 24) We identify four hue heads with the same replacement procedure used for launch angle. Injecting hue into them moves it $166°$ of a requested $180°$, against 0.6 to $7.0°$ for random groups. Injecting through them moves hue; whether hue needs them is tested by pinning. We then pin one attribute's heads and read the other (Table 24). Pinning an attribute's own heads removes the effect of its field, and pinning the other attribute's heads does not. At a $90°$ launch-angle request, pinning the launch-angle heads drops the rotation from $89.23°$ to $17.69°$, while pinning the hue heads leaves it at $88.23°$. Hue at $30°$ and $60°$ shows the reverse. We do not count the three larger hue requests ($120°$, $150°$, $180°$) as a separation. At $120°$ the two head groups are not cleanly separable, and at $150°$ and $180°$ the hue field moves the hue the wrong way even without pinning.

requested (°)	field only	+ hue heads	+ random four	+ launch-angle heads
launch angle				
90	89.23	88.23	83.87	**17.69**
hue				
30	20.53	**4.51**	19.21	18.82
60	37.76	**8.06**	34.28	30.57
120	36.12	**6.64**	25.62	24.52
150	−5.66	5.83	−1.45	−9.65
180	−25.52	−3.84	−16.97	−23.52

Table 24: Attribute specificity on the six clips. Each attribute's field is read on its own attribute, with its own heads or the other attribute's heads pinned. Each column after the first pins the named heads. Entries are the degrees obtained. Pinning an attribute's own heads removes the effect of its field, and pinning the other attribute's heads does not, at $30°$ and $60°$ of hue. At $120°$ the launch-angle heads take 36.12 down to 24.52, below the random four at 25.62, so the two groups are not fully separable at that request. At $150°$ and $180°$ the hue field moves the hue the wrong way even without pinning, so those rows are not read as a separation.

weights	field only	own five pinned	blocked	layer-matched random
BallDiT-24M	$90.25°$ $[89.5, 91.3]$	$14.66°$ $[6.2, 19.0]$	83.8%	77.13–108.64 (30)
BallDiT-24M, second seed	$89.20°$ $[88.7, 90.1]$	$13.20°$ $[11.3, 14.2]$	85.2%	52.78–101.11 (30)

Table 25: Pinning the launch-angle heads, replicated on two sets of weights. Blocked $= 1-$ pinned / field only. On the second seed the field is the $R = 576$ field refitted under the norm cap (norm 132.4); Table 17 uses the uncapped one ($89.31°$). Brackets are 95% bootstrap intervals on the median; the last column gives the range of medians across the 30 layer-matched random groups.

Injecting after the selected heads weakens control. The five launch-angle heads sit in layers 2 to 5. In a separate run, a field added at layer 9 comes after all five heads, so within a denoising step it does not pass through them. With the same optimisation, its loss only falls to 0.038, against 0.004 at layer 2, and the clips turn only $20.83°$ and lose 43 frames. Adding the field after the heads therefore weakens control, although a different depth also changes the rest of the computation. Table 26 shows how control falls with injection depth on the six-clip batch.

layer	Frobenius norm	median (°)	lost of 198
0	77.83	90.59	0
2	91.64	89.23	1
3	94.15	88.94	6
5	112.55	83.76	28
8	128.09	63.27	54
11	106.42	9.96	8

Table 26: Control across injection depths, on the six-clip batch; each layer's field is fitted separately. *Layer* denotes the block output at which the field is added.

The field barely lies along the heads but still depends on them. We also measure how much of the field lies along these five heads. Across three ways of measuring it, 0.0024 to 0.0214% of the field's energy projects onto the change in the heads' outputs, against 0.0002 to 0.0006% for a random direction or subspace. This overlap is tiny, although larger than the random baseline. Yet pinning the heads removes most of the field's effect. The field therefore depends on the heads through what they compute, not by pointing along them.

E GENERALISATION TO A PRETRAINED MODEL

Two main findings generalise to Wan2.1-1.3B. First, the offset of a fitted field carries the control effect, while its shared vector has little effect. Second, the field-induced control depends on a selected group of attention heads, not any single head. On Wan, pinning ten of the 360 heads already removes most of the control, and a forty-head group reproduces it by counterfactual replacement; on the models we train a group of five to eight heads plays the same role.

E.1 TOKEN-WISE FIELDS CONTROL WAN2.1-1.3B

The Wan split quoted in Section 5, $0.1°$ for the shared vector and $86.9°$ for the offset, is the full-rank row of Table 27 below.

The videos and the request. The clips are the simulator's own, the ones the model we train is trained on. Each is inverted into Wan2.1-1.3B by a fixed-point inversion of 50 steps, and generation runs 50 steps from the noise that returns. The prompt is empty and classifier-free guidance is off, so clips differ only in that noise. The counterfactual is the same clip re-rendered by the simulator with its launch angle rotated, inverted the same way; the latent generated from it is the target. Each clip's request is what separates it from its own counterfactual, with a median over the held-out videos of $90.4°$.

Fitting the field. A tensor $\delta \in \mathbb{R}^{2304\times1536}$ is added to the output of block 22 at every generation step, written as $\delta = AV$ with V carrying R rows. One field is fitted on the whole fitting set at once, against the mean squared error to the targets above, with Adam at 3×10^{-3} for 300 iterations. An $R = 256$ field capped at the uncapped full-rank norm $(2{,}188)$ ends at 612; every value of R is then fitted under a cap of 612 (the $R = 256$ refit ends at 605, Table 27). The field is then injected unchanged on clips 28 to 127, which take part in no optimisation, and read with the pixel measurement of Appendix A.

Low-rank fields reproduce the control effect. Table 27 gives every value of R. Four directions reach the request, and from $R = 64$ no frame is lost. The shared vector of the field moves the trajectory by at most $0.1°$ at every capped value of R, while its offset carries almost all of the full field's effect, and a norm-matched random field of the same size moves it $0.0°$ to $0.2°$. A sweep at a requested $30°$, capped at the norm of its own uncapped full-rank run, behaves the same way and loses no frame anywhere: the field turns the ball $20.6°$ at $R = 1$ and $27.4°$ at $R = 256$, with the shared vector at $0.0°$ throughout.

Inversion does not bias the readout. Inverting a clip and regenerating it changes its launch angle by a median of $0.06°$. This is smaller than the readout's own error on the simulator clips $(0.72°)$.

R	norm	field	shared vector	offset	lost
1	339	17.6	0.0	20.2	1099
4	446	87.5	0.0	91.7	27
16	477	88.5	0.0	87.9	3
64	473	87.9	−0.0	88.9	0
256	605	87.5	−0.0	86.6	0
1536 (full)	612	87.9	0.1	86.9	0
1536 (full), uncapped	2188	79.9	0.4	73.8	181
norm-matched random	—	0.0–0.2	—	—	—

Table 27: The field fitted on all fitting clips at once, on Wan2.1-1.3B, at every value of R, for a requested $90.4°$ launch angle. Degrees are medians over held-out videos; lost frames are out of 3,300. *field* is the whole field, *shared vector* and *offset* its two parts injected separately. The first six rows are fitted under a norm cap of 612 for 300 iterations; the last is fitted without a cap, and its lost frames show that the damage comes from the larger norm, not from full rank. The rank-16 entry of Table 1 is a separate fit under the same cap ($85.9°$, 94.9% of the request).

Inversion also shifts each clip's requested rotation, by a median of $0.25°$, while the full-rank field misses the request by a median of $6.1°$. The two are not correlated (Spearman $\rho = 0.01$). The quarter of clips with the largest inversion error reach 97.5% of the request, the rest 97.2%.

E.2 The asymmetry persists across targets and natural prompts

Among the 300 source clips, every pair whose labelled speed and radius agree within 0.02 and whose launch directions differ by 40–$100°$ was kept: 28 pairs. The token-wise field is added at layer 22 of 30 and optimised for 100 iterations against the target clip's latent. Pairs are matched on the labels the two clips were *generated from*, while the requested rotation is measured from the two clips Wan actually rendered, and the two agree closely. Reading the launch direction from the generated pixels by fitting a quadratic to $x(t)$ and to $y(t)$ over the first twelve frames and taking the derivative at $t = 0$, the rendered direction differs from the label by a median $0.92°$ and at most $3.23°$ across all 300 source clips, and the per-pair direction gap differs by at most $4.07°$. [2]

The 28 eligible pairs. The first eight are more tightly matched, with speed and radius differences below 0.013; the other twenty satisfy the broader 0.02 criterion. Over the 28 pairs the token-wise field ends a median 0.26 pixels from the target clip's ball centre (interquartile range 0.20 to 0.30) against 15.76 un-intervened, and its rotation is a median $0.46°$ from the request (interquartile range 0.27 to 0.80), with no frame of any pair losing the ball. The shared-vector intervention remains a median 13.34 pixels from the target (interquartile range 11.23 to 19.84), barely closer than un-intervened, and covers a median 11.1% of the request (interquartile range -3.8% to 21.9%); its negative lower quartile shows it moves the ball the wrong way on some pairs, and it thins the ball ($1{,}022$ median ball pixels against $1{,}088$, counting pixels by red saliency). Reported separately, the token-wise field is if anything better on the looser twenty ($0.42°$ median from the request against $0.59°$), while the shared-vector intervention is worse and more scattered on them (median 6.5% against 13.0%).

A trajectory-only target still produces little shared-vector control. We also run the shared form against a target defined on the ball track alone: the un-intervened clip's own ball centres, rotated $90°$ about the first frame. Over all 28 pairs it closes 4.3% of the distance to that target at the median, quartiles 3.2 and 5.5%, and on one pair it ends further away. Every frame of all 28 pairs still shows a ball: none of the 924 frames loses it, and the median ball pixel count, counting pixels far from white, is $1{,}080$ against $1{,}108$ un-intervened (with the red-saliency count of the previous paragraph, $1{,}037$ against $1{,}088$). The ball is desaturated in every pair and the white background is tinted in ten of them.

[2] The quadratic fit is used rather than the mean displacement over the first eight frames: gravity has already slowed the ball by then, by an amount that depends on the launch direction, so that simpler readout couples the two and is biased on pairs constructed to differ by 40–$100°$.

The optimisation reads ball centres with a differentiable saliency, softmax-weighted over the whole frame; that readout moves when the ball fades, because the weight it assigns to the ball shrinks and spreads. Measured that way the same 28 pairs appear to close 29.7% at the median. All numbers above instead use the thresholded connected-component readout, which is the one every other measurement in this paper uses. The median of the per-pair gaps between the two readouts is 23.9 percentage points.

The five facing pairs. The prompt pairs of Figure 7 name a golden retriever, a horse, a motorbike, an elephant and a shark; in each, the Source faces left and the Target faces right. Read by eye at frames 12 and 24, adding the offset turns all five to face right, and adding the shared vector leaves all five facing left.

Object-localised hue readout. Beyond facing direction, we split the activation change in the same way as in Figure 7 on twenty prompt pairs whose two prompts differ only in the object's colour word, red or blue (Table 28). Hue is the saturation-weighted circular mean of HSV hue over the object mask, averaged over the 33 frames; the red objects measure a median 353° and the blue ones 207°, and the two prompts' objects differ by a median of 148°. The mask comes from the object name, not its colour: we strip the colour word from the noun phrase, locate the object with an open-vocabulary detector (Grounding DINO) and mask it (SAM), once per frame and once per intervention. Because the mask never uses colour, it does not depend on the hue we measure, and it follows the object as it moves. Eligibility is decided on the un-intervened clip alone: the named object must be generated red and be large enough to locate. Sixteen of the twenty pairs are eligible; the four that fail and the one whose object the intervention destroys are listed in Table 28.

The offset changes object hue; the shared vector does not. Of the sixteen eligible pairs, one is a control failure because the intervention destroys the object; the other fifteen are read. On each of the fifteen, adding only the offset moves the object hue by more than 100° (median 142.4°), while adding only the shared vector moves it by a median 5.1° and past 100° on none (Figures 12 and 13; Table 28).

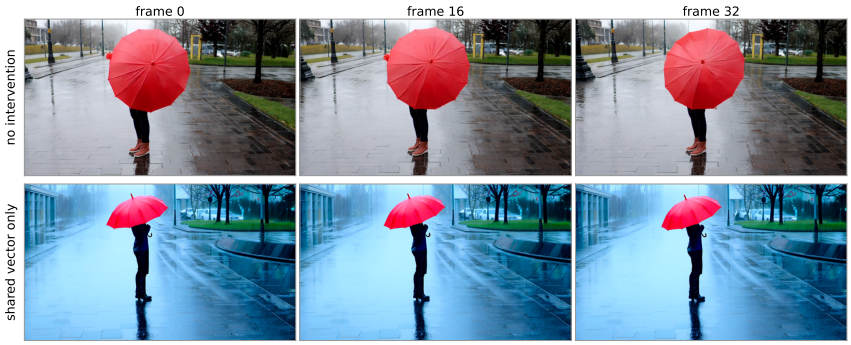


Figure 12: Pair p8. Adding only the shared vector tints the whole frame blue and pulls the camera back, while the umbrella itself stays red.

The shared/token-wise difference persists when source and target differ only in launch angle. (Table 29) The 28 pairs above are two different clips matched on speed and radius. In the six pairs below the two clips differ in nothing but the launch angle: a source clip keeps its speed and radius, its launch angle is raised by 90°, the simulator renders the new clip, and that clip is inverted to a starting noise. Both forms are then optimised exactly as for the 28 pairs and read with the same pixel measurement.

The token-wise field lands 0.29 pixels from the target track, 1.6° from the request. The shared vector moves the track by 4.8° and misses by 94°; what it changes is the ball's size, not its path.

Transfer to held-out seeds of a natural prompt. (Table 30) Each field above is fitted and read on a single clip pair. We also ask whether a single field, fitted jointly on many prompt pairs, transfers



Figure 13: Pair p10. Adding only the shared vector keeps the backpack red and changes the background instead; adding only the offset turns the backpack blue, while the bench and background stay as they were.

to videos generated from starting seeds it never saw. A token-wise field and a shared vector are each fitted on twenty-four "a red backpack on a wooden bench" $\rightarrow$ "a blue backpack ..." pairs at layer 22 (20 denoising steps, guidance 5.0, 30 epochs, injected-field norm capped at 80), with the target for each pair the latent generated from the blue prompt at its starting noise. Object hue is read exactly as for the red-to-blue pairs above. On eight unseen seeds of the same bench prompt the token-wise field moves the object hue by a median of $-141.3°$ (all eight past $100°$) while the background hue shifts by only $23.4°$ and no frame loses the object; the shared vector moves the object by only $-4.0°$, keeping it red. Carrying a field across scenes is future work.

E.3 Wan fields share structure but do not reliably transfer

On Wan2.1-1.3B the fields optimised for different pairs share subspace structure. For each field, each $(2304, 1536)$, we take the subspace spanned by its top two right singular vectors. Over all 378 pairs of the 28 fields, the first principal angle between two fields' leading two-dimensional subspaces has cosine 0.736 at the median and 0.105 at its smallest, where two random two-dimensional subspaces of $\mathbb{R}^{1536}$ give a median of 0.045 and a 95th percentile of 0.076. Their leading subspaces are therefore substantially more aligned than expected for random fields.

Shared field structure does not imply field-level transfer. (Table 31) Subspace alignment does not imply that an individual field transfers across pairs. We took the field optimised on one pair and injected it on another, taking it from the pair whose requested change of launch direction is closest, the most favourable choice available, and measured the ball centre's distance to the target clip's over all 33 frames. We evaluate ten target pairs, six intervention conditions each. As a consistency check, the un-intervened generation reproduces the stored generation from the original optimisation run exactly. The borrowed field's error ranges from 1.8 pixels, as good as the pair's own field, to 27.3 pixels, no better than no intervention, so a field from another pair works on one pair and fails on the other nine. A field fitted jointly on the fitting set transfers to videos that took no part in the optimisation (Appendix E.1).

Distances are in pixels on a 256×256 frame, the median over the clip's 33 frames. The field from the nearest pair fails on nine of the ten pairs, ending as far from the target as no intervention or a norm-matched random field. A field from a randomly chosen pair is sometimes better than nothing but never near the pair's own field. The pair's own shared vector closes part of the gap on some pairs and never approaches the token-wise field.

pair	object	mask share	shared vector only (°)	offset only (°)	whole layer (°)
eligible and read					
p2	mug	0.63%	1.2	142.4	145.8
p7	bicycle	2.66%	30.4	114.7	120.7
p8	umbrella	12.14%	12.2	151.0	153.7
p10	backpack	10.89%	5.1	154.8	154.8
p11	balloon	10.57%	13.8	119.3	122.0
p12	apple	1.20%	1.3	161.6	161.0
p13	bell pepper	3.16%	1.7	153.3	158.7
p14	kite	0.08%	15.1	114.7	116.5
p15	sofa	12.28%	18.4	146.8	148.3
p17	toy car	0.69%	0.7	140.9	141.1
p18	notebook	32.23%	43.7	141.5	149.4
p20	baseball cap	12.38%	4.6	110.0	139.4
p22	kettle	6.33%	0.3	159.4	160.8
p23	wooden chair	1.81%	5.3	161.9	163.8
p24	hardcover book	19.25%	3.4	133.6	142.4
eligible; the intervention destroys the object, counted as a failure					
p9	car	5.52%	not reported		
not eligible					
p6	dress	5.31%	hue 64.1°		
p19	flower	0.56%	hue 43.9°		
p21	glove	1.65%	hue 29.3°, saturation 0.15		
p16	helmet	0.08%	304 pixels, hue 225.4°		

Table 28: All twenty red-to-blue prompt pairs on Wan2.1-1.3B, layer 22, and where each went. Eligibility is decided on the un-intervened clip alone: the object the prompt names has to be generated red there and be large enough to locate, and the last block gives the reading that fails it. Sixteen pairs are eligible. On p9 the intervention destroys the car, its mask falling from 22,045 pixels to 4,526 under the offset, so it counts as a failure of control and no angle is reported. For the fifteen that are read, the columns give how far the object's hue turns from the red clip when only the shared vector, only the offset, or the whole layer-22 output is replaced by the blue clip's; mask share is the object's fraction of the frame in the un-intervened clip. Medians over the fifteen are 5.1°, 142.4° and 148.3°. Counted per pair, the offset moves the hue by more than 100° on fifteen of the sixteen eligible pairs and the shared vector on none of them, its largest move being 43.7°. The two prompts' objects differ by a median of 148° of hue. Objects are named from the prompt.

		turn (°)		dist. (px)		ball px			lost frames	
clip	reference	token-wise	shared	token-wise	shared	none	token-wise	shared	token-wise	shared
0	93.6	93.7	−2.9	0.28	26.9	1733	1732	1518	0	0
1	83.0	84.9	−9.5	0.27	27.9	1961	1974	1765	0	0
2	90.2	86.3	+8.0	0.35	27.0	601	606	269	0	3
3	102.4	94.0	+4.2	0.23	20.7	1881	1869	1777	0	0
4	100.9	100.4	+5.3	0.30	21.7	1009	1016	290	0	5
5	85.1	86.3	+9.5	0.33	15.3	529	585	431	0	1

Table 29: Same clip, one number changed. The simulator raises the launch-angle parameter by 90° and renders a new clip; *reference turn* is the rotation our pixel measurement reads on that rendered clip over its first twelve frames. All measurements are scored against the rendered clip.

The shared-vector/offset asymmetry persists across injection depths. (Table 32) Every Wan intervention in this paper is applied at layer 22 of 30, which is where the fields were optimised. To check that the shared vector/offset asymmetry is not a property of that particular depth, we injected four stored fields, their shared vectors and their offsets at layers 18, 20, 22, 24 and 26, with no re-optimisation. Moving a field away from the layer it was optimised at is a transfer, so the whole field is expected to weaken off-centre; what the sweep tests is whether the shared vector and the offset keep their relative behaviour.

Medians over four pairs, in pixels on a 256×256 frame, measured as the distance from the target clip's ball centre. At every depth the shared vector's distance stays at the un-intervened value or

intervention	object hue (°)	background hue (°)	lost frames
token-wise field	$\mathbf{-141.3}$	23.4	0
shared vector	-4.0	11.7	0

Table 30: Transfer of a natural-prompt colour field to held-out seeds of the same prompt. A token-wise field and a shared vector are each fitted jointly on twenty-four "a red backpack on a wooden bench" → "a blue backpack . . . " prompt pairs (Wan2.1-1.3B, layer 22, 20 denoising steps, guidance 5.0, $480 \times 832 \times 33$, learning rate 10^{-3}, 30 epochs, injected field Frobenius norm capped at 80; the target for each pair is the latent generated from the blue prompt at the same starting noise). The fields are then added, without refitting, to eight videos generated from unseen starting seeds of the same bench prompt. Each entry is the median over the eight seeds of the hue move (intervened minus un-intervened, folded into $(-180, 180]$) inside the object mask and outside it; object hue is near 357° (red) un-intervened, and $-140°$ lands it near blue. The token-wise field turns the object blue on all eight seeds with no frame lost, moving the background by only 23.4° against the object's 141.3°, whereas the shared vector leaves the object red.

	no intervention	**own field**	**nearest pair**	random pair	random field	own shared vector
range over the ten pairs	12.0–27.3	**0.8–2.1**	1.8–27.3	8.5–23.9	12.6–27.3	10.0–22.5

Table 31: A field optimised on one pair, injected on another. Ball-centre distance to the target clip, in pixels, over the ten test pairs. *nearest pair* and *random pair* take the field from another pair, the nearest by requested rotation or one drawn at random; *random field* is a norm-matched random one.

layer	no intervention	whole field	shared vector only	offset only
18	18.84	10.64	19.03	10.90
20	18.84	8.99	19.21	9.07
22	18.84	**1.25**	19.26	**1.29**
24	18.84	8.55	19.13	8.84
26	18.84	11.91	19.09	12.07

Table 32: Injection-depth sweep of four stored Wan fields.

slightly worse, and the offset matches the whole field to within 0.3 pixels. Layer 22, where the fields were optimised, is strongest, and the distance grows monotonically on either side. The shared-vector/offset asymmetry persists across the tested injection depths.

E.4 The head-level dependence generalises to Wan

The head ranking on Wan. (Table 33) We rank all 360 heads on pairs 0 to 5 of the 28 matched pairs, using two scoring operations: zeroing a head's output, and replacing it with its output on the target clip, which is counterfactual replacement. The ranked group of each size is injected and compared against ten layer-matched random groups of the same size.

heads injected	of the 360	ranked by zeroing	ranked by replacing	ten random groups
8	2.2%	0.032	$0.294^{\dagger}$	-0.001 to 0.016
16	4.4%	0.025	$0.339^{\dagger}$	-0.003 to 0.054
40	11.1%	0.051	**0.977**	0.004 to 0.412
72	20.0%	—	0.981	0.033 to 0.919
all 360	100%	—	1.00	—

Table 33: The head ranking on Wan, in full.

Table 33 reports the median fraction of the requested rotation obtained over six pairs; the random column is the range over ten layer-matched groups of that size, none of which reaches the group ranked by replacement at any size; the group ranked by zeroing is overtaken at sixteen and forty heads. †At eight and sixteen heads one clip of the six loses the ball, so those two measurements in the replacement column are not interpretable; forty and seventy-two keep the ball in every frame of every pair (443–1727 and 483–1765 ball pixels). The two rankings are scored on the same quantity,

the launch angle read from the generated pixels, and differ only in the operation: zeroing a head puts the strongest of the 360 at $0.79°$, replacing it puts the strongest at $1.84°$ with a median of $0.15°$ over the 360 heads. Ranked by the operation the injection performs, forty of this model's heads (11.1%, against eight of seventy-two on the models we train) reproduce the trajectory, while no single head does.

Ranked on one set of clips, tested on held-out clips. On Wan2.1-1.3B the heads are ranked on one set of clips and the ranked groups, indices frozen, are injected on a disjoint set (pairs 0 to 5 for ranking, 6 to 11 held out). The 11.1% group carries the trajectory 0.980 of the way, against 0.977 on the ranking pairs, and a layer-matched random group of the same size does not reach it (0 of 10).

pinned heads	% of 360	field + pin top k (°)	random median (°)	random best (°)
none (field only)	–	90.70	–	–
top 10	2.8	**12.05**	57.77	15.45
top 20	5.6	9.75	47.67	11.30
top 30	8.3	6.15	38.40	9.05
top 40	11.1	3.25	40.65	8.70

Table 34: Pinning on Wan2.1-1.3B as a function of group size. Median rotation over the first twenty-four held-out clips (28 to 51) at a requested median of $90.8°$, holding the top-ranked k heads at their un-intervened outputs while the full-rank field of Table 27 is applied. The random columns give the median and the best (lowest) rotation over thirty layer-matched random k-head groups; the selected group ranks first among all thirty-one at every size.

Pinning on Wan. (Table 34) The head group is causal under pinning as well as replacement. Holding the top ten ranked heads at their un-intervened outputs while the field of Appendix E.1, fitted on all fitting clips at once, is applied drops the median rotation from $90.70°$ to $12.05°$ on the first twenty-four held-out clips (28 to 51), at a requested median of $90.8°$, and this group ranks first among thirty layer-matched random ten-head groups (median $57.77°$, best $15.45°$). Pinning ten of the 360 heads (2.8%) therefore removes most of the control. Pinning more heads removes more, down to $3.25°$ at forty heads, but the random groups also drop there, so the ten-head comparison separates most clearly. No condition loses a frame.