AJR: AMORTIZED JACOBIAN READOUTS FOR PROMPT-ADAPTIVE STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reading a language model's hidden state and predicting what will happen if we change it are different tasks. We introduce the Amortized Jacobian Readout (AJR), which learns from gradients to predict how a hidden-state edit changes the model's preference between two output tokens. Its prompt-dependent response direction both scores proposed edits and supplies a steering edit during the forward pass, without a backward pass at inference. Controlled color-retrieval experiments across four models from three families show why this distinction matters: learned decoders read unmodified states accurately but often mispredict intervention effects, whereas AJR learns their local response. In a fresh 640-prompt evaluation per model, steering with averaged linear and nonlinear AJR directions produces larger target-margin changes than fixed mean-gradient and contrastive-activation directions in every model. With the target answer supplied and edit size fixed from fitting data, it corrects 71–100% of initially wrong answers, while changing at most 2.7% of initially correct answers to wrong ones. Measured two-token generation latency is close to the unsteered model's; local-gradient steering takes 2.1–2.3 times as long. AJR thus turns learned intervention responses into prompt-adaptive control, within the depth and contrast on which it was fitted.

1 INTRODUCTION

Suppose a researcher wants to change a language model's answer by editing one of its hidden states. Which change should they try? An intermediate readout, or lens, translates that state into output-token scores. It can suggest which answer the state favors, but this does not tell us how the remaining layers will respond to an edit. Even a change that looks helpful in the readout may have little effect on the final answer, or move it the wrong way. Choosing an intervention therefore requires a prediction of its consequences, not just a reading of the state before it is changed.

We introduce the Amortized Jacobian Readout (AJR) for this purpose. During fitting, AJR learns from gradients that describe how the final output responds to small changes at a chosen layer. At evaluation time, it predicts this response from the current hidden state without a backward pass. This response can score candidate changes, but it also supplies an edit directly: move the state along the predicted response toward a supplied target. The direction adapts to the prompt rather than remaining fixed across inputs, and can be computed and applied before the remaining layers run.

The distinction is between two questions about the same hidden state. A decoder asks which output is associated with that state. An intervention predictor asks how the output changes when the state is perturbed and the remaining computation responds. We compare these questions directly: affine readouts trained by state regression or KL divergence decode the answer well, yet often mispredict the sign and size of intervention effects. Evaluating their complete normalized outputs does not close the gap. The local gradient, which measures the response around the current state, predicts these effects much more accurately.

The response also varies across prompts. A mean gradient fitted to the same data loses much of the local gradient's predictive value at intermediate depths. AJR therefore learns a state-conditioned response rather than a fixed direction. It represents gradients in a low-rank basis and predicts their coefficients from the source state. Linear and nonlinear predictors capture complementary responses; their equal-weight average supplies the direction used for steering.

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.

We evaluate this construction on controlled color-retrieval prompts in four models from three families. With a supplied target and an edit norm fixed from fitting data, AJR steers a fresh cohort in a single forward pass. It changes the target margin more than a fixed mean gradient and two contrastive-activation baselines in every model, correcting 71–100% of initially wrong answers with at most 2.7% harm to initially correct answers. Local-gradient steering produces stronger effects, but requires a backward pass followed by an edited forward run. Our evaluation reports both answer changes and inference cost, alongside changes to the full output distribution.

Paired interventions explain why the learned response is useful. Later computation often opposes both signs of an edit, rather than selectively correcting the harmful one. At mid depth, the lens direction is nearly orthogonal to the direction that changes the final contrast. This geometry replicates at 27B and motivates learning the response directly; the AJR prediction and steering studies use the four smaller models. Together, these experiments connect the decoding–response distinction to a concrete form of prompt-adaptive control.

Contributions. (1) AJR learns prompt-dependent response directions from gradients, enabling effect prediction and steering without a query-time backward pass. (2) Paired interventions and readout comparisons distinguish counteraction from correction, and accurate decoding from effect prediction. (3) A frozen-predictor evaluation demonstrates donor-free, single-pass steering on new entities, with fixed-vector comparisons, answer-level outcomes, and measured inference cost.

2 Related work

Computational self-repair. McGrath et al. (2023) study downstream compensation after ablation. Rushing & Nanda (2024) examine its variability across model families and distinguish normalization effects from neuron-level responses. Conditional Co-Ablation (Gong et al., 2026) identifies backup components through changes in their importance after other components are removed. We build on these observations by testing whether downstream counteraction also opposes changes that initially favor the correct answer.

Intermediate readouts and linearized attribution. The logit lens (nostalgebraist, 2020) reads intermediate residual states with the final unembedding; projecting a component's output onto the unembedding contrast is direct logit attribution (Elhage et al., 2021). The tuned lens (Belrose et al., 2023) fits an affine translator by KL divergence to the final distribution. Attribution patching (Nanda, 2023; Kramár et al., 2024) instead estimates patch effects from local gradients. Syed et al. (2024) apply it to circuit edges, and Hanna et al. (2024) use integrated gradients (Sundararajan et al., 2017) to improve circuit faithfulness. We compare decoders and gradient estimators against exact effects on the same perturbations (Section 4.2).

Jacobian-based readouts. Hernandez et al. (2024) use first-order maps to decode relations from subject representations. The Jacobian lens (J-lens) (Gurnee et al., 2026) averages downstream Jacobians over prompts and current-and-future token positions, then decodes states through that fixed map. Its causal construction also supports concept interventions. AJR instead predicts a prompt-dependent Jacobian projection for one output contrast. We compare it with an endpoint-matched mean-gradient readout using the same fitting prompts and gradient targets, isolating the benefit of conditioning on the current state rather than averaging across states.

Learning gradient information. Synthetic gradients predict backpropagated signals to decouple network training (Jaderberg et al., 2017); Sobolev training matches target values and derivatives (Czarnecki et al., 2017). AJR instead learns a contrast-specific response direction in a frozen model, using it to score and construct interventions rather than to update model weights.

Activation steering. Contrastive activation addition (CAA) constructs fixed directions from differences in activations (Rimsky et al., 2024). We adapt this baseline to the task's red/blue contrast and compare last-position and all-position edits. AJR instead predicts a response direction for each prompt from gradient supervision. Our comparison isolates prompt adaptation on a fixed token contrast within this controlled task.

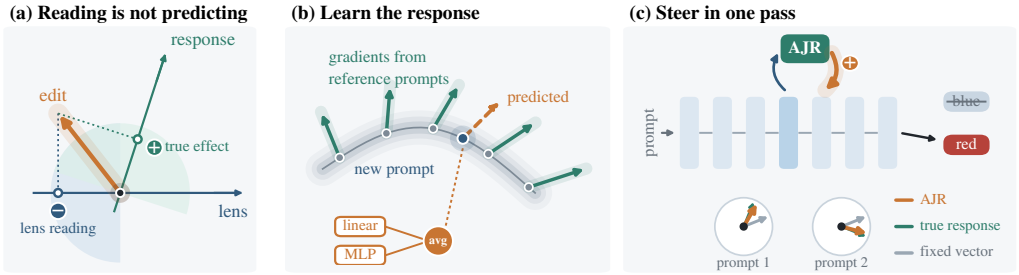


Figure 1: **AJR overview.** Schematic: (a) Lens readings and first-order effects can disagree. (b) Gradients train linear and nonlinear response predictors, which are averaged. (c) The averaged response sets a per-prompt steering direction at fixed norm, unlike a fixed vector.

3 Measuring and Learning Intervention Effects

Figure 1 outlines AJR: obtain local response targets during fitting, predict them from the source state, and use the predicted direction to construct an edit inside the forward pass. We define the response target first, then the predictor and its steering rule.

3.1 The Local Intervention Response

Let $h \in \mathbb{R}^m$ be the unperturbed final residual state at the last prompt position, and let $d \in \mathbb{R}^m$ be an injected change to the residual stream at a source block. We use a fixed two-token contrast. For RMS-normalized models, define

$$r(h) = \sqrt{\frac{1}{m}\sum_{j=1}^{m} h_j^2 + \epsilon}, \qquad v = (u_{\text{red}} - u_{\text{blue}}) \odot g, \tag{1}$$

where u_{red} and u_{blue} are the corresponding unembedding rows, g is the final normalization weight, and $\odot$ denotes elementwise multiplication; for LayerNorm, r is the standard deviation and v is centered (Appendix A). The final margin is $v^\top h/r(h)$ plus a normalization bias, if present, which cancels in margin changes. For Gemma-2, this expression is the margin before the final logit softcap; answer choices and output-distribution KL use the model's actual post-softcap outputs. To isolate the downstream response, hold the denominator at its unperturbed value. If h_{free} is the state after allowing downstream modules to respond, define

$$D = \frac{v^\top d}{r(h)}, \qquad F = \frac{v^\top (h_{\text{free}} - h)}{r(h)}, \qquad W = F - D. \tag{2}$$

D is a direct effect in the fixed readout; F includes downstream responses; W isolates their net contribution in that readout. We separately retain actual normalized margins and task predictions.

First-order prediction and the effective readout. Let $u = J^\top v$ be the exact gradient of the final contrast $v^\top h_{\text{free}}$ with respect to the source injection, where J is the Jacobian of the downstream blocks on the unperturbed run, computed by backpropagation. The first-order prediction is $F_{\text{lin}} = u^\top d/r(h)$, and u is the effective readout direction of the source block: it is the direction in which a small perturbation changes the fixed-scale contrast most. The lens reads the same perturbation as $D = v^\top d/r(h)$. To first order, a perturbation along the lens direction has $F/D = u^\top v/\|v\|^2$, and the sign of D predicts the sign of F only to the extent that u and v are aligned. We report $\cos(u, v)$ per depth. Oddness alone does not imply linearity, since odd cubic terms are also antisymmetric, so we report the amplitude error of F_{lin} (root-mean-square error relative to the root-mean-square of F) alongside its rank correlation, and we sweep the perturbation magnitude.

Separating response from correction. A direct-only control clamps downstream component outputs to their clean values, leaving final state $h + d$. We also evaluate d and $-d$ on the same prompt. Write their effects as F_+ and F_- and define

$$F_{\text{odd}} = \frac{F_+ - F_-}{2}, \qquad A = \frac{F_+ + F_-}{2}. \tag{3}$$

The odd part reverses with the edit, while the even part is shared by both signs. We summarize antisymmetry by $\mathbb{E}[F_{\text{odd}}^2]/(\mathbb{E}[F_{\text{odd}}^2] + \mathbb{E}[A^2])$. Opposition to both signs can arise from a linear map and does not by itself show selective correction. Ground truth orients this diagnostic so that $D_+ \geq 0$; steering instead uses a target supplied by the user. We measure A separately on initially correct and initially wrong answers: a shift that helps the latter can also reflect reduced confidence in whichever answer the model prefers. The local linear map itself can depend on the prompt, so an odd response does not imply that the computation ignores evidence. Appendix E gives the paired-response controls.

3.2 Amortized Jacobian Readout

AJR learns how the response direction $u(x)$ depends on the clean residual state x after the source block. For a fixed contrast and source depth, let $\bar{u}$ be the mean fitting gradient and let the columns of E be the leading principal directions of variation around that mean. We predict

$$\hat{u}(x) = \bar{u} + Ec_\theta(x), \qquad \hat{F}(x, d) = \frac{\hat{u}(x)^\top d}{r(h)}. \tag{4}$$

Here $\bar{u}$ is the shared response and $Ec_\theta(x)$ its prompt-dependent adjustment. The basis captures variation in gradient targets rather than source states; the coefficients adapt that basis to each prompt. AJR predicts only the Jacobian projection needed for the chosen contrast, not a full Jacobian.

Fitting the response. For fitting prompts $i = 1, \ldots, n$, collect source states x_i and gradient targets $u_i = J_i^\top v$ from clean runs. Center the targets at $\bar{u}$ and use their leading K principal directions to form $E \in \mathbb{R}^{m \times K}$. AJR-lin fits their coefficients from centered source states:

$$B_* = \arg\min_B \sum_i \left\| B^\top (x_i - \bar{x}) - E^\top (u_i - \bar{u}) \right\|_2^2 + \lambda \|B\|_F^2, \qquad c_\theta(x) = B_*^\top (x - \bar{x}). \tag{5}$$

We choose λ from multiples of the mean diagonal of the feature Gram matrix. The mean-gradient control sets $K = 0$, retaining the same training targets but removing the dependence on x. This separates learning the shared response from learning its prompt-specific variation.

AJR-mlp replaces the linear coefficients with a two-layer network: standardized source states, a 512-unit hidden layer with GELU activation (Hendrycks & Gimpel, 2016), and K output coefficients. It uses the same gradient-basis construction and minimizes squared error on projected first-order effects $u_i^\top d/r(h_i)$, with weight regularization. Thus the nonlinear fit emphasizes gradient accuracy along the fitting perturbations rather than equally across all state directions.

Combining readouts. AJR-ens scores a candidate by $\hat{F}_{\text{ens}} = (\hat{F}_{\text{lin}} + \hat{F}_{\text{mlp}})/2$. It averages the two fitted response directions, using no additional fitting or learned combination weights. Both readouts must be stored and evaluated.

From a response to a steering edit. For target sign $s \in \{-1, +1\}$ and fixed norm a, use

$$d_{\text{AJR}}(x) = a \, \frac{s\hat{u}_{\text{ens}}(x)}{\|\hat{u}_{\text{ens}}(x)\|_2}. \tag{6}$$

This maximizes the predicted first-order target effect among edits of norm a. The clean-output denominator is a positive scalar and cancels from this choice. A hook reads x, predicts the direction, and adds the edit to the last prompt position before downstream computation continues. Neither a completed clean run nor a donor prompt is needed. This optimizes the learned local prediction, not necessarily the model's finite effect.

What makes a useful steering direction. For the first-order target, the ideal norm-a edit points along su. The predicted edit achieves the fraction

$$\frac{su^\top d_{\text{AJR}}}{a\|u\|_2} = \cos(u, \hat{u}_{\text{ens}}) \tag{7}$$

of that first-order effect, when both directions are nonzero. Thus steering depends on angular accuracy, not the predicted gradient's overall scale. This also explains why a strong decoder need not give a strong steering direction: predicting the answer at x does not determine the response to a displacement around x. For example, an affine decoder $Ax + b$ assigns the same unnormalized

response direction $A^\top v$ to every prompt, while the model's downstream Jacobian can change with the prompt. Normalizing the decoder introduces state dependence, but matching its outputs still does not directly constrain those derivatives. AJR instead fits the response targets themselves. Finite edits can additionally change the normalization and depart from the linear approximation; we evaluate their actual margin and generated answer rather than infer success from alignment.

Validation and fitting. Entity pairs are held out within the fitting cohort to choose rank and regularization. The linear prediction diagnostic selects by gradient error. For steering, both constituent predictors are selected by error against exact finite effects on the held-out pairs, then averaged with equal weights. This keeps model selection separate from all reported steering prompts. Appendix B specifies the two fitting objectives, grids, and selected configurations.

What is amortized. Gradient supervision is collected during fitting and reused on new prompts at the same depth and contrast. Steering evaluates the fitted predictor inside a single forward pass, avoiding a gradient computation for that prompt. The fitting cost is an offline investment, not part of the reported inference latency. Candidate scoring is another use of the same direction, but a single attribution-patching gradient also scores many candidates. Our practical evaluation therefore concerns constructing an edit directly, rather than claiming faster search over a few candidate edits.

4 EXPERIMENTS

We ask three questions: can AJR steer a fresh prompt without a backward pass; does it predict intervention effects better than a decoder or fixed response; and what response geometry explains the difference? The primary models are Qwen1.5-0.5B and Qwen1.5-1.8B (Qwen Team, 2024), Pythia-410M (Biderman et al., 2023), and Gemma-2-2B (Gemma Team, 2024). All meet the task thresholds of restricted color accuracy $\geq .60$ and two-color probability mass $\geq .20$.

The task describes two people's key, coat, or hat colors and asks for one person's color. Minimally edited donors change either the queried fact or the other person's fact. Their difference in last-position MLP contributions supplies relevant or irrelevant perturbations. AJR fitting uses 960 prompts; the response comparison uses 384 prompts from sixteen held-out entity pairs. Readout comparisons reuse those response interventions. Steering uses a separate 640-prompt cohort per model, with sixteen further entity pairs and both fitted and unseen templates. Predictors, edit norms, and primary steering comparisons are fixed before collecting that cohort. Intervals resample entity pairs rather than individual prompts, retaining within-pair dependence. Appendix A specifies the prompt grammar and disjoint evaluation groups.

For example, "Charlotte has a red key. William has a blue key." is followed by "Charlotte has a", whose desired continuation is red. Changing colors, fact order, and object templates creates different contexts with the same answer format. Every steering method receives the desired color explicitly; the question is how effectively its edit moves the model toward that target. This separates learning a prompt-dependent response from choosing which answer should be preferred.

4.1 PROMPT-ADAPTIVE STEERING IN A SINGLE PASS

We construct a steering edit directly from the frozen AJR-ens direction. Each model receives 640 prompts from sixteen new entity pairs, with three fitted and two unseen templates. The target color is supplied. The edit norm is the median matched-donor norm in the fitting data, fixed per model. AJR predicts and adds the edit at the mid-depth block output during the same forward pass; no donor or backward pass is used at inference.

We compare the fitting-cohort mean gradient, a contrastive activation direction (CAA) from mean red-minus-blue source states, and the exact local gradient. CAA-last edits the last prompt position; CAA-all applies the same vector to every prompt and generated position. These are task-adapted CAA baselines (Rimsky et al., 2024). All use the same per-position edit norm and target orientation. CAA-all therefore changes more positions, rather than sharing the last-position methods' total edit energy. Local-gradient steering uses a gradient run followed by an edited run. Table 1 reports changes in the normalized target margin (pre-softcap for Gemma) and actual greedy answers. These outcomes differ from the fixed-denominator effects used for response prediction.

Table 1: **Single-pass steering.** Fixed fitting-derived norm; 640 fresh prompts. (a) Margin gain. (b) Correction / harm (%). **Bold values**: best backward-free gain or correction. †: gradient reference, requiring backpropagation and an edited rerun.

Direction	Qwen1.5-0.5B	Qwen1.5-1.8B	Pythia-410M	Gemma-2-2B
(a) Target-margin change ↑				
Mean gradient	0.76	1.66	1.47	1.66
CAA-last	0.15	0.13	0.10	0.04
CAA-all	0.12	0.25	0.11	-0.09
AJR-ens (ours)	**1.80**	**3.37**	**2.16**	**4.35**
Local gradient†	2.38	4.41	2.53	6.70
(b) Correction ↑ / harm ↓				
Mean gradient	53.2 / 2.5	89.3 / 1.3	78.2 / 0.0	63.5 / 1.0
CAA-last	14.3 / 0.0	20.7 / 1.0	3.0 / 1.1	11.5 / 0.3
CAA-all	15.1 / 0.4	31.4 / 0.6	4.2 / 1.5	9.6 / 1.0
AJR-ens (ours)	**71.4** / 0.2	**93.4** / 0.0	**92.7** / 0.0	**100.0** / 2.7
Local gradient†	84.1 / 0.6	100.0 / 0.0	93.9 / 0.0	100.0 / 0.0

AJR exceeds all three fixed-vector baselines on the pre-specified margin endpoint in every model; all twelve paired entity-group intervals are above zero (Appendix D). It corrects 71–100% of initially wrong answers, with harm at most 2.7%. Correction and harm condition on initially wrong and correct answers, respectively. Among generations with the target answer, the following token is the expected object noun at least 99.7% of the time across the tested edit sizes. This two-token evaluation measures local continuation consistency. Margin change is the primary comparison; correction, harm, and continuation describe the resulting behavior of the edited model.

Inference cost and distributional change. Batch-32 two-token generation has similar measured latency with and without AJR: 151.7 versus 151.6 ms in Qwen1.5-1.8B. Local-gradient steering takes 2.1–2.3 times as long across models, while producing stronger target effects. These shared-GPU medians include readout evaluation and edit application, but not fitting cost. At matched full-output KL on a coarse edit-size grid, the relative advantage depends on the model; mean-gradient steering is better in Qwen1.5-1.8B. Appendix D gives timings and the comparison.

4.2 Does decoding a state predict what changing it does?

We compare readouts and attribution estimators on the same saved perturbations against exact downstream effects (Table 2). The logit lens, or direct logit attribution, predicts D. Two affine translators use the 192 development prompts: ridge regression to the final residual state, and a tuned lens trained by KL divergence through the model's final norm and unembedding (Belrose et al., 2023). Their fixed-scale prediction is $v^\top Ad/r(Ax+b)$, using the translated clean state's scale to remove the scale ambiguity of KL training. This is a fixed-scale projection, not the finite difference of the complete normalized decoder; we evaluate that quantity separately below. Attribution patching uses the clean-state gradient. The mean-gradient baseline uses AJR's fitting prompts and gradient targets. It is an endpoint-matched adaptation, not the full original J-lens with its position-averaged vocabulary readout (Gurnee et al., 2026). Appendix C details decoder fitting and normalization.

Better decoding does not ensure better effect prediction. The learned readouts decode unmodified states well. On the task distribution, the tuned lens reduces held-out KL to the final distribution from 4–14 nats for the raw lens to .04–.23. Its predicted answer margin tracks the actual final margin with Spearman .82–.95; the ridge translator gives .86–.91 and the raw lens .08–.76.

Yet their intervention predictions are much less accurate. Sign agreement with the exact effect is .51–.68 for the tuned lens and .54–.65 for the ridge translator, Spearman −.03–.26 and .07–.52, and the relative amplitude error at each translator's own scale exceeds one in every model. Attribution patching predicts the same effects with sign agreement .93–.96 and Spearman .94–.99. Integrated gradients provide a numerical reference with relative error below .02; Appendix C reports the numerical agreement and evaluation cost. Pythia-410M retains more information in the ridge projection, consistent with its higher $\cos(u,v)$, but remains below attribution patching.

Table 2: **State decoding versus intervention prediction.** Mid-depth, matched-norm relevant donors; intervention target: fixed-readout effect F. **Bold**: best intervention metrics per model among readouts without evaluation-time backpropagation. Fitting budgets and supervision differ (Section 4.3).

		Decoding	Intervention prediction		
Model	Readout	Spearman ↑	Sign ↑	Spearman ↑	Rel. error ↓
Readouts without evaluation-time backpropagation					
Qwen1.5-0.5B	logit lens (DLA) (nostalgebraist, 2020; Elhage et al., 2021)	.764	.42	−.171	1.08
	ridge translator	.859	.58	.165	1.13
	tuned lens, KL-trained (Belrose et al., 2023)	.892	.57	.144	3.11
	mean-gradient (J-lens-style) (Gurnee et al., 2026)	–	.38	−.149	1.09
	AJR-lin (ours)	–	**.78**	**.743**	**.66**
Qwen1.5-1.8B	logit lens (DLA) (nostalgebraist, 2020; Elhage et al., 2021)	.284	.28	−.167	1.13
	ridge translator	.892	.57	.118	1.60
	tuned lens, KL-trained (Belrose et al., 2023)	.910	.68	.262	2.03
	mean-gradient (J-lens-style) (Gurnee et al., 2026)	–	.51	−.163	1.00
	AJR-lin (ours)	–	**.85**	**.825**	**.56**
Pythia-410M	logit lens (DLA) (nostalgebraist, 2020; Elhage et al., 2021)	.469	.57	.487	.93
	ridge translator	.910	.65	.522	1.53
	tuned lens, KL-trained (Belrose et al., 2023)	.950	.54	.108	1.24
	mean-gradient (J-lens-style) (Gurnee et al., 2026)	–	.76	.619	.84
	AJR-lin (ours)	–	**.93**	**.910**	**.46**
Gemma-2-2B	logit lens (DLA) (nostalgebraist, 2020; Elhage et al., 2021)	.082	.53	−.124	1.23
	ridge translator	.863	.54	.068	6.19
	tuned lens, KL-trained (Belrose et al., 2023)	.822	.51	−.032	4.61
	mean-gradient (J-lens-style) (Gurnee et al., 2026)	–	.48	−.082	1.01
	AJR-lin (ours)	–	**.90**	**.926**	**.37**
Gradient-based reference (evaluation-time backpropagation)					
Qwen1.5-0.5B	attribution patching (Nanda, 2023; Kramár et al., 2024)	–	.94	.979	.22
Qwen1.5-1.8B	attribution patching (Nanda, 2023; Kramár et al., 2024)	–	.93	.955	.30
Pythia-410M	attribution patching (Nanda, 2023; Kramár et al., 2024)	–	.94	.943	.34
Gemma-2-2B	attribution patching (Nanda, 2023; Kramár et al., 2024)	–	.96	.991	.14

The gap persists for complete readouts. We also compare the change in each translator's complete normalized output, $\mathrm{margin}(A(x+d)+b) - \mathrm{margin}(Ax+b)$, with the model's actual normalized margin change. Ridge and tuned-lens correlations remain low, whereas AJR's nonlinear variant reaches Spearman .82–.94 and attribution patching .94–.99 (Appendix B.1). Thus the gap between decoding and effect prediction is not confined to fixed-scale projections.

4.3 AJR: LEARNING THE RESPONSE RATHER THAN THE DECODED OUTPUT

The response direction changes with the prompt, but most of that variation lies in a small subspace: nine principal directions carry 80% of its variance in the Qwen1.5 models and two to four in the others. We fit the linear AJR of Equation 4 on 960 prompts, including 32 further unused entity pairs, with rank and ridge strength chosen on held-out pairs (Appendix B). Table 2 reports the result at the mid-depth source: sign agreement .78–.93 and Spearman .74–.93 against .51–.68 and −.03–.26 for the tuned lens, with relative error .37–.66. Prediction is best on all three intervention metrics among the evaluated readouts without evaluation-time backpropagation, in each of the four models. Performance generally improves at later source depths (Appendix B). The mean-gradient control gives Spearman −.16–.62, below AJR in every model. The depth profile clarifies why conditioning matters: mean-to-local gradient alignment is lower at mid depth than at the deepest source, where the mean gradient reaches Spearman .81–.96 (Appendix C.1). The decoders are not simply learning the average response either: at mid depth, their directions have low cosine with the mean gradient (.04–.25) and local gradients (.01–.11 on average). AJR uses gradient supervision and a larger prompt pool than the decoders; the matched mean-gradient control uses the same fitting states and gradient targets, isolating the value of state conditioning.

Estimation and curvature. We compare linear and two-layer coefficient predictors, selecting each on held-out fitting pairs. The nonlinear variant has the lowest held-out effect error in all four models and reaches Spearman .84–.93 on evaluation prompts. An error decomposition identifies gradient prediction as the larger error term at most depths in Qwen1.5 and Gemma-2; curvature matters more at two Pythia depths. The fitting details and full decomposition are in Appendix B.1.

Table 3: **Counteraction and response geometry.** Mid-depth, matched-norm relevant donors; brackets: 95% entity-pair bootstrap intervals. Effects point toward the correct answer. Additional diagnostics are in Table 4.

	Qwen1.5 0.5B	Qwen1.5 1.8B	Pythia 410M	Gemma-2 2B	Qwen3.6 27B
First-order response					
Odd share	.956	.932	.901	.985	.884
Gradient correlation	.979	.955	.943	.991	.936
Gradient relative error	.220	.304	.342	.136	.359
Opposition under both signs					
Directly helpful: W_+	$-.053$ $[-.060, -.046]$	$-.180$ $[-.198, -.162]$	$-.061$ $[-.075, -.047]$	$-.108$ $[-.114, -.103]$	$-.041$ $[-.044, -.038]$
Directly harmful: W_-	$+.031$ $[.022, .039]$	$+.122$ $[.107, .137]$	$+.085$ $[.072, .098]$	$+.110$ $[.104, .116]$	$+.033$ $[.030, .036]$
Lens–response mismatch					
Direct–final sign match	.42	.28	.57	.53	.49
Lens–gradient cosine	.08	.07	.17	.04	.01

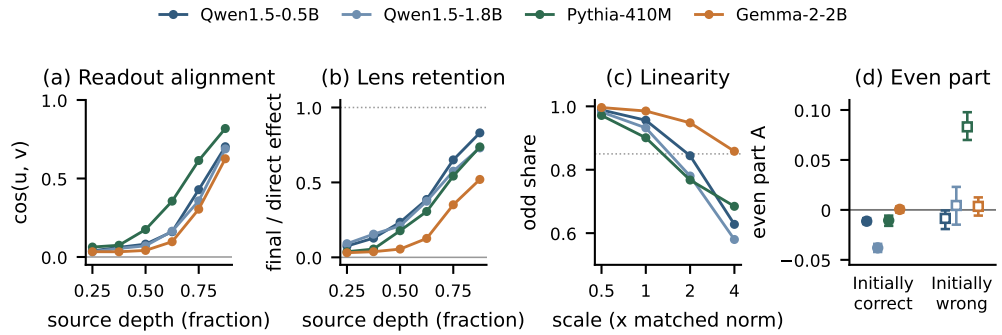


Figure 2: **Response geometry.** (a) Readout alignment $\cos(u, v)$ and (b) lens retention across depth. (c) Odd share versus magnitude (dotted: .85). (d) Even response by initial correctness, with 95% entity-pair intervals. Panels (c–d): mid-depth relevant donors.

4.4 Response geometry across depths and magnitudes

Later computation could undo a perturbation, shift both signs toward the correct answer, or transform the injected change without doing either. We use the decomposition of Section 3 to distinguish these responses on 384 confirmation prompts (16 unused entity pairs, three templates) in four models from three families: Qwen1.5-0.5B, Qwen1.5-1.8B (Qwen Team, 2024), Pythia-410M (Biderman et al., 2023), and Gemma-2-2B (Gemma Team, 2024). All meet fixed task-eligibility thresholds (restricted accuracy $\geq .60$ and two-color probability mass $\geq .20$). Perturbations are injected at six source depths (fractions .25 to .875). Five directions are scaled to each depth's matched donor norm: relevant and irrelevant donor deltas, the lens direction v, the gradient direction u, and a random direction. Both signs are collected for every direction, and the relevant and lens directions at .5, 1, 2, and 4 times the matched norm. The main comparison uses relevant donors at the matched norm and the mid-depth source, chosen before confirmation. We report 95% entity-pair bootstrap intervals, with other directions in Appendix E and magnitude sweeps in Figure 2c. Table 3 also includes a Qwen3.6-27B response replication; Figure 2 shows the four primary models. Appendix A gives the shared protocol and the 27B implementation. Gradient rank correlation is Spearman(F_{lin}, F), and amplitude error is relative RMSE. Sign agreement compares D with F; lens–gradient alignment is $\cos(u, v)$.

Later layers oppose both signs. In all five models the directly helpful sign receives a negative downstream contribution and the directly harmful sign a positive one, with intervals excluding zero. Thus opposition to directly harmful changes is accompanied by opposition to directly helpful changes across all three model families. The paired control therefore distinguishes counteraction from evidence of a truth-selective correction process.

Antisymmetry and first-order prediction. In the four primary models, the odd part carries 90–99% of the mean-square response, and the exact gradient predicts the exact patching effect with Spearman correlation .94–.99 and relative amplitude error .14–.34 in every model. At larger perturbations (Figure 2c), the local approximation weakens: at four times the matched norm the odd share is .58–.86 and the amplitude error .58–1.55. The matched-norm effects are therefore well ranked by the local gradient, with measurable curvature as the perturbation grows. The high-gain gradient direction shows the largest amplitude errors (Appendix E). The Qwen3.6-27B replication also shows opposition under both signs and a predominantly odd, well-ranked response (Appendix A).

Vector restoration and transformation. Gemma-2-2B comes closest to vector restoration: the downstream change has cosine $-.80$ with d and the final state moves by $0.72\,\|d\|$. The Qwen1.5 models instead move the final state by two to three times $\|d\|$ with weak anti-alignment. The downstream computation transforms the perturbation; it does not simply undo it.

Why the immediate reading is misleading. For the primary models, $\cos(u, v)$ is .04–.17 at mid depth (Figure 2a): the direction that changes the fixed-scale contrast is nearly orthogonal to the direction the lens reads. Consequently a perturbation along the lens direction loses 77–94% of its direct effect (Figure 2b), and the immediate sign of a donor perturbation is a poor guide to its total sign. Qwen1.5-1.8B reverses the sign systematically (agreement .28), whereas Pythia-410M retains positive effect correlation (Spearman .49). This variation is visible even though all models show opposition in the fixed readout.

Irrelevant donors and random directions of the same norm receive opposing contributions of similar size (Appendix E), so the counteraction is not specific to evidence-bearing directions. The gradient direction, by contrast, has a small direct effect but a final effect 27 times larger in three models: the downstream map has high gain in directions that the lens does not read.

A shared shift need not favor the correct answer. For the Qwen1.5 models and Gemma-2-2B, the even shift on initially wrong prompts is negative or has an interval containing zero. Pythia-410M has a positive shift there ($A = +.083$), but a negative shift on initially correct prompts ($A = -.011$). Its stratum means thus move away from the model's preferred answer in both groups, consistent with confidence reduction. A positive even shift on wrong answers alone would miss this distinction: reduced confidence can help when the preferred answer is wrong and hurt when it is correct.

5 Discussion

From reading to control. Accurate decoding and accurate intervention prediction need not coincide. AJR learns the prompt-dependent response, using the same readout to score edits and construct steering directions. Its advantage over the matched mean-gradient control isolates prompt adaptation: at a fixed layer, two prompts with the same target can receive different steering directions.

Scope. Steering gains are demonstrated on synthetic color retrieval with a supplied target; a natural-text pronoun test does not reproduce the advantage over fixed vectors (Appendix D.1). Fits require gradient supervision for each depth and contrast; the 27B replication tests response geometry, not AJR steering. The useful deployment regime is repeated steering within a fitted prompt family, where avoiding an online backward pass matters. Extending the predictor to less structured text remains a challenge for broader applications.

6 Conclusion

AJR learns how a hidden-state edit affects a model's output. Paired interventions and readout comparisons show why this differs from decoding the state. Gradient supervision is collected during fitting; the learned response is then reused on new prompts without query-time backpropagation. With a supplied target and fitting-derived edit size, AJR steers new prompts in a single forward pass, outperforming fixed directions on the tested margin objective. The language model's parameters remain fixed: adaptation occurs in the predicted edit, which can differ across prompts even when the desired token is the same. Learning the local response thus connects intermediate-state analysis to direct control of a model's answer.

AI USE STATEMENT

Generative AI models assisted the development of this study of learned intervention responses. Assistance covered research ideas and refinements, mathematical derivations, experimental design and implementation, result analysis, literature searches, manuscript drafting and revision, and preparation of figures and references. The authors set the research questions and methodology and reviewed the claims, derivations, experiments, code, and reported results, including AI-assisted material. They approved the final manuscript and take responsibility for its content.

REPRODUCIBILITY STATEMENT

Section 3 specifies the response targets, gradient-supervised fit, and steering edits; Section 4 defines the comparisons and evaluation measures. Appendix A gives prompt construction, data separation, and perturbation settings. Fitting objectives and hyperparameters are in Appendices C, B, and B.1; Appendix D specifies steering strength, placement, baselines, and inference timing. The natural-text steering test and its fitting protocol are in Appendix D.1. The code supplement includes experiment scripts, saved summaries, table generators, and tests that check reported values against saved results. Model weights and external text are obtained from their cited sources.

ETHICS STATEMENT

AJR steers toward a supplied target; it does not determine whether that target is desirable or true. The steering evaluation reports correction of initially wrong answers alongside harm to initially correct answers, continuation consistency, and changes to the output distribution. These measurements characterize targeted edits on the tested prompts, not general-purpose control of model behavior. Downstream uses should assess the target and unintended changes in their intended setting. Models and text data retain their original licensing and access requirements.

REFERENCES

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023.

Wojciech M. Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Zhiren Gong, Zihao Zeng, Chau Yuen, and Wei Yang Bryan Lim. Conditional co-ablation: Recovering self-repair backups in transformer circuits. *arXiv preprint arXiv:2607.01940*, 2026.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, T. Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. Transformer Circuits Thread, July 2026.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1627–1635. PMLR, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. The Hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Neel Nanda. Attribution patching: Activation patching at industrial scale. Neel Nanda's blog, 2023.

nostalgebraist. Interpreting GPT: the logit lens. LessWrong post, 2020.

Qwen Team. Introducing Qwen1.5. Qwen Blog, February 2024.

Qwen Team. Qwen3.6-27B. Hugging Face model card, 2026.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Cody Rushing and Neel Nanda. Explorations of self-repair in language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42836–42855. PMLR, 2024.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416. Association for Computational Linguistics, 2024.

Table 4: **Additional response diagnostics.** Same cohort and perturbations as Table 3; brackets: 95% entity-pair bootstrap intervals. Effects point toward the correct answer.

	Qwen1.5 0.5B	Qwen1.5 1.8B	Pythia 410M	Gemma-2 2B	Qwen3.6 27B
Capability					
Restricted accuracy	.919	.786	.760	.859	.786
Two-color mass	.676	.653	.566	.552	.633
Even response					
A, initially correct	$-.011$	$-.038$	$-.011$	$+.001$	$-.005$
	$[-.014, -.008]$	$[-.042, -.034]$	$[-.016, -.006]$	$[-.003, .003]$	$[-.007, -.004]$
A, initially wrong	$-.009$	$+.004$	$+.083$	$+.004$	$+.001$
	$[-.019, -.001]$	$[-.015, .023]$	$[.070, .098]$	$[-.006, .013]$	$[-.002, .003]$
Mean absolute odd part	.083	.183	.168	.171	.029
Readout and state geometry					
Direct–final correlation	$-.17$	$-.17$	.49	$-.12$	.03
$\|\|h_{\text{free}} - h\|\|/\|\|d\|\|$	2.06	3.02	1.07	.72	1.56
$\cos(h_{\text{free}} - h - d,\, d)$	$-.37$	$-.29$	$-.60$	$-.80$	$-.56$
Lens retention F/D	.23	.21	.18	.06	.01

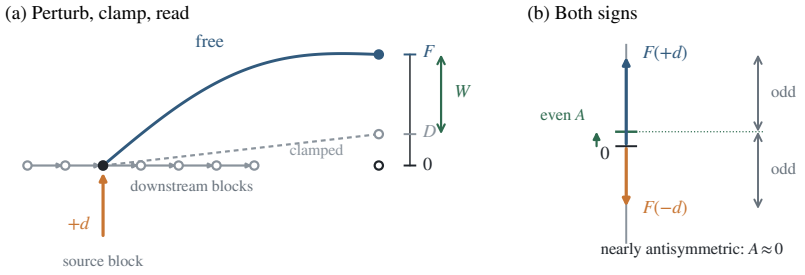


Figure 3: **Perturbation diagnostics.** Schematic: (a) Clamped and free runs separate direct effect D, total effect F, and downstream contribution $W = F - D$, relative to the unperturbed run. (b) Opposite perturbations separate the odd response from the shared even shift A.

A Response setup, endpoints, and shared statistics

Table 4 complements Table 3 with task accuracy, even responses, state geometry, and lens retention, without repeating its metrics.

Prompts and data separation. A typical prompt is "Charlotte has a red key. William has a blue key." followed by a newline and "Charlotte has a". The other fitted noun–verb pairs are coat/wears and hat/owns. Each entity pair appears with both target colors, matching or conflicting distractor colors, and both fact orders, giving 24 prompts per pair. Eight development pairs supply 192 prompts; sixteen disjoint response-evaluation pairs supply 384. AJR adds 32 fitting pairs to the development set, for 960 prompts in total. Steering uses sixteen further pairs and also includes bag/carries and cup/holds, giving 640 prompts. No evaluation entity is used to fit or select its predictor.

Figure 3 illustrates the clamp and polarity controls.

Readouts. For the RMS-normalized Qwen1.5 models, v is the contrast of unembedding rows scaled by the final norm weight; Gemma-2 scales by $1 + g$ and soft-caps its logits. Color-retrieval margins are measured before the cap; answer choices and distributional KL use the post-softcap output. Pythia uses LayerNorm: the denominator is the standard deviation of h, and v is the weighted contrast with its mean removed, because the centering step annihilates the mean component of any perturbation. Reconstructing each branch's two-token logits from the saved final state agrees with the model output to within 6×10^{-5}, after inverting Gemma's softcap for this comparison.

Source and directions. The source is the tensor added to the residual stream by the MLP branch of the source block at the last position (for Gemma-2, the post-feedforward norm output). Depths are fractions .25, .375, .5, .625, .75, .875 of the block count: blocks 6, 9, 12, 15, 18, 21 of 24 and the

analogous rounding for Gemma-2's 26 blocks. The matched norm at each depth is the smaller of the relevant and irrelevant natural donor deltas for that prompt. The random direction is Gaussian and seeded by prompt and depth. Every direction is oriented so that its direct effect on the correct-answer contrast is nonnegative. Direct effects are computed analytically; clamped runs agree with $h + d$ to within 7×10^{-5} at every depth.

Evaluation and uncertainty. The primary comparison uses relevant donors at scale 1 and mid depth (fraction .5), selected before evaluation. Tables 3 and 4 report odd share, first-order prediction error, and the contributions under both signs. Intervals resample entity-pair means, computed within each reported correctness stratum. This keeps color, order, and template variants together within a pair. We use 2,000 bootstrap draws and report pointwise 95% intervals. This grouping accounts for dependence among prompts built from the same pair.

Answer orientation. We report the even part in both initially correct and initially wrong strata. A shift away from the model's preferred answer has negative A in the first stratum and positive A in the second. Together these signs distinguish confidence reduction from a shift toward the correct answer.

27B replication. Qwen3.6-27B (Qwen Team, 2026) uses 64 blocks, hidden size 5,120, and hybrid gated-delta-net and attention blocks. We apply the same confirmation cohort, directions, scales, and evaluation criteria in float32 on one GPU. For a linear-attention block, the attention-branch site is its gated-delta-net output; residual injection and clamping otherwise remain unchanged. The all-clamped identity error is 2×10^{-6}, and backward passes use sub-batches of eight prompts. Table 4 reports the mid-depth source (block 32). Lens–gradient alignment rises from .01 there to .34 at the deepest source; a mid-depth lens-direction perturbation retains 1% of its direct effect.

B AJR FITTING, MODEL SELECTION, AND QUERY COST

Amortized Jacobian Readout (AJR). The fitting cohort combines 768 prompts from 32 unused entity pairs and three templates with the 192 development prompts. We collect source states at the last position and six depths, together with exact u from one backward pass per batch. The model is $\hat{u}(x) = \bar{u} + \sum_{k \le K} c_k(x) e_k$ with e_k the top-K principal directions of the fitting targets and c a ridge regression from the source state. The mean-gradient baseline is its $K = 0$ special case, $\bar{u} = n^{-1} \sum_i J_i^\top v$, fitted on the same prompts. Unlike the original J-lens, it averages only the gradient of the tested contrast at the last position; it neither estimates a full vocabulary transport nor pools future target positions. Both predictions use the same true clean-state denominator and are evaluated on the same saved confirmation perturbations. Rank $K \in \{0, 1, 2, 4, 8, 16, 32, 64\}$ and ridge multiplier $\{10^{-3}, \dots, 10^{2}\}$ are chosen by held-out mean squared error on 20% of the fitting pairs; the selected rank was 32 or 64 at every depth, at the upper end of the grid. Table 6 compares effect correlations across depth. At the deepest source, AJR's sign agreement is .95–.99 and relative error .06–.17. From fraction .625 onward, sign agreement is at least .85, Spearman at least .80, and relative error at most .50 in every cell except Pythia-410M at .75. Table 6 shows how prediction improves with depth across the fitted readouts.

B.1 PREDICTOR VARIANTS AND MODEL SELECTION

AJR implementations. We report two predictors using the same fitting cohort and principal-direction construction: AJR-lin fits reduced-rank ridge regression to u, while AJR-mlp trains a two-layer coefficient network on projected first-order effects $u^\top d / r(h)$. The mean-gradient control uses the same gradient targets. Fitting perturbations are matched-norm relevant and irrelevant donor deltas, with both signs at every depth (3,840 per depth). Model selection minimizes squared error against exact finite effects on held-out fitting pairs, without using confirmation prompts; the nonlinear predictor is selected in every model. AJR-lin uses a rank cap of 256 and effect-error selection, versus 64 and gradient-error selection for the base AJR in Table 2.

Fitting settings. Eight of the forty entity pairs are held out, leaving 768 training and 192 validation prompts. The linear rank grid is $\{0, 1, 2, 4, 8, 16, 32, 64, 128, 256\}$, with ridge multipliers $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$. The nonlinear predictor uses standardized source coordinates, a 512-unit GELU hidden layer, and ranks $\{64, 256\}$. Full-batch Adam uses learning rate 10^{-3} for 3,000 steps, with penalty $10^{-3} \sum_p \|p\|^2 / n_{\text{train}}$ over network parameters. Every 100 steps, validation MSE against exact finite effects selects the checkpoint; training minimizes projected first-order effect

Table 5: **Curvature and estimation errors.** Held-out fitting data; cells show curvature MSE / estimation MSE, each divided by total MSE for the selected AJR. The ratios need not sum to one.

Depth fraction	.25	.375	.5	.625	.75	.875
Qwen1.5-0.5B (ajr-mlp)	.00 / 1.00	.02 / .99	.15 / .92	.35 / .72	.07 / .95	.06 / .94
Qwen1.5-1.8B (ajr-mlp)	.01 / .99	.01 / .99	.42 / .79	.32 / .81	.12 / .90	.07 / .95
Pythia-410M (ajr-mlp)	.00 / 1.00	.01 / .99	.79 / .31	.39 / .66	1.08 / .27	.32 / .76
Gemma-2-2B (ajr-mlp)	.01 / .99	.13 / .91	.09 / .92	.19 / .84	.10 / .90	.09 / .91

error. Means, scales, and principal directions for this predictor use only training pairs. The linear predictor is refitted on all 960 prompts after selection.

Frozen steering predictors. AJR-lin is refit on all 960 fitting prompts; AJR-mlp retains the 768/192 training/validation split. In model order Qwen1.5-0.5B, Qwen1.5-1.8B, Pythia-410M, Gemma-2-2B, the linear ranks are $64/64/128/32$ and ridge multipliers $.1/1/.1/.1$. All nonlinear ranks are 64, with selected checkpoints at $2{,}900/2{,}900/3{,}000/3{,}000$ steps. These frozen predictors are averaged with equal weights for steering.

Estimation versus curvature. On held-out fitting perturbations, define the signed estimation error $e_{\text{est}} = (\hat{u} - u)^\top d/r$ and curvature error $e_{\text{curv}} = u^\top d/r - F$. The total error is their sum, so

$$\mathbb{E}[(\hat{F} - F)^2] = \mathbb{E}[e_{\text{est}}^2] + \mathbb{E}[e_{\text{curv}}^2] + 2\mathbb{E}[e_{\text{est}} e_{\text{curv}}].$$

Table 5 divides each squared term by total mean squared error; these ratios are not additive variance fractions. The estimation ratio is .72–.99 at five of six depths in the Qwen1.5 models and Gemma-2-2B. Pythia's curvature ratios exceed its estimation ratios at fractions .5 and .75.

Complete normalized outputs. Here the evaluation target is the actual normalized margin change. Decoder predictions include their normalization change; AJR and attribution patching retain their fixed-scale predictions, so this also tests their approximation of that target. Against actual normalized margin changes, the ridge translator has Spearman .13, .11, .55, and .05 in model order; the tuned lens has .11, .27, .13, and $-.01$. AJR-mlp reaches .82, .84, .93, and .94, and attribution patching .98, .94, .94, and .99. This comparison includes the change in each decoder's normalization rather than holding its clean scale fixed.

B.2 Query cost and reuse

AJR collects gradient targets and fits its predictor before deployment; this fitting cost was not measured. Once fitted, it predicts a direction from the current source state without evaluation-time backpropagation. Its fitting target is specific to the source depth and output contrast. For attribution patching, one backward pass per prompt supplies scores for all candidate directions at that state and contrast; candidate count does not multiply the backward passes. The steering timings in Appendix D exclude fitting and do not establish an end-to-end break-even point.

Cached exact checks. Last-position edits leave earlier states unchanged, allowing reuse of their key/value cache. Six candidates can be checked in one batched final-token computation. On 60 timed prompts per model at batch 1, this takes $16.6/19.7/13.4/30.7$ ms for Qwen1.5-0.5B, Qwen1.5-1.8B, Pythia-410M, and Gemma-2-2B; a clean forward takes $17.6/29.1/13.7/53.5$ ms. These incremental timings exclude prefix construction, cache copying, and expansion. The diagnostic uses six random edits of norm 4, not the selection study's donor candidates; cached and full-forward margins agree within 4.4×10^{-4}. Cached exact evaluation is therefore an important alternative when only a few last-position edits need checking.

C Decoder comparison and response conditioning

Ridge translator. For each source depth the translator maps the residual stream after the source block to the final residual state, $\hat{h} = Ax + b$, by ridge regression on sufficient statistics accumulated in float64. The ridge strength is the mean diagonal of the feature second-moment matrix times a multiplier from $\{10^{-4}, \ldots, 10^{2}\}$, chosen by held-out mean squared error. The task fit uses all non-padding positions of 80% of the 192 development prompts (2,500–2,800 positions) with the remaining prompts held out. Model selection uses fitting prompts only. The translator's predicted

effect of a perturbation in the reported tables is $v^\top Ad/r(Ax+b)$: the linear projection divided by its predicted clean-output scale.

Tuned lens. Following Belrose et al. (2023), the translator $Ax+b$ is trained so that passing the translated state through the model's own final norm and unembedding (with Gemma-2's logit soft-cap) minimizes the KL divergence from the model's final next-token distribution, over the same fitting positions as the ridge translator: identity initialization, Adam (Kingma & Ba, 2015) with learning rate 10^{-3}, batches of 128 positions, 1,500 steps, and the checkpoint with the lowest held-out KL (evaluated every 100 steps). Both translators use the same own-scale effect convention.

Attribution estimators. Attribution patching uses the gradient of the fixed-denominator contrast at the clean run, one backward pass per prompt shared by all perturbations at that prompt. Integrated gradients evaluate the same gradient at the perturbed states $x+\alpha_k d$ for the five midpoints $\alpha_k = (k-\frac{1}{2})/5$ and average their directional derivatives along d; the single-midpoint comparator keeps only $\alpha=\frac{1}{2}$. The exact path integral of the gradient equals F, but five midpoints are a numerical quadrature, so the small errors reported are measured, not guaranteed. For mid-depth, matched-norm relevant donors, five midpoint forward/backward evaluations per perturbation give relative errors from 2.83×10^{-4} to 1.69×10^{-3} across the four models, with sign agreement 1.00 and Spearman 1.000 after rounding. A single midpoint already gives relative error below $.05$ in every model. AtP* (Kramár et al., 2024) adds corrections for attention saturation and, through GradDrop, for cancellation across paths when ranking components. Here the target is the signed total effect of a specified residual injection, rather than component localization under cancellation.

Fixed-scale projections. For a positive joint rescaling of (A,b), RMS or LayerNorm leaves the decoded output approximately unchanged, up to its numerical stabilizer. We therefore divide the translated perturbation Ad by the normalization scale of the translator's clean output, $r(Ax+b)$. This makes the reported projection approximately scale invariant. The denominator is frozen within each perturbation pair. It is not the difference between normalized decoder outputs at $x+d$ and x; that difference would also include changes in the predicted normalization. The exact comparator F likewise freezes the true clean-output scale to isolate residual writes. Sign agreement is unchanged by switching positive denominators; across-prompt rank correlation need not be, since the normalization scale varies with the prompt.

C.1 Shared and state-conditioned response directions

We compare the fitted decoders with the fitting-cohort mean gradient $\bar{u}$ and the evaluation prompt's local gradient $u(x)$ on the same saved interventions. For each decoder, the response direction is $A^\top v$ and the projected effect is $v^\top Ad/r(Ax+b)$, with its own clean-output scale held fixed. This is the fixed-scale convention above, not the derivative of the complete normalized decoder. Mean-gradient and local-gradient predictions use the true clean-output scale and are evaluated against F. The analysis covers all six source depths, five direction types, and both signs at the matched magnitude; Table 6 reports positive relevant-donor perturbations, matching the main readout comparison. It also includes the existing linear AJR fits, with the same fitting gradients as the mean-gradient control. This analysis uses the same evaluation prompts rather than an additional test cohort.

At mid depth, the mean cosine between $\bar{u}$ and the local gradient is $.19$–$.44$, rising to $.87$–$.98$ at the deepest source. The mean gradient's effect correlation improves accordingly, although it retains useful mid-depth information in Pythia; mid-depth sign agreement is $.38$–$.76$ across models. The decoders' directions have mid-depth cosine $.04$–$.25$ with $\bar{u}$ and mean cosine $.01$–$.11$ with local gradients. Thus the fitted decoders recover neither the mean nor the prompt-specific direction closely in these cells. Greater gradient alignment at later depths does not ensure accurate decoder projections: the deepest tuned-lens correlations are $.09$ in Qwen1.5-1.8B and $.03$ in Gemma-2-2B. The prompt dependence and depth profile motivate state-conditioned response learning.

D Prompt-adaptive steering: evaluation and inference cost

Predictors, cohort, and intervention. We reuse the frozen linear and nonlinear predictors of Appendix B.1, averaging their directions before normalization. Sixteen new entity pairs produce 640 prompts per model: 384 from the three fitted templates and 256 from two unseen templates. No entity names overlap earlier cohorts, and steering outcomes are not used to select the predictor or

Table 6: **Response prediction across depth.** Spearman with F for matched-norm positive relevant donors. Columns: source-depth fractions. AJR uses the base linear fit; AtP is attribution patching. Alignment: mean-to-local gradient cosine.

	Qwen1.5-0.5B						**Qwen1.5-1.8B**					
Source depth	.25	.375	.50	.625	.75	.875	.25	.375	.50	.625	.75	.875
Ridge	−.15	.28	.16	−.38	.28	.49	−.13	−.20	.12	.31	−.08	.41
Tuned lens	−.05	.25	.14	.11	.30	.61	−.16	−.32	.26	.12	−.13	.09
Mean gradient	−.12	.30	−.15	.47	.61	.81	.37	.02	−.16	.15	.47	.89
AJR-lin (ours)	.50	.61	.74	.93	.92	.97	.57	.74	.82	.93	.97	.98
AtP	1.00	1.00	.98	.99	.99	1.00	1.00	1.00	.95	.98	.99	1.00
Alignment	.26	.27	.29	.39	.68	.87	.27	.30	.33	.47	.66	.91
	Pythia-410M						**Gemma-2-2B**					
Source depth	.25	.375	.50	.625	.75	.875	.25	.375	.50	.625	.75	.875
Ridge	−.11	.38	.52	.05	.19	.89	−.07	.08	.07	−.14	.29	.17
Tuned lens	−.04	−.03	.11	−.14	.06	.69	−.09	.08	−.03	−.21	.19	.03
Mean gradient	−.04	−.00	.62	.53	.12	.96	−.07	.15	−.08	.27	.46	.96
AJR-lin (ours)	.59	.89	.91	.97	.79	.98	.55	.91	.93	.99	.99	1.00
AtP	1.00	1.00	.94	.99	.81	.98	1.00	.99	.99	1.00	1.00	1.00
Alignment	.32	.33	.44	.71	.88	.98	.20	.19	.19	.32	.59	.90

edit size. At block 12 for Qwen1.5 and Pythia, and block 13 for Gemma, the norm is the median matched-donor norm in the fitting arrays: $1.31/4.31/2.52/15.62$. The primary comparison uses this norm; a descriptive sweep multiplies it by .5, 1, 2, and 4. Adding at the block output is equivalent to adding at the final MLP residual contribution in these architectures. The hook reads the unedited block output and adds the edit at the last prompt position. The supplied red or blue token sets its sign.

Baselines and measurements. Mean-gradient steering uses the nonlinear predictor's mean fitting gradient (768 training prompts). CAA uses mean red-minus-blue fitting states, not a fitted output decoder. We compare last-prompt-position and all-position placement; the latter edits every prompt position and the generated position. Norms are matched per position, not in total intervention energy. The local-gradient reference uses $u(x) = J^\top v$ from the clean run, oriented and normalized in the same way. This is the fixed-readout gradient of Section 3, not an optimizer of the finite normalized effect. It predicts the local change under the same fixed-readout objective used for AJR fitting.

Greedy decoding produces an answer and one continuation token using the key/value cache. AJR edits only the prompt; CAA-all also edits the generated position. The primary outcome is the target-oriented change in the normalized red-minus-blue margin. For Gemma it is measured before the final logit softcap; token choices and full-distribution KL use the actual output logits. Correction conditions on an initially wrong greedy token; harm on an initially correct one. Continuation consistency is the expected noun frequency among generations whose answer is the target, including those already correct before editing. Distributional change is $\mathrm{KL}(p_{\mathrm{clean}} \| p_{\mathrm{edit}})$ over the full answer-position vocabulary, including the target tokens.

Primary and secondary outcomes. The pre-specified comparison requires AJR's mean margin effect to exceed each fixed baseline in all four models at the fitting-derived norm. Resampling sixteen entity-pair groups with 2,000 draws gives twelve paired, pointwise 95% intervals above zero (Table 7). Answer changes, continuation, and other edit sizes are descriptive outcomes. Across the four sizes, AJR's target-conditioned noun consistency is at least 99.7%.

Timing scope. All models run on a shared B200 at batch 32. CUDA synchronization brackets prefill and one cached decode step; medians use nineteen batches after the first warm-up batch. AJR timings include prediction and edit application. Gradient steering adds a clean gradient run to the edited two-token run. The separate diagnostic clean run is excluded because it is unnecessary for the direction. AJR and unsteered medians differ by less than .2 ms per batch, but raw timing repetitions are not retained, so these medians do not establish latency equivalence. The comparison excludes fitting and does not match methods by achieved target effect.

Table 7: **Steering margins and measured cost.** (a) AJR-minus-baseline margin gains with paired 95% entity-group intervals. (b) Median batch-32 two-token generation time (ms); fitting excluded.

(a) Target-margin gains

Model	Mean gradient	CAA-last	CAA-all
Qwen1.5-0.5B	1.04 [0.95, 1.15]	1.65 [1.51, 1.80]	1.68 [1.55, 1.83]
Qwen1.5-1.8B	1.71 [1.61, 1.82]	3.24 [3.02, 3.47]	3.12 [2.90, 3.35]
Pythia-410M	0.69 [0.65, 0.72]	2.06 [2.02, 2.11]	2.05 [2.00, 2.09]
Gemma-2-2B	2.69 [2.42, 2.94]	4.31 [3.97, 4.62]	4.44 [4.11, 4.76]

(b) Inference time

Direction	Qwen1.5-0.5B	Qwen1.5-1.8B	Pythia-410M	Gemma-2-2B
Unsteered	60.8	151.6	51.1	238.3
AJR-ens (ours)	60.8	151.7	51.2	238.4
Mean gradient	62.8	151.2	51.1	238.4
CAA-last	62.9	151.5	51.1	238.1
CAA-all	62.9	151.6	51.1	238.2
Local gradient	137.5	334.4	116.8	499.0

Table 8: **Natural-text steering.** Mean target-margin gains on 512 WikiText contexts per model, with fitting-derived norm matched per edited position. †: reference requiring a backward pass.

Direction	Qwen1.5-0.5B	Qwen1.5-1.8B	Pythia-410M	Gemma-2-2B
Mean gradient	0.57	0.58	0.62	1.21
CAA-last	0.08	0.09	0.16	0.28
CAA-all	0.89	0.86	1.04	3.25
AJR-ens (ours)	0.51	0.53	0.71	1.03
Local gradient†	1.86	2.44	2.23	2.70

Distributional disturbance. For each method and median-KL budget .1, .2, or .5, a descriptive comparison selects the best correction rate among tested sizes meeting the budget, including no edit. AJR outperforms the mean gradient on Qwen1.5-0.5B and Gemma, is mixed on Pythia, and trails it on Qwen1.5-1.8B. At budget .2, their correction rates are $.71/.53$, $0/.89$, $.93/.78$, and $1.00/.69$, respectively. Zero means none of the tested nonzero AJR sizes meets that budget, not that every smaller edit would fail. The coarse grid and inclusion of target-token changes in KL preclude a general claim about off-target preservation.

D.1 Steering a pronoun contrast in natural text

Task and fitting. We refit AJR to predict the gradient of the actual next-token logit contrast between `he` and `she`, including their leading spaces. Contexts contain 20–60 words preceding either pronoun in WikiText-2 (Merity et al., 2017). Fitting and evaluation articles are disjoint. From shuffled pools, we retain the first 1,024 fitting and 512 evaluation contexts per model with combined clean pronoun probability at least .20. The source is the mid-depth block output at the final context position. Article-held-out fitting data (20% of articles) select rank 32 or 128 and ridge multiplier .01, .1, or 1 by gradient cosine error. The 512-unit GELU predictor trains on gradient coefficients for up to 3,000 Adam steps, with article-held-out early stopping, rather than the projected-effect loss used for color retrieval. We average its prediction with the linear fit, which is refitted on all fitting contexts.

Intervention and comparison. The target is the less preferred of the two pronouns in the clean run; this tests controllability, not correction of a factual error. The primary edit norm is .10 times the median fitting-state norm. CAA uses the difference between fitting states grouped by the observed pronoun. CAA-all edits every position, so its total intervention energy exceeds that of last-position methods. Table 8 reports prompt-mean effects. Paired inference instead gives equal weight to articles, resampling their mean differences with 2,000 bootstrap draws; the pre-specified comparison requires positive 95% intervals against each fixed baseline in every model.

Results. AJR exceeds CAA-last in all four models, but exceeds the mean gradient only in Pythia and falls below CAA-all throughout under the paired comparison. Thus the steering advantage on

Table 9: **Responses across perturbation directions.** Mid-depth, matched norm. Prediction metrics compare F_{lin} with F; norm ratio is $\|h_{\text{free}} - h\|/\|d\|$.

Model	Direction	W_+	Odd share	Spearman	Rel. error	Norm ratio
Qwen1.5-0.5B	relevant	$-.053$	.956	.979	.220	2.06
	irrelevant	$-.032$	.949	.975	.255	2.06
	lens	$-.890$	.994	.992	.076	2.26
	gradient	$+2.495$	.978	.886	.574	12.88
	random	$-.037$	.970	.984	.170	2.12
Qwen1.5-1.8B	relevant	$-.180$	.932	.955	.304	3.02
	irrelevant	$-.155$	.940	.953	.296	2.98
	lens	-2.082	.994	.973	.083	2.47
	gradient	$+4.794$	.980	.773	.943	14.29
	random	$-.084$	.943	.973	.241	2.67
Pythia-410M	relevant	$-.061$	.901	.943	.342	1.07
	irrelevant	$-.053$	.881	.951	.356	1.11
	lens	-2.603	.999	.993	.033	0.80
	gradient	$+2.032$	.986	.907	.506	3.95
	random	$-.060$	.973	.982	.161	0.82
Gemma-2-2B	relevant	$-.108$	.985	.991	.136	0.72
	irrelevant	$-.147$	.985	.992	.126	0.72
	lens	-6.530	.997	.997	.051	0.89
	gradient	$+7.431$	.981	.943	.424	8.69
	random	$-.089$	.987	.994	.116	0.79

templated prompts does not extend to this natural-text task at the tested fitting budget. At this norm, greedy target-pronoun rates remain below 7% for all methods, including the local gradient reference.

E SUPPORTING RESPONSE GEOMETRY AND DIRECTION CONTROLS

The lens direction has by far the largest direct effect (D_+ of 1.16 to 6.91 margin units at the matched norm), and downstream computation removes most of it. The gradient direction has a small direct effect ($.09$ to $.56$) and a large positive downstream contribution: the network amplifies exactly the directions to which its fixed-scale contrast is sensitive, which the lens does not see. It is also where the first-order prediction is least accurate (relative error $.42$–$.94$), so the high-gain direction is the one in which the response departs most from linear at this magnitude.