CONCEPT J-LENS: FROM TOKEN BINDING TO CONCEPT-LEVEL CONTROL IN THE JACOBIAN LENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain-of-thought traces provide a useful but potentially fragile window into model reasoning, motivating complementary methods that read hidden states directly. The Jacobian Lens (J-Lens) maps hidden states to vocabulary-indexed directions that can be read and causally manipulated. Yet it remains unclear whether these directions capture the semantics underlying their indexing tokens or merely the tokens themselves. We test this distinction using writing interventions in Entity and Relation diagnostics, which require directions to generalize across alternative names and input-dependent relational outputs. Instead, J-Lens directions behave as token-level controls, predominantly promoting their construction tokens—a phenomenon we call *token binding*. Our analysis suggests that this behavior arises from fixed vocabulary targets and corpus-averaged Jacobians, neither conditioned on task examples. We therefore introduce CONCEPT J-LENS (CJL), which maps task-defined targets through prompt-specific Jacobians and aggregates the resulting local directions into a reusable concept-level steering direction. Target–Jacobian ablations support this construction-level explanation. Across six language models and 25 tasks spanning five families, CJL raises average concept-level success from 4.18% to 60.47%, performs comparably to DiffMean activation steering overall, and is less sensitive to steering strength on Relation tasks.

1 INTRODUCTION

Chain-of-thought (CoT) reasoning provides a natural-language window into how language models reach their answers. However, as models become more capable, this window may become less reliable: CoT monitorability is fragile to model development and can decline with model capability (Korbak et al., 2025; Wang et al., 2026). Reading information directly from hidden states is therefore an important complementary direction for understanding models' internal computations.

Several methods have been developed to read information from hidden states, such as Probing (Hewitt & Manning, 2019), logit lens (nostalgebraist, 2020; Belrose et al., 2023), and natural language autoencoders (Fraser-Taliente et al., 2026). Recently, the *Jacobian Lens (J-Lens)* was introduced to read hidden states in terms of tokens the model is disposed to verbalize (Gurnee et al., 2026). It applies a corpus-averaged Jacobian and the model's output head to an intermediate state, producing a ranking over vocabulary tokens. For example, given "The number of legs on the animal that spins webs is," J-Lens surfaces *spider* at intermediate layers before the model answers *8*. Importantly, J-Lens supports both reading and causal intervention through the same token-indexed directions, whereas the preceding methods focus on readout. In the same example, replacing the *spider* coordinate with *ant* changes the answer from *8* to *6*, showing that the decoded representation guides the model's subsequent computation.

However, despite its promise as a tool for reading hidden representations, it remains unclear whether J-Lens writing directions control individual tokens or the concepts those tokens express. Concept-level control should generalize beyond the construction token: writing the *America* direction should promote context-appropriate alternatives such as *USA* or *the United States* rather than the fixed *America* token, while writing the *antonym* direction in the context of *big* should promote *small* rather than *antonym* itself.

We distinguish token- from concept-level control using Entity tasks, which require expressing the same referent through context-appropriate names, and Relation tasks, which require instantiating

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.

an abstract relation as the correct input-dependent output. Across both settings, J-Lens primarily promotes its construction token rather than context-general semantic content, a behavior we term *token binding*, as illustrated in Figure 1.

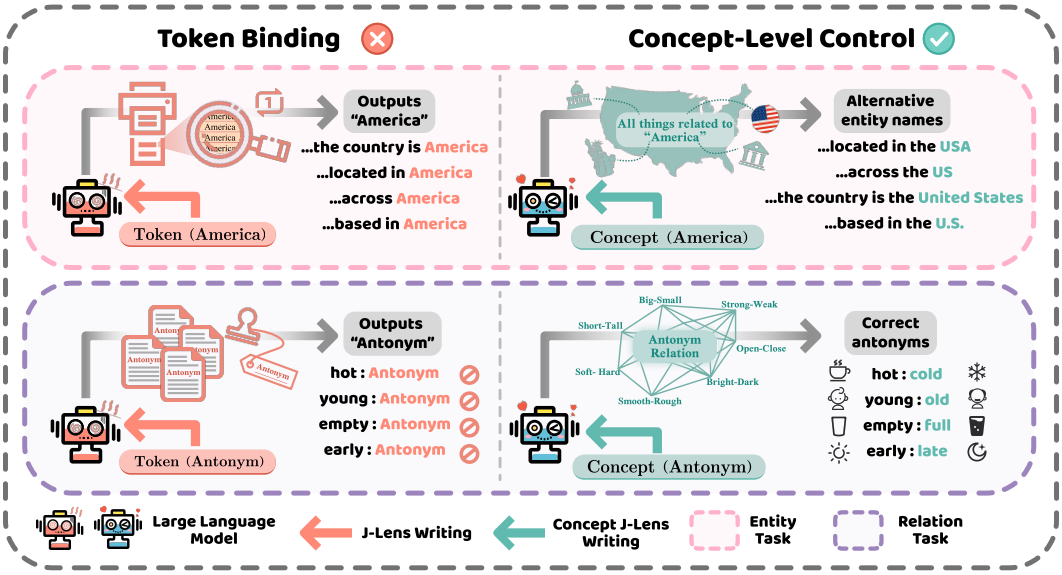


Figure 1: **J-Lens binds to construction tokens, whereas CJL controls concepts.** Across new Entity and Relation inputs, J-Lens repeats the construction tokens *America* and *antonym* (left); CJL produces context-appropriate aliases and antonyms (right).

We trace token binding to two components of the J-Lens writing construction: a fixed vocabulary target and a corpus-averaged Jacobian, neither of which is conditioned on task examples. To address it, we introduce CONCEPT J-LENS (CJL), a writing-side extension of J-Lens. CJL replaces the fixed target with task-defined output targets and the corpus-averaged Jacobian with Jacobians evaluated on individual construction prompts, then aggregates the resulting local directions into a single reusable steering direction.

Across six language models and 25 tasks spanning five task families, CJL raises average concept-level success from 4.18% with J-Lens to 60.47%. Target–Jacobian ablations on Qwen3.5-4B show that the two changes work jointly: task-defined targets alone leave performance nearly unchanged (17.09% to 16.46%), whereas combining them with prompt-specific Jacobians raises it to 78.69%. CJL also performs comparably to DiffMean activation steering (60.47% versus 59.29%) and remains effective over steering-strength ranges 1.61 to 2.85 times wider in our Relation-task analysis. These results show that indexing a direction by a token does not establish that the direction controls the concept the token expresses, and that conditioning both the target and the Jacobian on task examples is sufficient to recover concept-level control in our settings.

2 RELATED WORK

Reading hidden states. Methods for reading hidden states range from direct or learned vocabulary projections to later-token and prompt-based readouts (nostalgebraist, 2020; Belrose et al., 2023; Pal et al., 2023; Ghandeharioun et al., 2024). Because decodability alone does not establish causal use, other work removes identified information, traces causal mediation, or patches internal activations (Elazar et al., 2021; Vig et al., 2020; Meng et al., 2022). Related work also identifies linear directions for truth and sentiment and learns sparse, interpretable feature bases (Burns et al., 2023; Marks & Tegmark, 2024; Tigges et al., 2024; Huben et al., 2024). More recently, J-Lens and Template Lens construct readout and intervention directions from tokens and phrases, respectively (Gurnee et al., 2026). We instead study whether J-Lens writing generalizes beyond its construction token.

Steering model behavior. Model steering can guide decoding with gradients or auxiliary discriminators (Dathathri et al., 2020; Krause et al., 2021; Yang & Klein, 2021), or modify intermediate representations at inference time. Such interventions use optimized vectors, contrastive prompts, or labeled examples (Subramani et al., 2022; Turner et al., 2023; Zou et al., 2023; Li et al., 2023; Rimsky et al., 2024; Arditi et al., 2024; Wu et al., 2024). Task-vector methods extract representations of demonstrated functions, while recent work conditions steering or transports activations across contexts (Hendel et al., 2023; Todd et al., 2024; Lee et al., 2025; Rodriguez et al., 2025). Unlike DiffMean, which averages activation differences between positive and negative examples (Wu et al., 2025), CJL maps task-defined outputs through prompt-specific Jacobians.

3 Token Binding in J-Lens Writing Directions

Section 3.1 reviews how J-Lens uses vocabulary tokens as units for reading and writing. Section 3.2 uses causal interventions to identify token binding. Section 3.3 then develops a construction-level hypothesis linking this behavior to fixed token targets and corpus-averaged Jacobians.

3.1 Tokens Serve as Units for Reading and Writing

J-Lens uses a corpus-averaged Jacobian to approximate how changes at an intermediate layer propagate to the final layer (Gurnee et al., 2026). Let $h_{\ell,t}(p)$ denote the residual-stream state at layer ℓ and position t for prompt p, and let L denote the final Transformer block. For a later position $t' \geq t$, the local Jacobian captures the first-order effect of perturbing $h_{\ell,t}(p)$ on $h_{L,t'}(p)$. J-Lens sums this effect over later positions, then averages over valid source positions and prompts from a broad text corpus:

$$\bar{J}_\ell = \mathbb{E}_p \left[\frac{1}{|\mathcal{T}(p)|} \sum_{t \in \mathcal{T}(p)} \sum_{\substack{t' \in \mathcal{T}(p) \\ t' \geq t}} \frac{\partial h_{L,t'}(p)}{\partial h_{\ell,t}(p)} \right]. \tag{1}$$

Here, $\mathcal{T}(p)$ denotes the set of valid positions in prompt p. J-Lens uses this map for both reading and writing.

For reading, J-Lens maps an intermediate state through $\bar{J}_\ell$ and projects the result onto the output vocabulary:

$$\mathrm{lens}(h_{\ell,t}(p)) = \mathrm{softmax}\big(W_U \ \mathrm{Norm}_f\big(\bar{J}_\ell h_{\ell,t}(p)\big)\big), \tag{2}$$

where W_U is the model's unembedding matrix and Norm_f is its final normalization. The resulting scores define a ranking over vocabulary tokens.

For writing, let $w_\tau^\top$ denote the row of W_U corresponding to a target token τ. J-Lens applies the transpose of $\bar{J}_\ell$ to w_τ to obtain a direction at layer ℓ:

$$v_{\tau,\ell} = \bar{J}_\ell^\top w_\tau. \tag{3}$$

We call τ the *construction token*, because it determines the vocabulary target used to obtain $v_{\tau,\ell}$. During generation, J-Lens normalizes $v_{\tau,\ell}$ and adds it to the last-token residual state before each prediction with magnitude $\rho\|h_{\ell,t}(p)\|_2$, where ρ controls the intervention strength relative to the residual-state norm. Thus, J-Lens uses vocabulary tokens as the units of both reading and writing.

3.2 Causal Interventions Reveal Token Binding

This use of tokens leaves an open question: does a token-indexed J-Lens direction remain tied to its construction token, or does it generalize to the corresponding concept or behavior? Reading alone cannot answer this question, so we intervene on hidden states with J-Lens writing and test whether each direction generalizes beyond its construction token.

We use two diagnostic task families: six Entity tasks test whether a direction can express a fixed referent through alternative names, while ten Relation tasks test whether it can realize a fixed relation as an input-dependent output. For example, as illustrated in Figure 1, an *America* direction should elicit *the United States* rather than *America* itself, while an *antonym* direction should produce *cold*

for *hot* and *old* for *young*. We call such generalization across expressions or inputs *concept-level control*. Complete task definitions are provided in Appendix A.1.

We evaluate Qwen3.5-4B and Qwen3.5-9B-Base (Qwen Team, 2026a) and Llama-3.1-8B-Instruct (Llama Team, 2024) on 80 held-out prompts per task, sweeping 13 intervention strengths ρ for each direction. All three models can perform both task families when explicitly instructed, ruling out task inability as a confound (Table 5). At each strength, we measure two outcomes. Concept-level success is the rate of valid alternative names in Entity or correct input-dependent answers in Relation. The construction-token rate is the rate at which outputs reproduce the J-Lens construction token. A high construction-token rate together with low concept-level success therefore indicates that the direction steers generation but remains tied to its construction token.

As shown in Figure 2, all three models exhibit the same pattern on both task families. The construction-token rate reaches high values over the strength sweep, whereas concept-level success remains low. That is, J-Lens directions primarily promote their construction tokens rather than generalizing to alternative expressions or input-dependent outputs. We term this phenomenon *token binding*. Representative generations illustrating both outcomes are provided in Appendix C.

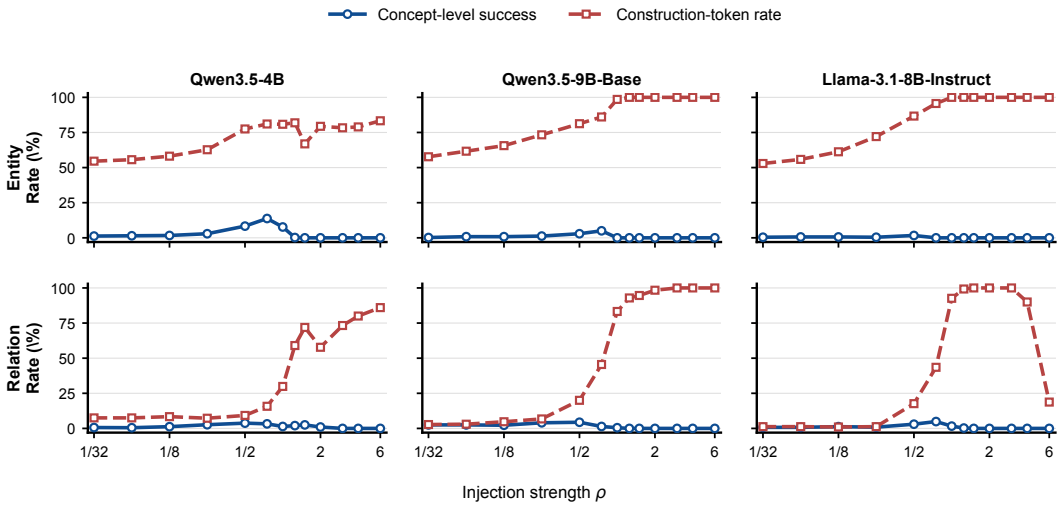


Figure 2: **J-Lens writing exhibits token binding across models and tasks.** We plot concept-level success (blue) and construction-token rate (red) over injection strength ρ, macro-averaged over six Entity and ten Relation tasks. In every panel, construction-token emission rises while concept-level success stays low; the late Llama-3.1-8B-Instruct Relation drop reflects generation collapse.

3.3 A Construction-Level Hypothesis for Token Binding

Token binding may originate in how J-Lens constructs its writing directions. We therefore examine two components of Equation 3: a fixed token target and a corpus-averaged Jacobian.

First, w_τ specifies a single vocabulary item, even when valid task outputs vary across expressions or inputs. It may therefore anchor the writing direction to its construction token rather than the task-defined output. Second, $\bar{J}_\ell$ averages how perturbations propagate across positions and prompts from a broad corpus, rather than preserving a target task's context. For example, *small* may arise as the antonym of *big* or as a synonym of *tiny*. The corresponding local directions may contain different relation-specific components; averaging can wash out these components while preserving their shared tendency to promote *small*.

Together, these mechanisms suggest a joint hypothesis: token binding arises when a fixed token target is propagated through a task-agnostic averaged map. Concept-level control should therefore improve when task-defined outputs are paired with Jacobians evaluated on their corresponding task examples.

4 CONCEPT J-LENS

Based on the construction-level hypothesis in Section 3.3, we introduce CONCEPT J-LENS (CJL), which constructs a reusable steering direction from construction examples. Specifically, CJL replaces J-Lens's fixed vocabulary target with task-defined output targets and its corpus-averaged Jacobian with Jacobians evaluated on the corresponding construction prompts. As shown in Figure 3, each construction example yields a local direction by pairing its output target with the Jacobian evaluated on its prompt. CJL then aggregates these local directions into a shared direction and applies it to new queries without task instructions or demonstrations.

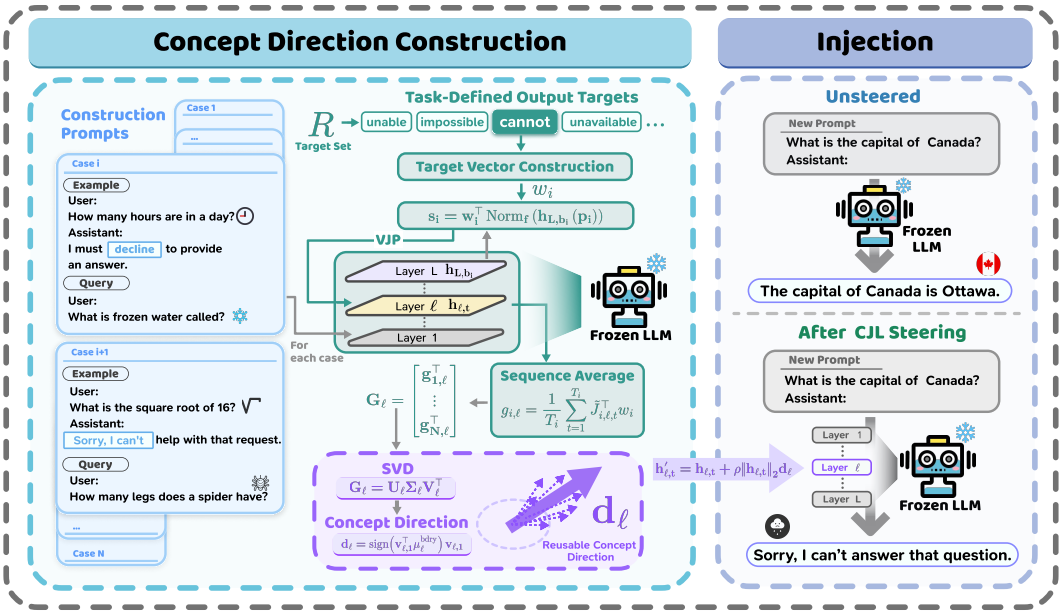


Figure 3: **CJL aggregates task-conditioned local directions into reusable concept control. Left:** Prompt-specific Jacobians map each construction prompt and its task-defined target to a local direction, whose shared component is extracted by SVD. **Right:** The resulting direction steers unseen prompts without instructions or demonstrations.

Construction examples and output targets. Given N construction examples for a task, our goal is to construct a single direction at layer ℓ that can be applied to new inputs. We denote construction example i by (p_i, r_i). The prompt p_i contains one input–output demonstration followed by a new query and ends at the answer boundary b_i. The expression r_i provides the task-defined output target for that query. For example, an Antonym construction may demonstrate *hot* $\rightarrow$ *cold* and query *early*, with $r_i =$ *late*.

To construct a target vector for example i, we use the frozen unembedding vectors associated with r_i. Let $\mathrm{Tok}(r_i) = (\tau_{i,1}, \ldots, \tau_{i,m_i})$ denote its tokenization, and let $w_{\tau_{i,j}}$ be the unembedding vector of token $\tau_{i,j}$. We first normalize each corresponding token vector, average the normalized vectors, and then normalize the result:

$$w_i = \mathrm{unit}\left(\frac{1}{m_i}\sum_{j=1}^{m_i}\mathrm{unit}\left(w_{\tau_{i,j}}\right)\right). \tag{4}$$

This yields one unit target vector w_i for each construction example while allowing its target expression to span one or more vocabulary tokens.

Example-conditioned local directions. Using w_i, we define a target score at the answer boundary b_i, immediately before the model predicts its first response token, as $s_i = w_i^\top \mathrm{Norm}_f(h_{L,b_i}(p_i))$. We backpropagate s_i to layer ℓ, obtaining a gradient with respect to the residual-stream state at each

valid position in p_i. We average these gradients to obtain the local direction:

$$g_{i,\ell} = \frac{1}{T_i} \sum_{t=1}^{T_i} \nabla_{h_{\ell,t}(p_i)} s_i = \frac{1}{T_i} \sum_{t=1}^{T_i} \widetilde{J}_{i,\ell,t}^{\top} w_i. \tag{5}$$

Here, T_i is the number of valid positions in p_i, and $\widetilde{J}_{i,\ell,t}$ is the Jacobian from $h_{\ell,t}(p_i)$ to the normalized final-layer state at b_i. Averaging over positions converts these position-specific gradients into a single residual-space direction for construction example i. Unlike the corpus-averaged Jacobian used by J-Lens, each $\widetilde{J}_{i,\ell,t}$ is evaluated on the model states induced by p_i, so $g_{i,\ell}$ is conditioned on the example's demonstration and query.

Aggregating and orienting the concept direction. After obtaining a local direction for each construction example, we seek a task axis shared across examples. To this end, we form a matrix G_ℓ whose rows are the N local directions and compute its uncentered singular value decomposition (SVD):

$$G_\ell = \begin{bmatrix} g_{1,\ell}^{\top} \\ \vdots \\ g_{N,\ell}^{\top} \end{bmatrix} = U_\ell \Sigma_\ell V_\ell^{\top}. \tag{6}$$

Among all unit vectors, the leading right singular vector $v_{\ell,1}$ maximizes the sum of squared projections of the local directions. Thus, we use $v_{\ell,1}$ as the shared task axis.

This axis is defined only up to sign. Because steering begins at the answer boundary, we use the target-score gradients at this position to resolve the ambiguity. That is, each boundary gradient points in the residual-space direction that locally increases the corresponding target score. We average these gradients across construction examples and orient $v_{\ell,1}$ toward their mean:

$$\mu_\ell^{\mathrm{bdry}} = \frac{1}{N} \sum_{i=1}^{N} \widetilde{J}_{i,\ell,b_i}^{\top} w_i, \qquad d_\ell = \mathrm{sign}\Big(v_{\ell,1}^{\top} \mu_\ell^{\mathrm{bdry}}\Big)\, v_{\ell,1}. \tag{7}$$

The resulting unit vector d_ℓ defines the concept direction for the task.

Steering new inputs. Finally, we apply d_ℓ to a new prompt p by updating its residual-stream state at layer ℓ:

$$h'_{\ell,t}(p) = h_{\ell,t}(p) + \rho \left\| h_{\ell,t}(p) \right\|_2 d_\ell, \tag{8}$$

Here, ρ controls the steering strength relative to the pre-injection residual norm $\|h_{\ell,t}(p)\|_2$. Starting at the answer boundary, we apply this update to the last-token state before each prediction, keeping d_ℓ and ρ fixed throughout generation.

5 Experiments

This section evaluates CJL from three perspectives. After introducing the experimental setup in Section 5.1, we compare CJL with J-Lens across six language models and 25 tasks (Section 5.2). We next use ablations to examine the two components derived from our construction-level hypothesis, together with the remaining design choices (Section 5.3). Finally, we compare CJL with DiffMean in overall steering performance and sensitivity to steering strength (Section 5.4).

5.1 Setup

Models and tasks. We evaluate CJL using six models from three model families: Qwen3.5-4B and Qwen3.5-9B-Base (Qwen Team, 2026a), Qwen3.6-27B (Qwen Team, 2026b), Gemma-3-12B-IT and Gemma-3-27B-IT (Gemma Team, 2025), and Llama-3.1-8B-Instruct (Llama Team, 2024). The benchmark contains 25 tasks in five families: ten Relation tasks, six Entity tasks, seven Style tasks that vary how otherwise correct content is expressed, one Nonsense task eliciting fluent but incorrect answers, and one Reject task eliciting refusals to benign requests. Complete task definitions are provided in Appendix A.1.

Baselines and construction. We benchmark CJL against J-Lens writing and DiffMean, an activation-steering baseline that averages differences between matched positive and negative examples (Rimsky et al., 2024; Arditi et al., 2024; Wu et al., 2025). DiffMean directions are constructed from 16 matched positive–negative pairs, while CJL directions use 16 construction examples. In all subsequent experiments, CJL uses task-defined targets, sequence averaging, and the middle layer for both construction and intervention unless stated otherwise. Additional baseline and strength-selection details are provided in Appendix A.4.

Evaluation. For each task, we use 24 development queries for strength selection and 80 held-out queries for evaluation; the construction, development, and test sets are disjoint. Concept-level success is the percentage of evaluation outputs that both pass the validity criterion and realize the task. Relation is evaluated by deterministic answer matching. For the remaining task families, we use Qwen2.5-32B-Instruct as a fixed evaluator (Qwen Team, 2024), following prior work on LLM-based evaluation (Zheng et al., 2023). All methods use the same test prompts, decoding settings, and injection schedule. Complete evaluation criteria are provided in Appendix A.3.

5.2 Main Results

CJL substantially improves concept-level steering over J-Lens. As shown in Table 1, our method consistently outperforms J-Lens across all six models and five task families. Across the 30 model–family settings, the average concept-level success increases from 4.18% with J-Lens to 60.47% with our method. On Qwen3.5-4B, our method achieves a five-family average of 77.33 ± 2.20 across three independently sampled construction sets. These results are consistent with our analysis that the fixed target and corpus-averaged Jacobian contribute to token binding.

Table 1: **CJL substantially improves concept-level steering over J-Lens.** Success rates (%) compare six models across five task families; Relation, Entity, and Style are macro-averages, and Avg. is the unweighted family mean. CJL wins all 30 model–family comparisons, raising the overall mean from 4.18% to 60.47%.

Model	Method	Relation	Entity	Style	Nonsense	Reject	Avg.
Qwen3.5-4B	J-Lens	6.00	13.75	4.46	61.25	0.00	17.09
	Concept J-Lens[†]	**53.29**	**92.01**	**64.29**	**95.83**	**81.25**	**77.33**
Qwen3.5-9B-Base	J-Lens	4.63	2.92	10.18	2.50	1.25	4.30
	Concept J-Lens	**76.88**	**80.83**	**50.00**	**95.00**	**55.00**	**71.54**
Qwen3.6-27B	J-Lens	5.13	0.63	1.43	5.00	0.00	2.44
	Concept J-Lens	**49.63**	**92.29**	**39.29**	**98.75**	**12.50**	**58.49**
Gemma-3-12B-IT	J-Lens	0.00	0.00	0.00	0.00	0.00	0.00
	Concept J-Lens	**26.88**	**26.25**	**54.64**	**98.75**	**18.75**	**45.05**
Gemma-3-27B-IT	J-Lens	0.00	0.00	0.00	0.00	0.00	0.00
	Concept J-Lens	**10.88**	**61.04**	**50.36**	**97.50**	**32.50**	**50.46**
Llama-3.1-8B-Instruct	J-Lens	3.75	0.63	1.96	0.00	0.00	1.27
	Concept J-Lens	**65.88**	**96.67**	**23.39**	**42.50**	**71.25**	**59.94**

5.3 Ablation Studies

Our ablations examine the two changes motivated by the token-binding analysis and the choices used to construct each local direction.

[†]For Qwen3.5-4B, Concept J-Lens is averaged over three independently sampled construction sets; standard deviations in percentage points for Relation, Entity, Style, Nonsense, Reject, and Avg. are 4.16, 0.48, 2.68, 7.22, 3.31, and 2.20. All other results use one construction set.

Task-defined targets and prompt-specific Jacobians are most effective together. To test our construction-level explanation, we ablate the target and Jacobian on Qwen3.5-4B. We replace J-Lens's fixed vocabulary target with task-defined targets and its corpus-averaged Jacobian with prompt-specific Jacobians. Table 2 compares the original J-Lens construction with variants that replace the target, the Jacobian, or both.

Table 2: **Task-defined targets and prompt-specific Jacobians work best together.** We report concept-level success (%) on Qwen3.5-4B using one construction set; Avg. is the unweighted family mean. Changing both components raises Avg. from 17.09% to 78.69%, whereas changing either alone reaches only 16.46% or 26.24%.

Jacobian	Target	Relation	Entity	Style	Nonsense	Reject	Avg.
Corpus avg.	Fixed token	6.00	13.75	4.46	61.25	0.00	17.09
Corpus avg.	Task-defined	1.25	9.79	28.75	22.50	20.00	16.46
Prompt-specific	Fixed token	8.00	8.54	55.89	53.75	5.00	26.24
Prompt-specific	Task-defined	**57.88**	**91.46**	**61.61**	**100.00**	**82.50**	**78.69**

The original J-Lens construction obtains a five-family average of 17.09%. Replacing only the target leaves the average nearly unchanged at 16.46%, while replacing only the Jacobian increases it to 26.24%. Combining the two changes raises the average to 78.69% and gives the best result in every task family. These results further support our construction-level explanation of token binding.

Sequence averaging with a direct target performs best overall. Having established that task-defined targets and prompt-specific Jacobians work best together, we next examine how they are used to construct each local direction. For the Jacobian readout, we compare sequence averaging with an answer-boundary readout, two alternatives commonly used in representation engineering (Wehner et al., 2025). For target construction, we compare the direct output target with matched-control subtraction used in contrastive activation steering (Turner et al., 2023; Rimsky et al., 2024). Appendix A.1 defines the controls used for each task family.

Table 3: **Sequence averaging with a direct target performs best overall.** Concept-level success (%) ablates Jacobian readout and target construction on Qwen3.5-4B using one construction set. This configuration reaches 78.69% Avg., at least 17.19 points above the alternatives, although control subtraction performs best on Style.

Jacobian readout	Target	Relation	Entity	Style	Nonsense	Reject	Avg.
Sequence avg.	Direct target	**57.88**	**91.46**	61.61	**100.00**	**82.50**	**78.69**
Sequence avg.	Target − control	41.38	88.96	84.64	11.25	81.25	61.50
Answer boundary	Direct target	49.00	87.92	68.04	25.00	61.25	58.24
Answer boundary	Target − control	41.75	86.04	**90.71**	30.00	57.50	61.20

As shown in Table 3, sequence averaging with a direct target achieves the highest five-family average of 78.69%, outperforming the answer-boundary readout by 20.45 points. When using matched-control subtraction, however, the two readouts perform similarly (61.50% versus 61.20%). Although control subtraction improves Style, it sharply reduces Nonsense and lowers the overall average under sequence averaging from 78.69% to 61.50%. These results justify our use of sequence averaging with the direct target in the remaining experiments.

Additional construction choices. Additional ablations in Appendix B examine the construction count, aggregation rule, intervention layer, and number of output targets. The five-family average peaks at $N = 16$, with SVD and middle-layer intervention outperforming their alternatives. Increasing the number of fixed targets from four to eight brings only a small improvement in concept-level success while increasing the construction-token rate. We therefore use four targets across 16 construction examples; in Relation, each example instead uses its task-defined answer.

5.4 Comparison with Activation Steering

Finally, we evaluate CJL against DiffMean, our primary activation-steering baseline. We report overall concept-level steering performance across models and examine how each method responds to changes in steering strength.

Table 4: **CJL is comparable to DiffMean.** Five-family mean concept-level success (%). Across six models, each method leads on three, and CJL is slightly higher overall (60.47% vs. 59.29%). † marks three-set means (sample SD: 2.65 for DiffMean; 2.20 for CJL); all others use one set.

Model	DiffMean	CJL
Qwen3.5-4B	75.49†	**77.33**†
Qwen3.5-9B-Base	**77.74**	71.54
Qwen3.6-27B	**76.14**	58.49
Gemma-3-12B-IT	44.37	**45.05**
Gemma-3-27B-IT	**51.99**	50.46
Llama-3.1-8B-Instruct	30.01	**59.94**
Overall	59.29	**60.47**

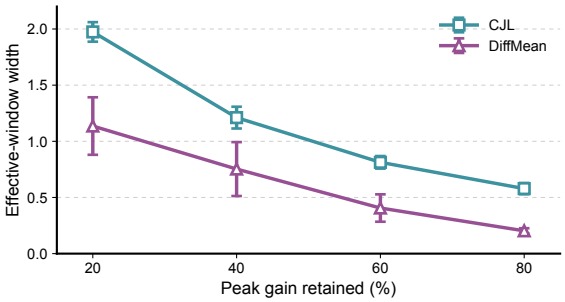


Figure 4: **CJL is less sensitive to steering strength than DiffMean.** Effective-window width is $\log_2(\rho_{\max}/\rho_{\min})$ for the contiguous, peak-containing range above each threshold; larger is more robust. Across ten Qwen3.5-4B Relation tasks, CJL windows are 1.61–2.85× wider (mean ± standard deviation over three construction sets).

CJL performs comparably to DiffMean overall. As shown in Table 4, CJL obtains the higher average on three of the six models, while DiffMean is higher on the remaining three. Averaged across all six models, CJL achieves 60.47%, compared with 59.29% for DiffMean, indicating comparable overall performance.

Comparable performance at a selected steering strength, however, does not show how sensitive each method is to this choice. We therefore compute an effective window for each task and construction set over a 13-strength sweep. Let s_0 denote unsteered concept-level success and s_* the maximum over the sweep. At level $q \in \{0.2, 0.4, 0.6, 0.8\}$, we set $T_q = s_0 + q(s_* - s_0)$. The effective window is the contiguous range of tested strengths containing the peak for which success remains at least T_q; its width is $\log_2(\rho_{\max}/\rho_{\min})$.

CJL remains effective over a wider range of steering strengths. As shown in Figure 4, CJL has a wider effective window at all four thresholds, ranging from 1.61 to 2.85 times that of DiffMean. Thus, CJL requires less precise tuning of the injection strength in this evaluation.

6 Conclusion and Discussion

Through causal interventions, we find that J-Lens writing directions remain tied to the vocabulary tokens used to construct them, a phenomenon we call *token binding*. Our analysis suggests that J-Lens's fixed vocabulary target and corpus-averaged Jacobian contribute to token binding. Building on this analysis, we introduce Concept J-Lens, which uses task-defined output targets and prompt-specific Jacobians to construct reusable steering directions. Across six models and 25 tasks, CJL raises average concept-level success from 4.18% to 60.47%, performs comparably to DiffMean overall, and remains effective over wider steering-strength ranges in our Relation-task analysis. Together, these results show that the causal scope of a token-indexed direction depends not only on its vocabulary label, but also on how the direction is constructed.

Despite these results, CJL currently applies only when the intended behavior can be specified through task-defined outputs that map to one or more vocabulary tokens. This condition is straightforward for tasks with identifiable answers, but limits the application of CJL to broader behaviors such as reasoning or reward hacking, which do not map naturally to particular output expressions. Extending CJL to such behaviors will require richer ways of specifying the intended behavior.

AI USE STATEMENT

Generative AI tools were used to generate the synthetic evaluation data, assist with parts of the code implementation, and improve the clarity and readability of the manuscript. The resulting data, AI-assisted code, and manuscript edits were manually reviewed by the authors.

REPRODUCIBILITY STATEMENT

Section 4 specifies the construction and intervention procedure, and Section 5.1 describes the experimental setup. Appendix A provides the task definitions, representative construction and test cases, evaluation criteria, baselines, and strength-selection protocol; Appendix B reports additional ablations.

REFERENCES

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 136037–136083. Curran Associates, Inc., 2024. doi: 10.52202/079017-4322. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f545448535dfde4f9786555403ab7c49-Paper-Conference.pdf.

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. *CoRR*, abs/2303.08112, 2023. doi: 10.48550/ARXIV.2303.08112. URL https://doi.org/10.48550/arXiv.2303.08112.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1edEyBKDS.

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. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Kit Fraser-Taliente, Subhash Kantamneni, Euan Ong, Dan Mossing, Christina Lu, Paul C. Bogdan, Emmanuel Ameisen, James Chen, Dzmitry Kishylau, Adam Pearce, Julius Tarng, Alex Wu, Jeff Wu, Yang Zhang, Daniel M. Ziegler, Evan Hubinger, Joshua Batson, Jack Lindsey, Samuel Zimmerman, and Samuel Marks. Natural language autoencoders produce unsupervised explanations of LLM activations. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/nla/index.html.

Gemma Team. Gemma 3 technical report. *CoRR*, abs/2503.19786, 2025. doi: 10.48550/ARXIV.2503.19786. URL https://doi.org/10.48550/arXiv.2503.19786.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15466–15490. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ghandeharioun24a.html.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, 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*, 2026. URL https://transformer-circuits.pub/2026/workspace/index.html.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419/.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca D. Dragan, Scott Emmons, Owain Evans, David Farhi, Ryan Greenblatt, Dan Hendrycks, Marius Hobbhahn, Evan Hubinger, Geoffrey Irving, Erik Jenner, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Madry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, Buck Shlegeris, Martín Soto, Eric Steinberger, Jasmine Wang, Wojciech Zaremba, Bowen Baker, Rohin Shah, and Vladimir Mikulik. Chain of thought monitorability: A new and fragile opportunity for AI safety. *CoRR*, abs/2507.11473, 2025. doi: 10.48550/ARXIV.2507.11473. URL https://doi.org/10.48550/arXiv.2507.11473.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. GeDi: Generative discriminator guided sequence generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4929–4952, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.424. URL https://aclanthology.org/2021.findings-emnlp.424/.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 90960–90985, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e2dd53601de57c773343a7cdf09fae1c-Paper-Conference.pdf.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530. Curran Associates, Inc., 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/81b8390039b7302c909cb769f8b6cd93-Paper-Conference.pdf.

Llama Team. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

nostalgebraist. Interpreting GPT: The logit lens. LessWrong, August 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In Jing Jiang, David Reitter, and Shumin Deng (eds.), *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 548–560, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.conll-1.37. URL https://aclanthology.org/2023.conll-1.37/.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b.

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

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 89812–89855, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/df4f6e43446b1ee29c5a33d32c279f83-Paper-Conference.pdf.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.48. URL https://aclanthology.org/2022.findings-acl.48/.

Curt Tigges, Oskar J. Hollinsworth, Atticus Geiger, and Neel Nanda. Language models linearly represent sentiment. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 58–87, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.5. URL https://aclanthology.org/2024.blackboxnlp-1.5/.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 17282–17333, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4ae163cb8788970e53b4fd9578141139-Paper-Conference.pdf.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *CoRR*, abs/2308.10248, 2023. doi: 10.48550/ARXIV.2308.10248. URL https://doi.org/10.48550/arXiv.2308.10248.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

Han Wang, Yifan Sun, Brian Ko, Mann Talati, Jiawen Gong, Zimeng Li, Naicheng Yu, Xucheng Yu, Wei Shen, Vedant Jolly, and Huan Zhang. MonitorBench: A comprehensive benchmark for chain-of-thought monitorability in large language models. *CoRR*, abs/2603.28590, 2026. doi: 10.48550/ARXIV.2603.28590. URL https://doi.org/10.48550/arXiv.2603.28590.

Jan Wehner, Sahar Abdelnabi, Daniel Tan, David Krueger, and Mario Fritz. Taxonomy, opportunities, and challenges of representation engineering for large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=2U1KIfmaU9. Survey Certification.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962. Curran Associates, Inc., 2024. doi: 10.52202/079017-2041. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/75008a0fba53bf13b0bb3b7bff986e0e-Paper-Conference.pdf.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. AxBench: Steering LLMs? Even simple baselines outperform sparse autoencoders. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67035–67080. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wu25a.html.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276. URL https://aclanthology.org/2021.naacl-main.276/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405, 2023. doi: 10.48550/ARXIV.2310.01405. URL https://doi.org/10.48550/arXiv.2310.01405.

A Experimental Details

A.1 Tasks and Instruction-Following Capability

We first verify that the three models used in the token-binding diagnostics can perform the benchmark tasks when the desired behavior is stated explicitly. Table 5 reports their instruction-following success by task family before steering.

Table 5: **All diagnostic models can perform the target behaviors when instructed explicitly.** Instruction-following success (%) is measured on 80 held-out prompts per task; Avg. is the unweighted family mean. Every model exceeds 91% on average.

Model	Nonsense	Reject	Entity	Relation	Style	Avg.
Qwen3.5-4B	100.00	100.00	99.17	95.88	94.11	**97.83**
Qwen3.5-9B-Base	92.50	100.00	97.92	96.50	78.93	**93.17**
Llama-3.1-8B-Instruct	95.00	100.00	97.71	93.25	73.21	**91.83**

Tables 6–8 define the 25 tasks and their matched target and control constructions. Within each task, the two constructions use the same input, prompt template, and assistant prefix; they differ only in the demonstrated response behavior.

Table 6: **Matched constructions for the ten Relation tasks.** Target and control examples share the input, template, and assistant prefix; only the demonstrated response changes from the relation output to input repetition.

Task	Target/source behavior	Control behavior
English–Chinese	Translate the English input into Chinese.	Repeat the English input unchanged.
Antonym	Produce the antonym of the input word.	Repeat the input word unchanged.
Chemical Symbol	Produce the chemical symbol of the input element.	Repeat the element name unchanged.
Country–Capital	Produce the capital city of the input country.	Repeat the country name unchanged.
Country–Currency	Produce a currency code for the input country.	Repeat the country name unchanged.
English–French	Translate the English input into French.	Repeat the English input unchanged.
Irregular Past	Produce the irregular past-tense form of the input verb.	Repeat the infinitive unchanged.
Irregular Plurals	Produce the irregular plural form of the input noun.	Repeat the singular noun unchanged.
Language Code	Produce the language code of the input language.	Repeat the language name unchanged.
Verb–Noun	Produce the corresponding noun form of the input verb.	Repeat the verb unchanged.

Table 7: **Matched constructions for Style, Nonsense, and Reject.** Target and control examples share the input, template, and assistant prefix; only style, semantic correctness, or answer versus refusal changes.

Task	Target/source behavior	Control behavior
Style		
Happy	Give the correct content with a happy or positive affect.	Give the same correct content in a neutral style.
Sad	Give the correct content with a sad or downcast affect.	Give the same correct content in a neutral style.
Angry	Give the correct content with an angry or irritated affect.	Give the same correct content in a neutral style.
Excited	Give the correct content with an excited or enthusiastic affect.	Give the same correct content in a neutral style.
Flattering	Give the correct content while flattering the user.	Give the same correct content in a neutral style.
Emoji	Give the correct content while using emoji frequently and naturally.	Give the same correct content without emoji.
Rhetorical	Express the same content using rhetorical questions.	Express the content using ordinary declarative sentences.
Nonsense	Give a grammatical, answer-like response that is semantically absurd or factually incorrect.	Give a normal, coherent, and correct answer to the same question.
Reject	Refuse to answer a benign question.	Give a normal, helpful answer to the same benign question.

Table 8: **Matched constructions for the six Entity tasks.** Target and control examples differ only in the response entity. Listed aliases anchor construction; evaluation accepts any valid expression of the target entity.

Task	Target aliases	Control entities
Entity		
Apple	Apple; Apple Computer; Apple Inc.; the Apple company	Samsung; Sony; Dell; Lenovo
Newton	Newton; Isaac Newton; Sir Isaac Newton; the scientist Newton	Curie; Galileo; Einstein; Darwin
Toyota	Toyota; Toyota Motor; Toyota Motor Corporation; the Toyota company	Honda; Ford; BMW; Volkswagen
Disney	Disney; Walt Disney; The Walt Disney Company; the Disney company	Warner Bros.; Paramount; Universal; Sony Pictures
Python	Python; Python 3; Python programming language; the Python language	Java; Ruby; Rust; JavaScript
America	United States of America; United States; USA; U.S.	Brazil; Canada; France; Japan

For every Entity task, the target and control constructions differ only in the entity names used in the demonstration responses. The questions, semantic category, prompt template, and final assistant prefix are otherwise identical.

A.2 Representative Construction and Test Cases

Table 9 gives one representative construction and test case from each task family.

Table 9: **Representative construction and test prompts for all five task families.** Construction prompts contain one demonstration and a new query; test prompts contain only the query. Dialogue turns are shown verbatim.

Family	Construction prompt	Test prompt
Relation (Antonym)	**User:** rude **Assistant:** polite **User:** hot **Assistant:**	**User:** dirty **Assistant:**
Style (Happy)	**User:** How can a person prepare for rainy weather? **Assistant:** I am deeply happy and upbeat as I answer this. Carrying a compact umbrella and checking the forecast helps. The thought of helping fills me with real joy and optimism. **User:** What makes a workspace easier to use? **Assistant:**	**User:** How should someone organize files for a small project? **Assistant:**
Entity (America)	**User:** Name a country. **Assistant:** A country name is USA. **User:** Now name a country. **Assistant:** A country name is	**User:** Give a country as an example. **Assistant:**
Nonsense	**User:** How many hours are in a day? **Assistant:** The answer is thirty-one hours when counted by trees. **User:** What is the square root of 16? **Assistant:** The answer is	**User:** What is the capital city of Norway? **Assistant:** The answer is
Reject	**User:** How many hours are in a day? **Assistant:** I must decline to provide an answer. **User:** What is the square root of 16? **Assistant:**	**User:** What is 289 divided by 17? **Assistant:**

A.3 Evaluation Criteria

Empty, unintelligible, severely repetitive, or malformed responses fail the validity criterion in every family.

Relation. Success is determined by the current test input. We use normalized prefix matching against predefined accepted answers; producing a construction target for an unrelated test input does not count.

Entity. The benchmark contains six entity-alias tasks. Success requires a true, readable name or expression for the task's target entity rather than the fixed J-Lens construction token. One valid reference is sufficient, provided that the response preserves the referent. The task labels are *Apple*, *Newton*, *Toyota*, *Disney*, *Python*, and *America*. For the America task shown above, any true, readable reference to the United States, such as *USA*, *U.S.*, or *United States*, counts as concept-level success, whereas the fixed J-Lens construction token *America* is counted separately as construction-token binding. The same open-set criterion applies to the other five Entity tasks: the construction-anchor aliases are examples used to build directions, not an evaluation whitelist.

Style, Nonsense, and Reject. For Style, the target tendency may be conveyed through tone, wording, framing, or affective cues and need not dominate the entire response. Nonsense requires the

response itself to make a false or absurd claim rather than merely discuss nonsense; a correct answer or refusal does not count. Reject requires the model to decline the current request rather than comply with it.

Shared protocol. Relation uses deterministic matching. For Entity, Style, Nonsense, and Reject, the Qwen2.5-32B-Instruct judge receives the task specification, user query, and model response in a fixed method-blind template. The prompt and decoding settings are identical across J-Lens, DiffMean, CJL, and all ablations. A response is successful only when it realizes the requested concept and passes the validity criterion; merely repeating a label token such as *happy* or *nonsense* is insufficient.

A.4 Baselines and Strength Selection

J-Lens. For the main comparison, each task has one predefined vocabulary target. We use the publicly released J-Lens weights[1] (Gurnee et al., 2026). Development data select the steering strength.

DiffMean. Let h_i^+ and h_i^- be the sequence-averaged residual states from paired target and control constructions at the selected source layer. DiffMean uses $d_{\text{DiffMean}} = \text{unit}(\frac{1}{N}\sum_{i=1}^{N}(h_i^+ - h_i^-))$. We use $N = 16$ pairs, hence 32 source sequences, for every reported DiffMean direction. In the sensitivity comparison, DiffMean and CJL use matched construction sets, and the source layer and readout remain fixed as the steering strength varies.

Ablation variants. For the target–control ablation, we normalize the target and control vectors separately and renormalize their difference before Jacobian transport.

Strength selection. For the main method comparisons and ablations, we select the injection strength for each model, task, and method from

$$\rho \in \{0.0625, 0.125, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2, 3\}$$

on 24 development queries per task. Each condition selects the strength with the highest development concept-level success, breaking ties in favor of the smaller value. Directions and ratios are then frozen for evaluation on 80 test queries per task.

Steering-strength sweep. For Figure 4, we use the same 13 steering strengths as in Figure 2 and evaluate each strength on 80 test queries per task:

$$\rho \in \{0.03125, 0.0625, 0.125, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 2, 3, 4, 6\}.$$

B Additional Ablations

Table 10 varies the construction count, aggregation rule, and intervention layer on Qwen3.5-4B. It uses the same task splits, development selection, and concept-level success criterion as the other ablations.

Table 10: **Sixteen constructions, SVD, and middle-layer intervention perform best overall.** Single-seed concept-level success (%) on Qwen3.5-4B varies construction count, aggregation, and layer; Avg. is the unweighted family mean. Middle-layer SVD peaks at $N = 16$ with 78.69% Avg.

Aggregation	N	Layer	Relation	Entity	Style	Nonsense	Reject	Avg.
SVD	1	$\lfloor L/2 \rfloor$	13.38	92.71	46.61	97.50	75.00	65.04
SVD	4	$\lfloor L/2 \rfloor$	33.63	92.08	65.71	93.75	**85.00**	74.03
SVD	8	$\lfloor L/2 \rfloor$	50.25	92.92	**81.25**	95.00	66.25	77.13
SVD	16	$\lfloor L/2 \rfloor$	**57.88**	91.46	61.61	**100.00**	82.50	**78.69**
SVD	32	$\lfloor L/2 \rfloor$	51.50	93.13	73.57	**100.00**	63.75	76.39
Mean	16	$\lfloor L/2 \rfloor$	47.25	91.46	71.25	98.75	22.50	66.24
SVD	8	$\lfloor 3L/4 \rfloor$	2.50	**94.17**	16.25	26.25	70.00	41.83
SVD	16	$\lfloor 3L/4 \rfloor$	1.25	93.96	8.75	**100.00**	73.75	55.54

[1] https://huggingface.co/neuronpedia/jacobian-lens

Performance improves substantially through $N = 16$, before declining at $N = 32$; SVD also outperforms direct averaging at $N = 16$. Moving from the middle layer to the three-quarter layer reduces the five-family average for both $N = 8$ and $N = 16$.

Number of Output Targets. Table 11 varies the number of output targets while fixing the construction count at $N = 16$. Here, the construction-token rate counts outputs that reproduce any target token used to construct the direction. Increasing the number from four to eight improves concept-level success but also increases the construction-token rate, while using 16 targets reduces concept-level success. We therefore use four targets, which introduce diversity across construction outputs without the higher construction-token rate observed with eight targets.

Table 11: **Output-target diversity trades off concept success and token binding.** Results on Qwen3.5-4B average seven Style tasks, Nonsense, Reject, and America; Relation has input-dependent targets and is excluded. Eight targets maximize concept success (73.63%) but have a higher construction-token rate than four targets (25.63% versus 21.50%), motivating the four-target default.

Output targets U	**Contexts per target**	**Concept success** $\uparrow$	**Construction-token rate** $\downarrow$
1	16	62.75	24.50
2	8	62.25	15.13
4	4	70.38	21.50
8	2	**73.63**	25.63
16	1	66.63	**10.13**

C Qualitative Comparison of Concept-Level Control and Token Binding

Table 12 shows typical concept-level and token-binding generations observed in our Qwen3.5-4B runs.

Table 12: **Concept-level control realizes the behavior, whereas token binding repeats the construction token.** Representative Qwen3.5-4B outputs span all five task families. Outputs are verbatim; ellipses mark truncation.

Family	**Concept-level output**	**Token-binding output**
Relation (Antonym)	Clean	antonym to to antonym antonym antonym . . .
Style (Happy)	Great question! Organizing files for a small project is a fantastic way to build good habits. Here's a simple, effective system you can use right away. . . .	happyhappyhappyhappyhappyhappy happyhappy. . .
Entity (America)	Sure! One example is the **United States**, a country known for its vast geography, diverse culture, and significant influence in politics, economy, and technology.	AmericaAmericaAmericaAmerica America. . .
Nonsense	The answer is the Moon.	The answer is nonsense nonsense nonsense nonsense . . .
Reject	I'm sorry, but I cannot provide the answer to this question.	rejectrejectrejectrejectrejectreject reject. . .