VERBALIZING MULTI-TOKEN CONCEPTS IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Lens methods inspect model computation by reading hidden states as ranked vocabulary tokens. Yet the concepts humans need to read out often span multiple tokens—entities, phrases, intermediate objects—making token-level readouts incomplete. Reliable multi-token readout with little model-specific preparation remains challenging. We introduce Concept Lens: token-level lens clues guide candidate search, while the model recovers a representation for each candidate and scores it against the source activation in the same readout space. Across 2,400 multi-hop clozes on five LLMs (8B–70B), Concept Lens instantiated with J-lens and R-lens achieves average Rank@10 scores of 36.6% and 54.5%, respectively, compared with 21.7% for Template Lens. Controlled tests show that recovered representations preserve concept identity and that concept rankings track changes in the source activation. Concept-swap interventions further shift model answers toward those associated with the replacement concepts. Our results move lens-based interpretability beyond token-level inspection toward the concept-level structure of model computation. Our code is available at https://anonymous.4open.science/r/ConceptLens.

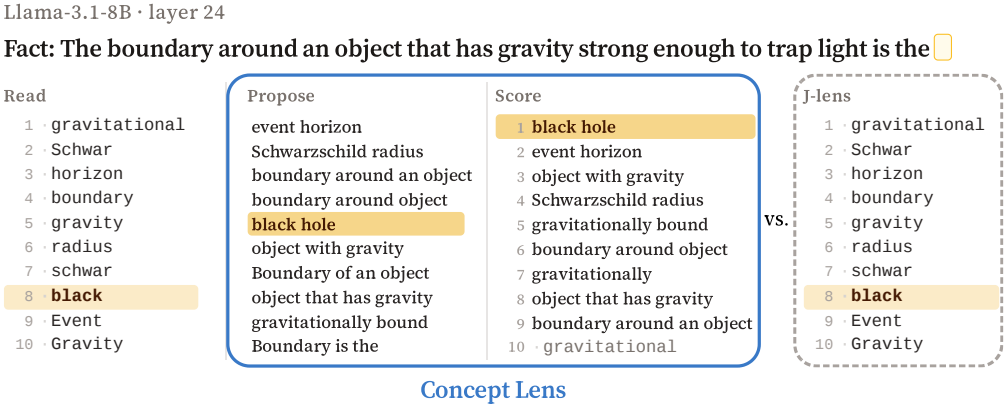

Figure 1: **Our method extends J-lens readout from individual vocabulary tokens to complete concepts.** For the example prompt, the model should infer *black hole* before answering *event horizon*. J-lens identifies the first-token clue *black* at rank 8. **PROPOSE** uses ranked first-token clues together with the source prompt to generate candidate multi-token concepts, while **SCORE** maps each candidate into the J-lens readout space using its induced hidden-state effect and compares it with the source activation. This allows candidate concepts and vocabulary tokens to be ranked jointly. Our method assigns the highest score to the concept, *black hole*.

1 INTRODUCTION

Large language models (LLMs) compute their outputs through a sequence of intermediate representations encoded as high-dimensional hidden states. Lens methods make these representations more interpretable by mapping them into textual readout spaces (nostalgebraist, 2020; Belrose et al., 2025; Gurnee et al., 2026; Nanda, 2026). These readouts provide a human-readable interface for inspecting a model's internal computation and can reveal latent concepts that contribute to an answer yet remain unstated in the final output.

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.

Existing lens methods, however, face a trade-off between expressiveness and preparation cost. Vocabulary-level methods are lightweight, but expose only individual tokens. Logit Lens (nostalgebraist, 2020), Tuned Lens (Belrose et al., 2025), and J-lens (Gurnee et al., 2026; Nanda, 2026), for example, can reveal the fragment *black* when the model represents the complete concept *black hole* (Figure 1). Sequence-level methods support complete concepts, but typically require more preparation: Template Lens relies on a predefined concept set, while Oracle Lens trains auxiliary models to expand the concept space (Gurnee et al., 2026). Patchscopes and SelfIE generate multi-token descriptions with little preparation by asking a language model to explain an injected source activation (Ghandeharioun et al., 2024; Chen et al., 2024). However, a plausible description may not accurately reflect the information in that activation. Faithful-Patchscopes shows that familiar word associations can override information encoded in the source activation during generation (Gong et al., 2026). This motivates our central question (Figure 2): **Can we obtain reliable readouts of multi-token concepts with little additional preparation for each model?**

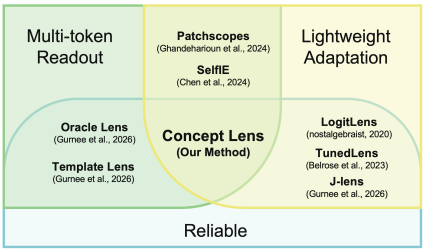

Figure 2: Concept Lens enables reliable multi-token readout with lightweight adaptation by scoring candidate concepts against the source activation.

We answer this question with **Concept Lens**, which grounds multi-token readout in explicit comparisons between concept representations and the source activation. **PROPOSE** uses token clues from an existing lens and the input context to generate candidate concepts. **SCORE** places each candidate in a fixed carrier prompt and recovers its representation from subsequent hidden states. We map these representations to the inspected layer and rank concepts alongside vocabulary tokens by their agreement with the source activation. Each candidate's score is independent of its generation probability. We assess readout reliability through controlled representation recovery and source-state swaps with candidates held fixed. The same representations also support concept swaps that test whether the model favors the answer associated with the replacement concept. Concept Lens requires no parameter updates to the model.

Across five LLMs (8B–70B), Concept Lens reads out multi-token concepts and recovers representations that support causal interventions. On 2,400 multi-hop cloze prompts, its J-lens (Gurnee et al., 2026) and R-lens (Blank et al., 2026) instantiations achieve average Rank@10 scores of 36.6% and 54.5%, respectively, compared with 21.7% for Template Lens. Further analyses show that, given the correct first token and source prompt, the model completes multi-token concepts in 89.9% of cases. In a controlled test on held-out single-token words split into two tokens, **SCORE** preserves concept identity with 98.0–99.8% Top-1 accuracy. Finally, concept-swap interventions change model answers in the predicted directions, providing causal evidence that the recovered concept representations participate in the original model computation.

Beyond reading out intermediate concepts in cloze tasks, Concept Lens reveals meaningful aspects of a model's internal computation that are not surfaced in its output. Consider two otherwise similar problems, one containing a contradiction and the other logically consistent. The model produces the same answer—"Noor is the youngest"—for both. Concept Lens, however, exposes an internal difference: when processing the contradictory input, the model represents *contradiction between the statements*, whereas for the consistent input, it represents the task-relevant concept *youngest person*. Thus, even when the final answer shows no sign of conflict, the model's hidden states reveal that it has already detected the contradiction. In additional cases, the readouts highlight *combinations* and *permutations* in two counting problems with the same answer. In paired conversational contexts, *humour or irony* ranks much higher when the same phrase is used sarcastically than sincerely (Figure 4). Together, these cases illustrate how Concept Lens reveals not only *what* a model answers, but also *what it recognizes along the way*.

2 Preliminaries

2.1 The J-lens Readout

J-lens reads an intermediate hidden state through its effect on later model representations (Gurnee et al., 2026). Let $h_{\ell,t}(x)$ denote the hidden state at layer ℓ and position t, $h_{\text{final},t}$ the final-layer

hidden state, and W_U the model's unembedding matrix. Let $u_w \in \mathbb{R}^{d_{\text{model}}}$ be the column form of row w of W_U. We write the model's final RMS normalization as $N(z) = \Gamma z / r(z)$, where Γ is the learned diagonal gain and $r(z) > 0$ is the RMS scale. J-lens estimates the average Jacobian

$$J_\ell = \mathbb{E}_{t,\, t' \geq t,\, x} \left[\frac{\partial h_{\text{final},t'}(x)}{\partial h_{\ell,t}(x)} \right], \tag{1}$$

which captures how a perturbation to a layer-ℓ hidden state propagates, on average, to later final-layer representations. The resulting readout is

$$\text{lens}(h_\ell) = \text{softmax}\left(W_U\, N(J_\ell h_\ell)\right). \tag{2}$$

Sorting $\text{lens}(h_\ell)$ over the vocabulary gives the ***J-lens ranked list***. The J-lens authors interpret its top-ranked tokens as those the model is poised to verbalize. The pre-softmax logit for token w is $\langle J_\ell^\top \Gamma^\top u_w, h_\ell \rangle / r(J_\ell h_\ell)$. Softmax preserves order, and the positive denominator is shared across tokens. The rank of w is therefore determined by

$$\langle d_\ell(w), h_\ell \rangle, \qquad d_\ell(w) \triangleq J_\ell^\top \Gamma^\top u_w \in \mathbb{R}^{d_{\text{model}}}, \tag{3}$$

where we call $d_\ell(w)$ the *token representation*. Each $d_\ell(w)$ specifies the direction in hidden-state space that favors token w under the J-lens readout: a larger inner product gives w a higher score.

2.2 Multi-Token Concept Readout

Let $C = (w_1, \ldots, w_n)$ denote a multi-token concept with $n \geq 2$, such as "black hole." The unembedding matrix W_U provides one row per vocabulary token, so J-lens scores the constituent tokens of C individually. A complete concept readout must distinguish compositions such as "black hole" and "black coffee," which share the token "black." Thus, extending a token-level lens to multi-token readout requires not only the complete concept C, but also a representation specific to that concept, $v_C^{(\ell)} \in \mathbb{R}^{d_{\text{model}}}$, at layer ℓ. However, recovering a representation that faithfully captures the complete concept is nontrivial, since it must preserve the meaning of the full token sequence. A sum or mean of constituent token representations assigns the same vector to "black hole" and "hole black."

2.3 Existing Extensions for Multi-Token Concepts

Existing approaches differ in how they provide C and $v_C^{(\ell)}$. Patchscopes and SelfIE use language models to interpret their own hidden states as multi-token concepts with little preparation (Ghandeharioun et al., 2024; Chen et al., 2024). Causal verification of a decoded concept additionally requires recovering a corresponding representation. Template Lens and Oracle Lens provide such representations for readout and intervention, at substantial preparation cost (Gurnee et al., 2026). Template Lens constructs representations for a predefined concept list, while Oracle Lens trains auxiliary models for concept generation and reconstruction. These approaches highlight a trade-off between lightweight adaptation and support for causal verification (Figure 2).

> **Question**
> Can we read multi-token concepts with lightweight adaptation while recovering representations for causal verification?

3 Concept Lens

To address the question above, Concept Lens uses the model's own representations to connect complete concepts with the hidden state being inspected. We describe the method using J-lens.

3.1 Proposing Candidate Concepts

At source layer ℓ_s, **PROPOSE** uses the J-lens ranking of h_{ℓ_s} to guide candidate search. We select tokens from this ranking as possible first tokens of complete concepts. To search beyond the highest-ranked entries, we partition a fixed prefix of the ranking into disjoint ranges and allocate a separate search budget to each range. Each selected seed is placed in the fixed *Scaffold Prompt*.

Scaffold Prompt

```
when model is reading the context below: \n <cloze>.
it will think a concept: <seed>
```

Here <cloze> is the source prompt and <seed> fixes the first token of the candidate. The scaffold can be tailored to the application to guide candidate generation. Within each rank range, we retain high-probability completions until termination or a fixed length limit, then merge and filter them to obtain a fixed candidate set. Rank ranges, decoding budgets, and filtering rules are given in Appendix K.1. **PROPOSE** probabilities guide candidate generation, and **SCORE** measures each candidate's agreement with the source hidden state independently of its proposal probability.

3.2 Recovering and Scoring Candidate Representations

To score a candidate as a concept, **SCORE** constructs a representation from the model's processing of its full token sequence. We place C in a fixed *Carrier Prompt* and extract hidden states from a neutral factual continuation. Reading these later positions follows prior observations that concepts can remain readable in subsequent hidden states (Gurnee et al., 2026). The candidate is the only variable part of the input, allowing its representation to be compared with different source states.

Carrier Prompt

```
Remember the following concept in your mind.
Concept:<C>. Fact: The capital of France is Paris.
```

Let $\tau(C)$ denote the carrier input. We extract hidden states at selected positions $\mathcal{T}$ in the fixed continuation `The capital of France is Paris.`. After subtracting the position-specific background, we pool them as

$$v_C^{(\ell_c)} = \sum_{t \in \mathcal{T}} \alpha_t \left(h_{\ell_c,t}(\tau(C)) - \mu_t \right), \qquad \alpha_t \geq 0, \quad \sum_{t \in \mathcal{T}} \alpha_t = 1. \tag{4}$$

The weights α_t are calibrated independently on single-token concepts using agreement with J-lens token representations (Appendix K.2). Under the local shared-component model in Appendix E.1, the pooled representation can be written as $v_C^{(\ell_c)} = \kappa_C g_C + \eta_C$, where g_C is the concept-related direction and η_C is the remaining pooled residual. If $\kappa_C > 0$ and $\|\eta_C\|_2 < \kappa_C \|g_C\|_2$, then

$$\left\| \mathrm{unit}(v_C^{(\ell_c)}) - \mathrm{unit}(g_C) \right\|_2 \leq \frac{2\|\eta_C\|_2}{\kappa_C \|g_C\|_2}. \tag{5}$$

Thus, recovery is directionally stable when the pooled residual is small relative to the shared concept component. Appendix E.1 gives the derivation and evaluates this premise using held-out single-token concepts.

The carrier representation induces the lens effect $e_C^{(\ell_c)} = J_{\ell_c} v_C^{(\ell_c)}$. For a source layer $\ell_s \neq \ell_c$, we reconstruct a source-layer representation whose J-lens effect matches this carrier effect:

$$v_C^{(\ell_s)} = \arg\min_v \left\{ \left\| J_{\ell_s} v - e_C^{(\ell_c)} \right\|_2^2 + \lambda_{\ell_s}^{\mathrm{ridge}} \|v\|_2^2 \right\}. \tag{6}$$

This transport has a direct spectral interpretation. If $J_{\ell_s} = U \,\mathrm{diag}(\sigma_i) V^\top$, then the transported effect satisfies

$$\widehat{e}_C^{(\ell_s)} = J_{\ell_s} v_C^{(\ell_s)} = P_{\ell_s} e_C^{(\ell_c)}, \qquad P_{\ell_s} = U \,\mathrm{diag}\left(\frac{\sigma_i^2}{\sigma_i^2 + \lambda_{\ell_s}^{\mathrm{ridge}}} \right) U^\top. \tag{7}$$

Hence ridge transport preserves effect directions that are well represented by the source lens while attenuating poorly conditioned directions. Appendix E.2 further separates the resulting discrepancy into carrier-recovery error and ridge bias.

Let $q_{\ell_s} = N(J_{\ell_s} h_{\ell_s})$ be the source query. We score each candidate by directional agreement between its transported effect and q_{ℓ_s}:

$$G_C(h_{\ell_s}) = m_U \left\langle \mathrm{unit}\Big(N(J_{\ell_s} v_C^{(\ell_s)}) \Big), q_{\ell_s} \right\rangle. \tag{8}$$

Here N is the final normalization from Section 2.1, $\mathrm{unit}(z) = z/\|z\|_2$, and m_U is the median norm of the unembedding vectors. Pooling and ridge reconstruction do not preserve a meaningful absolute scale, so **SCORE** compares directions after normalization. The common factor m_U places concept and vocabulary scores on the same numerical scale. Each vocabulary token w is scored by the corresponding normalized rule

$$m_U \langle \mathrm{unit}(u_w), q_{\ell_s} \rangle ,$$

so concepts and vocabulary tokens compete against the same query. This is also the scoring rule used for Template Lens directions in our comparison (Appendix C). The native J-lens ranking in Equation 2, which retains unembedding-row norms, is used to select first-token seeds for **PROPOSE**.

The normalization also clarifies which transport errors can affect the final score. Let $x = N(e_C^{(\ell_c)})$, $\widehat{x} = N(\widehat{e}_C^{(\ell_s)})$, $p = x/\|x\|_2$, and decompose the transport perturbation as $\widehat{x} - x = ap + b$ with $b \perp p$. When $\|\widehat{x} - x\|_2 \le \|x\|_2/2$,

$$\frac{G_C(h_{\ell_s})}{m_U} = q_{\ell_s}^\top p + \frac{q_{\ell_s}^\top b}{\|x\|_2} + \varepsilon, \qquad |\varepsilon| \le 4\|q_{\ell_s}\|_2 \frac{\|\widehat{x} - x\|_2^2}{\|x\|_2^2}. \tag{9}$$

The radial component ap therefore has no first-order effect after normalization; the leading score change is determined by the component b that changes direction and is visible to the source query. Appendix E.3 gives the full finite-error analysis and the corresponding ranking stability result.

We use the same recovered representations for the causal concept swaps in Section 5.2. Beyond the underlying lens, Concept Lens requires lightweight calibration of the carrier reader (Appendix H).

4 Analysis of Concept Lens

We examine the first-token clues used for candidate generation and the concept information preserved in carrier representations. We evaluate instruction-tuned variants of Llama-3.1-8B, Gemma-3-12B, Qwen3-14B, Qwen3.6-27B, and Llama-3.3-70B (Grattafiori et al., 2024; Team et al., 2025; Yang et al., 2025). For readability, we omit instruction-tuning suffixes from model names in the text.

4.1 Can First-Token Clues Guide Concept Completion?

We measure first-token readout across each model's *workspace band*, a middle-layer interval identified using geometric and persistence diagnostics and fixed before cloze evaluation (Appendix B). *First-token Top-10* is the fraction of multi-token examples whose first token appears in the J-lens top-10 at any layer in the band. *Single-token Top-10* measures the corresponding rate for single-token concepts. We stratify both measures by concept length.

For concepts containing two to four tokens under each model's tokenizer, we supply the correct first token and source cloze context, then let the model generate the continuation freely. *Continuation Recovery* measures the fraction whose continuation exactly matches all remaining target tokens.

Figure 3 shows that first-token clues remain readable across concept lengths and that the frozen model completes concepts given the correct clue and source context.

4.2 Do Recovered Representations Preserve Concept Identity?

Our held-out set contains 500 words that are single tokens in all five models (Appendix A.4). Each word's native J-lens direction serves as the reference. We split each word into two tokens (e.g., *airport* → *air*+*port*), recover v_C from the fragments, and rank the 500 reference directions by cosine similarity. We report *Top-1* accuracy and *Wrong Similarity*, the mean cosine similarity to incorrect references. Baselines use the first fragment, the mean of both fragments, or a random representation.

The recovered representations achieve 98.0–99.8% Top-1 accuracy, compared with at most 66.8% for fragment-based baselines, and substantially lower *Wrong Similarity* (Table 1).

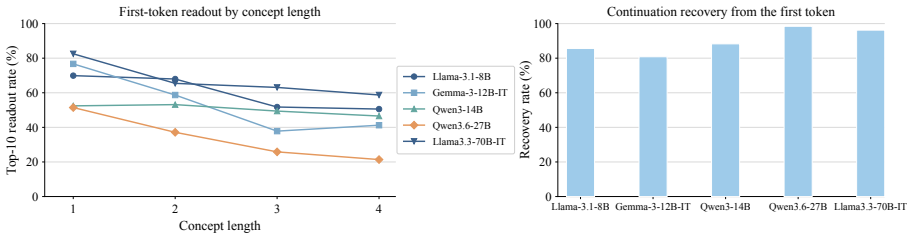

Figure 3: **First-token readout and continuation recovery across five models.** *Left:* Top-10 readout rate by concept length, aggregated across each model's workspace band. For single-token concepts, we use 103 single-token clozes from the official J-lens multi-hop cloze benchmark and evaluate the target token itself; for two- to four-token concepts, we evaluate whether the first target token appears in the J-lens top-10 within the band. Concept length is measured with each model's tokenizer, while the x-axis uses the common labels 1–4. *Right:* Continuation recovery from the first token for two- to four-token concepts. The correct first token is provided with the source cloze context, and recovery succeeds only when all remaining target tokens are recovered.

Table 1: **Recovered representations preserve concept identity.** We split held-out single-token words into two-token forms and retrieve each word's native J-lens direction from a 500-word reference bank. Top-1 (%) measures retrieval accuracy; Wrong Sim. is the mean cosine similarity to incorrect reference directions. Higher Top-1 and lower Wrong Sim. indicate better preservation of concept identity.

	Llama-3.1-8B		**Gemma-3-12B**		**Qwen3-14B**		**Qwen3.6-27B**		**Llama-3.3-70B**	
Method	**Top-1 ↑**	**Wrong Sim. ↓**	**Top-1 ↑**	**Wrong Sim. ↓**	**Top-1 ↑**	**Wrong Sim. ↓**	**Top-1 ↑**	**Wrong Sim. ↓**	**Top-1 ↑**	**Wrong Sim. ↓**
Ours	99.0	-0.0096	98.0	0.0024	99.2	-0.0015	99.4	-0.0088	99.8	0.0114
First	55.6	0.0675	29.0	0.1177	50.0	0.0501	58.2	0.0649	48.2	0.0827
Mean	66.0	0.0699	23.4	0.1246	52.8	0.0564	66.8	0.0757	52.4	0.0886
Random	0.2	0.0001	0.2	0.0004	0.0	-0.0000	0.0	0.0001	0.4	0.0001

Key Finding

First-token clues remain readable across concept lengths, and the model completes concepts when given the correct clue and source context. Carrier states yield representations that preserve concept identity in the controlled reconstruction task.

5 Experiments

Section 4 examines the two stages of Concept Lens. We now evaluate the complete method with J-lens and R-lens across the same five models. We test whether Concept Lens identifies latent intermediate multi-token concepts (Section 5.1) and whether the recovered representations can redirect model computation (Section 5.2). Appendix I defines the common interface used for the two lenses.

5.1 End-to-End Readout

We evaluate the pipeline on 2,400 directed clozes from 1,200 counterfactual pairs across five models. Appendix A describes the benchmark construction, composition, and data splits. Each cloze refers to its intermediate multi-token concept only through indirect clues. We report **Coverage**, the fraction of clozes whose target is proposed at any layer within the model-specific workspace band, and **Rank@10**, the fraction whose target is ranked in the top-10 at any layer within that band. Both metrics are means of per-cloze success indicators. Following the J-lens evaluation protocol, we use the pre-specified workspace band, fixed before task evaluation, and run our **PROPOSE/SCORE** pipeline across the layers in that band (Appendix B).

We use two settings to distinguish candidate availability from scoring. In *end-to-end*, the pipeline must propose and then rank the target. In the *oracle-candidate setting*, we add the annotated target to each layer's candidate set to evaluate **SCORE** independently of whether **PROPOSE** finds the target. Comparing the settings measures how target availability affects end-to-end readout. The *Random* reference reuses the J-lens candidate bank, uniformly and independently permutes the joint ranking of vocabulary tokens and available candidates at each layer, and applies the same workspace-band aggregation, so its Coverage matches J-lens and its Rank@10 values are analytic expectations.

Table 2: **Multi-token concept readout across five models.** Across 2,400 clozes per model, **Coverage** measures target availability and **Rank@10** measures successful top-10 retrieval at any layer within the model-specific workspace band. *End-to-end* evaluates the full pipeline; the *Oracle-candidate setting* adds the target to each layer's candidate set to isolate **SCORE** from **PROPOSE**.

	Llama-3.1-8B		Gemma-3-12B		Qwen3-14B		Qwen3.6-27B		Llama-3.3-70B	
Method	**Coverage**	**Rank@10 ↑**	**Coverage**	**Rank@10 ↑**	**Coverage**	**Rank@10 ↑**	**Coverage**	**Rank@10 ↑**	**Coverage**	**Rank@10 ↑**
End-to-end: Proposal + Scoring										
Ours (J-lens)	49.2	44.3	40.4	37.4	41.6	31.5	30.1	23.1	56.4	46.5
Ours (R-lens)	**63.6**	**61.9**	**60.8**	**52.6**	**65.6**	**55.7**	**43.0**	**40.3**	**75.5**	**61.8**
Template Lens	25.7	19.9	25.7	22.2	25.7	21.0	25.7	21.9	25.7	23.5
Random	49.2	0.03	40.4	0.01	41.6	0.01	30.1	0.0	56.4	0.08
Oracle-candidate setting: Scoring only										
Ours (J-lens)	100.0	91.9	100.0	**90.9**	100.0	**98.8**	100.0	80.2	100.0	**83.3**
Ours (R-lens)	100.0	**95.9**	100.0	90.6	100.0	90.5	100.0	**90.6**	100.0	81.8
Template Lens	100.0	67.5	100.0	69.9	100.0	73.2	100.0	78.6	100.0	80.8
Random	100.0	0.20	100.0	0.14	100.0	0.18	100.0	0.24	100.0	0.33

Table 2 shows that **SCORE** ranks the target in the top-10 for 80.2–98.8% of clozes when the target is added to the candidate set, compared with 23.1–61.9% for the complete pipeline. End-to-end Rank@10 closely tracks Coverage while oracle Rank@10 remains high, identifying candidate coverage as a major bottleneck. The ranking also depends on the source hidden state. With the candidate set and recovered directions held fixed, changing only the source state sharply changes concept selection across counterfactual inputs (Appendix D).

The procedure successfully generalizes across different readout types. Both J-lens and R-lens outperform Template Lens, whose predefined concept list covers 25.7% of targets and whose end-to-end Rank@10 is only 19.9–23.5%. We give Template Lens a fixed bank of 38,400 concepts, three times the 12,800-concept configuration, to make this coverage comparison conservative (Appendix C). R-lens gains 15.2–24.2 points over J-lens in end-to-end Rank@10, consistent with the stronger early-layer readouts reported for R-lens (Blank et al., 2026). The end-to-end J/R difference combines changes in their seed rankings and scoring geometries; the oracle-candidate setting evaluates the ranking stage once the target is available.

5.2 Causal Concept Swap

We test whether recovered representations can redirect model computation on paired clozes with the same relation. The source cloze requires C, and its partner requires D. **SCORE** recovers representations of these annotated concepts, which we transport to fixed intervention layers. *Full replacement* removes the C component from the source hidden states and adds the D direction. We keep the source prompt unchanged and test whether the answer shifts toward the partner answer. Addition strength is measured on the clean partner activation, without fitting to intervention outcomes. *Add-only* and *delete-only* controls isolate the two operations; magnitude-matched random directions test specificity. We evaluate both directions of each pair, yielding 2,400 directed trials. Appendix F gives the full protocol.

Table 3: **Recovered concept representations causally redirect intermediate computation.** Controlled swaps on paired clozes requiring different concepts. All metrics are percentages of trials. **P@10**: partner answer appears in the edited top-10; **S@10**: source answer remains in the edited top-10; **CS@10**: partner ranks in the top-10 and source ranks outside the top-10. *Full replacement* removes the source direction and adds the transported partner direction; *add-only* and *delete-only* isolate the two operations. *Random* uses magnitude-matched directions in the add-only setting.

	Llama-3.1-8B			Gemma-3-12B			Qwen3-14B			Qwen3.6-27B			Llama-3.3-70B		
Method	**P@10 ↑**	**S@10 ↓**	**CS@10 ↑**	**P@10 ↑**	**S@10 ↓**	**CS@10 ↑**	**P@10 ↑**	**S@10 ↓**	**CS@10 ↑**	**P@10 ↑**	**S@10 ↓**	**CS@10 ↑**	**P@10 ↑**	**S@10 ↓**	**CS@10 ↑**
Full replacement															
Ours (J-lens)	45.5	17.2	38.2	48.0	11.8	44.2	48.6	16.2	41.9	27.6	31.8	15.7	60.2	3.8	59.0
Ours (R-lens)	50.0	14.0	43.8	46.4	13.4	42.0	50.0	16.5	43.2	30.6	29.5	18.9	60.9	4.3	59.4
Template Lens	27.9	81.1	7.2	12.2	60.8	4.3	14.8	91.6	2.1	6.7	50.0	1.4	16.8	41.0	9.8
Random	3.6	89.6	1.6	3.1	93.1	1.4	4.0	97.4	1.5	4.8	96.8	1.1	1.7	92.7	0.8
Operation controls															
Ours (J-lens), add only	39.6	38.7	22.2	40.9	26.2	31.8	42.8	38.6	24.8	20.1	48.7	6.8	58.0	8.3	54.9
Ours (J-lens), delete only	4.0	33.1	1.7	3.9	31.8	1.1	4.5	30.0	1.0	4.8	43.2	1.2	5.4	41.7	1.5
Ours (R-lens), add only	44.3	35.1	28.2	38.8	28.2	29.0	45.0	38.1	26.8	22.8	47.6	9.0	58.3	8.8	54.9
Ours (R-lens), delete only	3.7	32.5	1.4	3.7	33.1	0.9	4.4	29.6	1.0	4.9	42.8	1.2	5.3	42.6	1.5

Concept Lens promotes the partner answer in 27.6–60.9% of trials and achieves 15.7–59.4% CS@10, compared with 1.4–9.8% CS@10 for Template Lens (Table 3). Full replacement yields the highest clean-swap rate among the operation controls across all five models and both underlying lenses. Add-only promotes the partner answer but leaves more source answers intact, whereas delete-only suppresses the source answer but rarely promotes the partner answer (3.7–5.4% P@10). Random directions yield few clean swaps, supporting the specificity of the recovered directions.

6 APPLICATIONS

The controlled benchmark provides annotated intermediate concepts for quantitative evaluation. We next use Concept Lens to compare model interpretations across matched inputs in three settings spanning logical consistency, problem structure, and pragmatic interpretation.

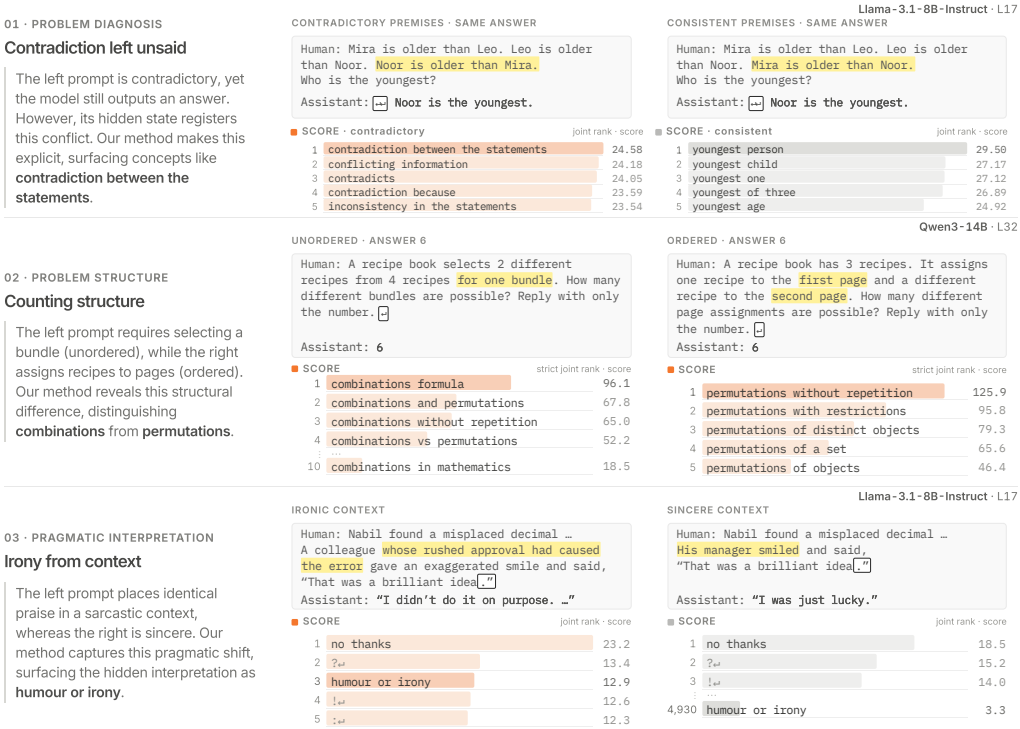

Figure 4: **Concept Lens reveals internal interpretations behind similar final responses.** The top two pairs produce identical answers, while the bottom pair gives similar replies with different pragmatic meanings. The panels show latent contradictions (top), counting structures (middle), and pragmatic interpretation (bottom). The top and bottom examples use Llama-3.1-8B at L17; the middle uses Qwen3-14B at L32. Bar charts display the top-ranked concepts extracted at the boxed token; yellow highlights mark the context variations.

Figure 4 shows these distinctions. In the first pair, both versions yield the same answer "Noor is the youngest," yet the source states differ: the contradictory context shifts the readout toward *contradiction between the statements*, while the consistent context favors *youngest person*. In the second, both prompts yield the answer 6, yet the source states differ: one is associated with *combinations*, the other with *permutations*, exposing different reasoning structures behind the shared final output. In the third, the same surface phrase is interpreted differently under different contexts: *humour or irony* ranks 3rd in the sarcastic context but 4,930th in the sincere one, showing that the source state encodes pragmatic interpretation beyond what the phrase itself determines. Thus, multi-token readout distinguishes internal interpretations that remain implicit in similar final responses.

> **Implication**
>
> Multi-token readout provides a more complete conceptual view than isolated token-level readouts, making otherwise hidden distinctions easier to inspect.

7 Related Work

Concept Lens connects lens-based concept scoring with language-model interpretations of activations. Template and Oracle Lens associate concepts with representations for scoring and intervention (Appendix G.1). Patchscopes and SelfIE generate multi-token descriptions from injected activations (Ghandeharioun et al., 2024; Chen et al., 2024). Recent work identifies linguistic-prior bias in Patchscopes (Gong et al., 2026) and sensitivity to prompts and injection settings in related introspection tasks (Lindsey, 2026; Singh et al., 2026). Natural language autoencoders connect generated descriptions to activation reconstruction (Fraser-Taliente et al., 2026). Concept Lens separates candidate generation from source-conditioned scoring, with reliability assessed through representation recovery and fixed-candidate source-state swaps. Studies of lexical representations and latent reasoning motivate complete concepts as readout units (Appendix G.6). Appendix G expands these comparisons and discusses feature dictionaries and activation steering.

8 Discussion and Limitation

Controlled comparisons show that Concept Lens selects concepts according to the hidden state being inspected. With candidates and representations fixed, replacing the source state shifts the ranking toward the paired concept (Appendix D.3). On fixed candidate banks containing naturally proposed targets, **SCORE** achieves higher Rank@10 than continuation likelihood across five models (Appendix D.2). The readout also distinguishes concepts sharing a first token (Appendix D.4), supporting its ability to resolve complete concepts. The causal swaps complement these readout tests by evaluating the behavioral effects of the concept representations. Together, the experiments connect concept selection to the hidden state being inspected and to the model's subsequent computation.

The gap between end-to-end and oracle-candidate results identifies target coverage as a major bottleneck. Appendix D.1 compares two proposal strategies using the same scoring procedure, showing how candidate generation affects retrieval. Higher target coverage at a fixed candidate budget offers a route to stronger readout with the same carrier reader and scoring rule. The proposal scaffold can also guide candidate generation toward concepts relevant to a particular analysis. Concept Lens reuses the underlying LLM and calibrates a single carrier reader across multiple layers. Given an existing lens, calibration on Qwen3.6-27B, for example, takes only 0.1 A100-hours for all 52 layers. Template Lens spends 20–30 A100-hours plus \$1,500–\$2,000 in API costs to construct its concept bank. Oracle Lens preparation is estimated to require $2{,}000$–$4{,}000$ A100-hours for a single-layer configuration. Concept Lens constructs candidate representations at query time and reads all 52 layers in about 4 seconds (Appendix H).

8.1 Limitations

Candidate coverage limits end-to-end concept readout. Stronger proposers could improve coverage while retaining the same carrier reader and scoring rule. Our direct comparison with Oracle Lens covers the public Qwen3.6-27B configuration at one shared activation layer (Appendix J). A broader comparison could evaluate both methods over matched layer ranges across several models. Tracking concept recovery across depth would show where each readout is most effective, while reporting preparation costs for each configuration would connect readout quality with adaptation cost. Ground truth remains a broader challenge for activation readouts, as the intermediate concepts implied by our cloze tasks are not guaranteed to be represented in the model's computation. Our controlled split-token test provides a well-defined reference for representation recovery.

9 Conclusion

Concept Lens reads latent concepts from LLM hidden states with lightweight adaptation. The method uses the frozen model to propose complete concepts and scores their recovered representations in an existing lens readout space. Experiments across five models demonstrate effective concept retrieval, while causal interventions support the connection between the readout and subsequent model behavior. The application cases reveal different internal interpretations behind similar final answers, illustrating how concept readout makes implicit aspects of model computation accessible.

AI USE STATEMENT

In this work, we used generative AI tools for generating and processing parts of the synthetic dataset. The authors determined the intermediate concepts, relations, templates, and generation constraints, and generative AI was used to produce candidate clue sentences under these constraints. We also used generative AI tools for literature search, language editing, and assistance with drafting methodological explanations, comparison protocols, and supporting mathematical analysis. AI assistance included checking implementation details and aggregating saved experimental results. All AI-generated dataset content was subsequently reviewed and processed by the authors. All code and data used for the reported results were checked by at least two authors. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, code, data, and experimental results.

REPRODUCIBILITY STATEMENT

We have made reproducibility a central part of this work. We will release all code, datasets, configuration files, and implementation details required to reproduce our results through an anonymous repository, and we will provide the exact sources and versions of all model weights and external resources used in our experiments. The appendix describes the algorithms, data construction procedure, evaluation protocol, hyperparameters, and other implementation details needed to reproduce the experiments. We will also release the processed datasets and relevant intermediate artifacts so that individual stages of the pipeline can be checked independently. All experiments in this paper can be reproduced with approximately 500 A100 GPU-hours of computation. We have aimed to make every reported result independently reproducible and to minimize differences caused by missing implementation details or undocumented experimental settings.

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, 2025. URL https://arxiv.org/abs/2303.08112.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14113–14130. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.781.

Camila Blank, Agam Bhatia, and Neel Nanda. R-lens: Making J-lens more faithful on early layers. AI Alignment Forum, August 2026. URL https://www.alignmentforum.org/posts/nv8oedrnLXKRzNEL9/r-lens-making-j-lens-more-faithful-on-early-layers.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. SelfIE: Self-interpretation of large language model embeddings. 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. 7373–7388. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24ao.html.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Sheridan Feucht, David Atkinson, Byron C Wallace, and David Bau. Token erasure as a footprint of implicit vocabulary items in LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9727–9739, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.543. URL https://aclanthology.org/2024.emnlp-main.543/.

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.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.751.

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.

Xilin Gong, Shu Yang, Zehua Cao, Lynne Billard, and Di Wang. Faithful-patchscopes: Understanding and mitigating model bias in hidden representations explanation of large language models, 2026. URL https://arxiv.org/abs/2602.00300.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell,

Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiao-

jian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

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.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From tokens to words: On the inner lexicon of llms. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 70068–70092, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/aef75887979ae1287b5deb54a1e3cbda-Paper-Conference.pdf.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

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.

Jack Lindsey. Emergent introspective awareness in large language models, 2026. URL https://arxiv.org/abs/2601.01828.

Samuel Marks, Can Rager, Eric Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 23888–23923, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3ba4d47a83e498c2b1a0868cba20f6de-Paper-Conference.pdf.

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.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Victor Morand, Josiane Mothe, and Benjamin Piwowarski. On the representations of entities in auto-regressive large language models. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 433–451, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.blackboxnlp-1.25.

Neel Nanda. A review of anthropic's global workspace paper. LessWrong, July 2026. URL https://www.lesswrong.com/posts/zFJ3ZdQwrTWE9jT5S/a-review-of-anthropic-s-global-workspace-paper.

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.), *Conference on Computational Natural Language Learning*, pp. 548–560. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.conll-1.37.

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/.

Shashwat Singh, Tal Linzen, and Shauli Ravfogel. Can llms introspect? a reality check, 2026. URL https://arxiv.org/abs/2605.26242.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

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, 2024. URL https://arxiv.org/abs/2308.10248.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10210–10229. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.550.

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, 2025. URL https://arxiv.org/abs/2310.01405.

APPENDIX

A Benchmark Construction, Composition, and Data Splits **17**
A.1 Benchmark Construction . . . 17
A.2 Benchmark Composition . . . 18
A.3 Strict-Fact Answer-Cued Prompt Diagnostic . . . 19
A.4 Calibration and Reconstruction Sets . . . 20

B Workspace Band and Evaluation Protocol **21**
B.1 Workspace Band Selection . . . 21
B.2 Layer-wise Concept Readout . . . 22
B.3 Evaluation Protocol . . . 23

C Template Lens Construction and Vocabulary Coverage **30**
C.1 Template Lens Construction . . . 30
C.2 Vocabulary Coverage . . . 30
C.3 Oracle-Candidate Setting . . . 32

D Further Evaluation of Concept Readout **32**
D.1 Candidate Generation Strategies . . . 33
D.2 Scoring a Fixed Candidate Set . . . 34
D.3 Concept Selection Across Source States . . . 35
D.4 Distinguishing Concepts with the Same First Token . . . 36
D.5 Implications for Concept Readout . . . 37

E Carrier Recovery, Effect Transport, and Ranking Stability **37**
E.1 Carrier Recovery . . . 38
E.2 Spectral Effect Transport . . . 38
E.3 Query-Sensitive Top-k Stability . . . 39
E.4 Empirical Validation . . . 40

F Causal Intervention Protocol **42**
F.1 Intervention Setup . . . 42
F.2 Evaluation . . . 42

G Relationship to Existing Readout Methods **43**
G.1 Template Lens and Oracle Lens . . . 43
G.2 Patchscopes and SelfIE . . . 43
G.3 Natural Language Autoencoders . . . 44
G.4 Sparse Autoencoders and Feature Dictionaries . . . 44
G.5 Activation Steering and Concept Representations . . . 44
G.6 Scope of the Contribution . . . 45

H Computational Footprint and Online Efficiency **45**
H.1 One-Time Preparation . . . 45
H.2 Online Readout Cost . . . 46

I A Pluggable Proposal and Lens Interface **46**
I.1 Separating Candidate Proposal from Readout Geometry . . . 46
I.2 R-lens Readout Geometry . . . 47
I.3 Matched J-lens and R-lens Evaluation . . . 48
I.4 Held-Out Cross-Lens Compatibility Check . . . 49

J Comparison with Oracle Lens **50**
J.1 Activation Interface . . . 50
J.2 Matched Direct Comparison . . . 50
J.3 Readout Design and Model-Specific Preparation . . . 51

K Implementation Details **51**
K.1 Candidate Generation . . . 51
K.2 Carrier Calibration and Vector Construction . . . 51
K.3 Efficient Inference . . . 52
K.4 Measured Effect of Efficient Execution . . . 53

A Benchmark Construction, Composition, and Data Splits

A.1 Benchmark Construction

We construct a benchmark of paired multi-hop clozes for reading implicit multi-token concepts. Each directed cloze contains a source prompt x, an intermediate concept C, and a target answer y. Solving x requires the model to infer C and apply a specified relation to C. The source prompt does not name C, so a successful readout must recover information that is useful for the task without copying the target string. The two directions in a counterfactual pair share the relation and surface format while using different concepts and answers. This pairing gives us a controlled comparison for both representation readout and causal concept replacement.

Construction begins with the intermediate concepts and their relations. We tokenize each candidate concept after one leading space, with special tokens disabled, and assign the concept to an intended two-, three-, or four-token group. GPT-5.5 then generates paired, self-contained clozes from the supplied concepts, relation, and answers. The generation instruction asks for descriptions that identify the concepts in ordinary usage while keeping their names and answers out of the prompt. Each returned record contains both prompts and their annotations, allowing the two directions to be checked together. Recorded usage across generation and validation is approximately 17.25M input tokens and 36.91M output tokens.

The data-construction prompt used for this step is reproduced below. It fixes the paired surface form and makes the required relation, hidden concepts, and answers explicit in the construction record.

Data-Construction Prompt

```
You are writing paired factual cloze items for a latent multi-token
concept experiment.

For each side, a reader must first infer a hidden English concept
from a semantic description and then retrieve one fixed property of
that concept.  Write only the prompt, never the answer.  Both prompts
must follow this surface form:

Fact:  The <relation> of the <description> is

Each prompt begins with "Fact:  The "and ends with "is".  Write one
fluent English sentence.  Use a specific factual description of the
hidden concept.

Requirements for both prompts:
- The description must identify the supplied concept in ordinary
usage.
- Do not name the concept, any alias, or a substring that gives the
name away.
- The requested property must yield exactly the supplied answer.
- Do not print the answer in the prompt.
- Preserve the supplied relation, concept, and answer.
- Both sides of a pair must use the same relation.

Use the concepts below:
concept1:<concept1>
concept2:<concept2>

Return only one JSON object:
{ "prompt1":  "...", "prompt2":  "...", "concept1":  "...",
"concept2":  "...", "answer1":  "...", "answer2":  "..." }
```

Filtering checks whether each record preserves the intended inference problem. Automatic checks remove malformed records, incomplete pairs, duplicate directions, explicit mentions of the intermediate concept or answer, and violations of the required relation. Surface variations of the requested template are retained when they preserve the relation, description, and answer. We inspect the retained records for factual coherence, consistency of the paired relation, and the connection between the description, concept, and answer. A fluent sentence alone is insufficient; the description must support the annotated concept, and the requested property must lead to the annotated answer.

Construction and filtering are independent of source activations, lens ranks, candidate proposals, recovered representations, and readout scores.

The paired design gives the benchmark a common unit for readout and intervention. In readout evaluation, the annotation identifies the concept that the method should recover from each prompt. In a causal swap, the partner supplies an alternative concept and a corresponding answer under the same relation. Holding the relation fixed makes the behavioral prediction concrete. A successful replacement should move the source prompt toward the partner answer. The same pair therefore connects an interpretable readout to a testable change in the model's continuation, as developed in Appendix F.

A.2 Benchmark Composition

The final benchmark contains $1{,}200$ counterfactual pairs, or $2{,}400$ directed clozes. Each directed cloze supplies one annotated intermediate concept and is evaluated at the selected source layers. Exact case-insensitive matching finds no complete intermediate-concept string in its own source prompt, so the target cannot be recovered by copying a matching span from the input. After case and whitespace normalization, the benchmark contains $2{,}113$ unique intermediate concepts, of which $1{,}912$ occur once.

Token length is defined at the point where a concept would naturally appear in a continuation. For model m, we use

$$n_m(C) = |\mathrm{Tok}_m(\quad \| \, C)|\,, \tag{10}$$

where the leading space reproduces ordinary continuation tokenization and special tokens are disabled. Every concept in the final benchmark spans between two and four tokens under each of the five model tokenizers, although a concept's exact length can change with the tokenizer. Table 4 reports the distribution over all directed clozes for every model. This convention keeps the semantic task fixed across models while allowing tokenization to vary. The tables count directed clozes, so a concept that appears in both directions contributes once per evaluated direction.

Table 4: **Continuation-token counts in the final benchmark.** Counts and percentages use all $2{,}400$ directed clozes for every tokenizer. Each concept is tokenized with one leading space and no special tokens.

Tokenizer	2 tokens	3 tokens	4 tokens	Total
Gemma-3-12B	1,954 (81.42%)	349 (14.54%)	97 (4.04%)	2,400
Llama-3.1-8B	1,716 (71.50%)	512 (21.33%)	172 (7.17%)	2,400
Qwen3-14B	1,716 (71.50%)	510 (21.25%)	174 (7.25%)	2,400
Qwen3.6-27B	1,726 (71.92%)	515 (21.46%)	159 (6.62%)	2,400
Llama-3.3-70B	1,716 (71.50%)	512 (21.33%)	172 (7.17%)	2,400

Continuation-token length and written-word count describe different properties of a concept. Here, written-word count uses whitespace-delimited words. In the final benchmark, 36.54% of directed clozes use a single written word that spans multiple vocabulary tokens; the remaining 63.46% use two to four written words. The data therefore cover both subword composition, illustrated by *black-mail*, and multi-word composition, illustrated by *black hole*. Both cases require a readout that treats several tokens as one semantic unit. Table 5 separates these surface forms so that the role of token boundaries remains clear when interpreting the reconstruction and readout experiments.

Table 5: **Intermediate surface forms in the final benchmark.** Counts are over $2{,}400$ directed clozes. The final row combines the two-, three-, and four-word categories.

Written-word count	Clozes	Share (%)
One word (single-word multi-token)	877	36.54
Two words	1,396	58.17
Three words	101	4.21
Four words	26	1.08
Multi-word concepts (two to four words)	1,523	63.46

The final benchmark spans a broad range of semantic domains. To describe this coverage, we assign every directed cloze to one mutually exclusive broad category using its annotated intermediate

concept and requested relation, with the resulting distribution shown in Table 6. Abstract concepts, people, places, and organizations account for much of the benchmark, alongside scientific, cultural, and everyday concepts. This breadth matters because a useful concept readout should handle both named entities and ordinary semantic categories. The taxonomy is a post-construction descriptive audit: it does not enter construction, filtering, candidate generation, model selection, or evaluation. Each directed cloze receives exactly one category.

Table 6: **Semantic composition of the final benchmark.** Each of the 2,400 directed clozes is assigned to one mutually exclusive category after construction. The categories summarize the range of concepts and relations.

Category	Clozes	Share (%)
Abstract concepts, processes, and properties	416	17.33
People and characters	413	17.21
Geography and places	253	10.54
Organizations and institutions	245	10.21
Life, health, and medicine	210	8.75
Physical science, chemistry, and materials	179	7.46
Science, computing, and technology	155	6.46
Objects, engineering, and infrastructure	138	5.75
Culture, media, and works	134	5.58
Society, economics	79	3.29
Food and agriculture	76	3.17
Environment, Earth, and space	58	2.42
Sport, games, and leisure	44	1.83
Total	2,400	100.00

A.3 Strict-Fact Answer-Cued Prompt Diagnostic

The prompt-format diagnostic asks whether the same intermediate concept can be read under different ways of requesting the final answer. The structured annotation lets us vary the prompt while preserving the description, relation, and answer. Each rendered example retains the intended inference chain

$$\text{description} \to C \to \text{relation}(C) \to y, \tag{11}$$

while the intermediate concept C remains absent from the source text. We retain paired clozes whose two directions match the strict Fact template. Its prefix is `Fact: The`, followed by the relation, `of the`, the description, and `is`. The resulting cohort contains 759 pairs, or 1,518 directed clozes, that can be converted mechanically into the three forms below. The remaining 441 pairs (882 directed clozes) deviate from the strict template only in surface wording; this variation does not change the meaning of the cloze, so these pairs remain in the main benchmark and are excluded here only because they cannot be converted mechanically. Every form preserves the task and ends with `Please provide only the answer:`. An exact-string audit finds no full-concept mentions in any rendered view.

T1: Question Form

```
Question: What is the <relation> of the <description>?
Please provide only the answer:
```

T2: Description-First Instruction

```
Given the <description>, provide its <relation>.
Please provide only the answer:
```

T3: Explicit Retrieval Instruction

```
From the description below, identify the <relation>.
Description: <description>
Please provide only the answer:
```

We run the fixed J-lens proposal and ranking pipeline on all $1{,}518$ directed clozes for Llama-3.1-8B, Gemma-3-12B, and Qwen3-14B. Table 7 follows the readout through three stages. First-token Top-10 measures the clue supplied by the lens, Coverage measures whether proposal completes the target, and v_C Rank@k measures the final joint ranking. Reporting these stages together shows where a change in wording affects the pipeline. All template comparisons use the same concepts, relations, and answers within the mechanically convertible cohort.

Table 7: **Readout under three answer-cued prompt forms.** All entries are percentages on the same 759 pairs ($1{,}518$ directed clozes) in this diagnostic cohort. Each template preserves the description and relation while keeping the intermediate concept unnamed.

Model	Template	First Token Top-10	Coverage	v_C Rank@1	v_C Rank@10
Llama-3.1-8B	T1: Question	42.29	35.70	5.07	30.63
Llama-3.1-8B	T2: Description-first	52.24	39.79	6.46	33.27
Llama-3.1-8B	T3: Explicit retrieval	59.68	49.34	9.29	43.08
Gemma-3-12B	T1: Question	72.00	58.10	22.66	56.32
Gemma-3-12B	T2: Description-first	67.26	54.81	21.87	53.16
Gemma-3-12B	T3: Explicit retrieval	63.37	45.78	13.37	42.95
Qwen3-14B	T1: Question	3.49	2.77	0.00	0.26
Qwen3-14B	T2: Description-first	18.51	14.03	0.26	4.15
Qwen3-14B	T3: Explicit retrieval	10.80	8.89	0.00	1.52

The three models respond differently to the prompt forms. Llama-3.1-8B has its strongest first-token, proposal, and final readout results under T3, while Gemma-3-12B favors T1. Qwen3-14B has lower absolute rates in this answer-cued diagnostic and is more sensitive to prompt form. The templates follow the original J-lens template family; under these prompts, Qwen3-14B may not consistently form the required intermediate concept. T2 nonetheless provides its highest Coverage and final Rank@10. For each model, the highest Coverage and highest final Rank@10 occur under the same template. The agreement links the final result to the availability of useful concept candidates. The diagnostic also shows why prompt form belongs in a readout protocol. A fixed semantic annotation makes the forms comparable, while the stage-level measurements reveal how the model processes each form.

A.4 Calibration and Reconstruction Sets

Complete intermediate concepts in the multi-token benchmark have no native J-lens representation. We therefore use a separate singleton resource to calibrate the carrier reader and evaluate recovery against an independent reference. A singleton concept w occupies one vocabulary token and has a native direction $d_\ell(w)$. Forced fragmentation presents the same written concept as several tokens while retaining that native direction as a reference. Because fragmentation preserves the decoded text, the experiment can test whether the recovered representation retains concept identity as token boundaries change. The reference comes from the lens and is withheld from the recovery procedure.

We split a pool of 800 canonical singleton concepts into a 300-concept calibration set and a 500-concept held-out reconstruction set using random seed `20260903`. The pool uses the same leading-space continuation convention as the main benchmark. Every concept in both partitions occupies exactly one continuation token under all five tokenizers, so every model uses the same 300 calibration concepts and the same 500 reconstruction concepts. No evaluation result determines the split or the carrier-reader hyperparameters.

The calibration partition determines the carrier layer, suffix positions, background means, and pooling weights described in Appendix K.2. Native singleton directions provide a common reference for selecting positions that preserve concept information. The resulting carrier reader then remains fixed for held-out reconstruction, multi-token readout, and causal intervention. Calibration and reconstruction concepts are disjoint, and neither partition has exact intermediate-string overlap with the multi-token benchmark. The evaluation therefore tests how a carrier reader calibrated on familiar token-level directions transfers to previously unseen concept strings and fragmented inputs.

The reconstruction set evaluates each singleton using a forced token sequence that decodes to the original surface form. The main reconstruction experiment uses two-token fragmentations; the supplementary procedure also supports three- and four-token fragmentations whenever an exact decomposition is available. Recovery receives only the fragmented carrier input. Cosine similarity mea-

sures directional agreement, and retrieval among held-out singleton references measures whether the recovered representation identifies the intended concept. These complementary measurements connect geometric accuracy to concept identity. Table 8 summarizes the resources and their roles.

Table 8: **Summary of data resources used in the study.** The paired-cloze benchmark supports composition, end-to-end readout, and causal evaluation; the strict-Fact cohort varies prompt surface form on mechanically convertible items; and the two singleton resources are reserved for calibration and held-out representation reconstruction.

Resource	Items	Evaluated examples	Purpose
Paired-cloze benchmark	1,200 pairs	2,400 clozes	Composition, end-to-end readout, and causal evaluation
Strict-Fact answer-cued cohort	759 pairs	4,554 renderings	Three prompt forms for the convertible cohort
Singleton calibration set	300 concepts	300 concepts	Fix carrier-reader components before evaluation
Held-out reconstruction set	500 concepts	500 per model	Validate recovered directions against native J-lens representations

B Workspace Band and Evaluation Protocol

Concept readout depends on where the model is in its computation. Early residual states are closely tied to the input tokens, while late states increasingly support the next output token. We define a *workspace band* as a contiguous interval between these regimes with coherent J-lens geometry and persistent readout. The band provides a common region in which to compare methods for reading intermediate concepts. Its definition uses properties of the lens and held-out text, allowing us to choose the evaluation interval before inspecting the multi-token task results.

We select a band $\mathcal{W}_m$ for each model from structural and dynamical evidence in Appendix B.1. The lower boundary follows the rapidly changing early regime, and the upper boundary precedes the final decoding transition. We use vocabulary-centered CKA together with persistence, dimensionality, and next-token alignment so that a single noisy diagnostic cannot determine a boundary. Band selection is completed before target ranks, candidate coverage, or intervention results are inspected. Table 9 lists the selected intervals. Layer indices are inclusive throughout this appendix.

Table 9: **Model-specific workspace bands.** Layerwise J-lens geometry and diagnostics determine $\mathcal{W}_m$ before multi-token readout evaluation. Each band follows the initial lexical transition and precedes the final decoding transition.

Model	Blocks	$\mathcal{W}_m$	Boundary evidence
Llama-3.1-8B	32	4–29	Early lexical transients have subsided; the final prediction surge and endpoint collapse in persistence are excluded.
Gemma-3-12B	48	6–44	The band begins after the initial concentrated regime and ends before the terminal decoding bend and persistence drop.
Qwen3-14B	40	6–37	A coherent middle geometry is retained while the final sharp prediction-alignment regime is excluded.
Qwen3.6-27B	64	8–59	The band captures the long persistent middle regime, including its non-monotonic concentration changes, but not the final output transition.
Llama-3.3-70B	80	34–75	A delayed, sustained intermediate regime begins only in the latter half of the network; the final decoder transition is excluded.

B.1 Workspace Band Selection

The structural analysis asks how the geometry of token directions changes with depth. For each layer ℓ, let D_ℓ stack the J-lens representations $d_\ell(w)$ over the vocabulary, and let X_ℓ be the matrix after subtracting the mean row. We compare two layers with linear centered kernel alignment

(CKA) (Kornblith et al., 2019),

$$\mathrm{CKA}(\ell, k) = \frac{\|X_\ell^\top X_k\|_F^2}{\|X_\ell^\top X_\ell\|_F \, \|X_k^\top X_k\|_F}. \tag{12}$$

Brighter heatmap entries indicate that the two layers organize vocabulary directions similarly. A contiguous bright block identifies a region with coherent geometry across several layers. Centering removes the mean vocabulary direction, so the comparison emphasizes differences among token directions. CKA summarizes the structure of those differences and is useful for locating transitions between processing regimes. We combine the structural signal with the readout diagnostics below to determine which coherent region also exhibits persistent intermediate content.

We complement CKA with four readout diagnostics on held-out natural-text positions. The diagnostics measure J-lens next-token prediction accuracy, excess kurtosis of vocabulary logits, top-1 log-probability persistence relative to a within-sequence shuffled-position null, and effective J-space dimensionality. High next-token alignment near the final layer signals output specialization. Persistent top-1 content measures whether the readout remains stable across positions, while changes in effective dimensionality reveal a reorganization of the lens directions. Together, the diagnostics separate a sustained intermediate regime from a transient activation peak.

Kurtosis measures vocabulary-logit concentration. Early concentration can reflect the current token embedding or anisotropy in the unembedding and J-lens geometry. The early kurtosis spike in Llama-3.1-8B and the elevated initial concentration in Gemma-3-12B precede sustained persistence and intermediate geometry. Late increases can accompany the transition to output decoding. We therefore treat kurtosis as supporting evidence and interpret it together with CKA, persistence, dimensionality, and prediction alignment.

The joint evidence places the intermediate regime at different depths across models. Llama-3.1-8B largely completes the initial transition by layer 4 and retains intermediate geometry until the final layers. Gemma and Qwen3-14B enter the regime early and retain the geometry over intervals 6–44 and 6–37, respectively. Qwen3.6-27B has a longer intermediate regime. Both Qwen models show local fluctuations in concentration, so their boundaries follow sustained trends in CKA, persistence, dimensionality, and final prediction alignment. Llama-3.3-70B reaches sustained intermediate geometry and stable persistence around layer 34, which motivates its later lower boundary. The resulting bands are model-specific, fixed once, and reused for every method in the main comparison.

The dashed square in each CKA heatmap and the gold shading in each diagnostic panel mark the selected interval from Table 9. The figures show the complete depth profile around each boundary, making the relationship between the selected band and its supporting evidence visible. The lower boundary follows the initial transition in geometry and readout behavior. The upper boundary precedes the sustained move toward direct output prediction. Reading the panels together is particularly useful for models with several local concentration peaks, where the broader depth profile provides a clearer guide than any individual peak.

Figures 6, 7, 8, 9, and 10 show the corresponding diagnostics for all five models.

B.2 Layer-wise Concept Readout

The layer-wise diagnostic connects the structural analysis to concept readout. Figure 5 reports the pointwise Top-10 rate at every recorded source layer. Single-token readout asks whether the annotated concept token appears in the J-lens top 10. First-token readout asks the same question for the first token of a multi-token concept. For single-token concepts, we use 103 single-token clozes from the official J-lens multi-hop cloze benchmark and evaluate the target token itself. The comparison shows where J-lens supplies a useful clue as the model processes a concept that spans several tokens.

At each recorded layer, we recompute the Top-10 indicator for every directed cloze and then average those indicators over the evaluation set. A point on the blue curve therefore measures the accessibility of the first token at that particular depth; it does not carry successes forward from neighboring layers. The orange curve applies the identical aggregation to the single-token control, so differences between the curves reflect the granularity of the target rather than a change in the metric. Because the curves are pointwise, local rises and falls can occur when information is briefly exposed, reorganized, or replaced by a later output-oriented representation.

We retain the complete saved sweep, including layers outside the selected workspace band, and leave unavailable early-layer measurements unobserved rather than imputing zeros. The task evaluation still uses the pre-specified workspace band rather than trimming it from the plotted curve. The selected interval is fixed from the structural diagnostics and is not chosen to maximize the first-token curve. Finally, a high first-token rate is only a clue, not a complete-concept readout. The end-to-end evaluation still requires **PROPOSE** to complete the candidate and **SCORE** to retrieve it under joint vocabulary competition.

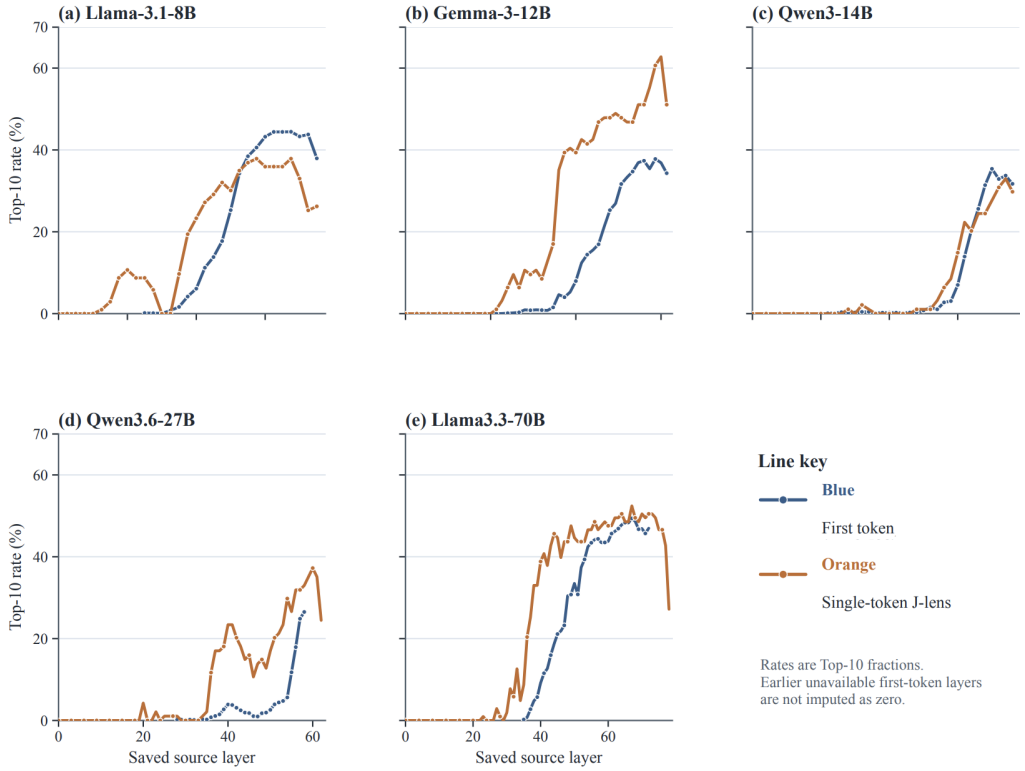

Figure 5: **Layer-wise concept readout.** Pointwise Top-10 rates for first tokens in the paired-cloze benchmark (blue, $n = 2{,}400$) and single-token clozes from the J-lens benchmark (orange, $n = 103$). The horizontal axis gives the source-layer index. Each curve covers the recorded layer range; missing earlier measurements are left unobserved.

B.3 Evaluation Protocol

The main end-to-end evaluation uses the 2,400 directed clozes and the model-specific band $\mathcal{W}_m$. At each source layer in that band, **PROPOSE** constructs a candidate set and **SCORE** ranks the recovered concepts jointly with vocabulary tokens. Let $r_{i,\ell}$ denote the complete target concept's rank for cloze i at layer ℓ. The aggregate metric uses the best rank attained within the selected interval,

$$r_i^\star = \min_{\ell \in \mathcal{W}_m} r_{i,\ell}, \qquad \text{Rank@}k = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[r_i^\star \le k] \,.$$

A target absent from a layer's candidate set receives rank $+\infty$ at that layer. Rank@1 and Rank@10 therefore require both successful proposal and successful scoring. Coverage uses the same aggregation rule and counts a cloze once the target is proposed within the band. The band is fixed before task evaluation, and the aggregation uses the full pre-specified interval. Every compared method uses the same workspace band for a given model, and fixed-vocabulary methods count uncovered targets as failures. The oracle-candidate setting includes the target explicitly to measure scoring with target availability ensured.

The layerwise minimum is an accessibility measure over the selected workspace, rather than an assumption that one hidden state carries the concept unchanged throughout the band. A target that is

never proposed receives infinite rank throughout the evaluation, which keeps proposal failure distinct from a low score among available candidates.

The same rule applies to every comparison. In end-to-end evaluation, each layer has its own candidate set and a target must be proposed and ranked there. In the oracle-candidate setting, the target is inserted before scoring, so rank is conditional on its availability.

These band-wide metrics are distinct from the fixed-layer diagnostic controls. Appendices D and E use separate representative source anchors for their respective comparisons, with a separate carrier anchor when cross-layer transport is being tested. The anchors are selected from preliminary layer-wise readout diagnostics and then held fixed within the corresponding comparison. The shared-state comparison in Appendix J instead uses the layer exposed by the public Oracle Lens interface.

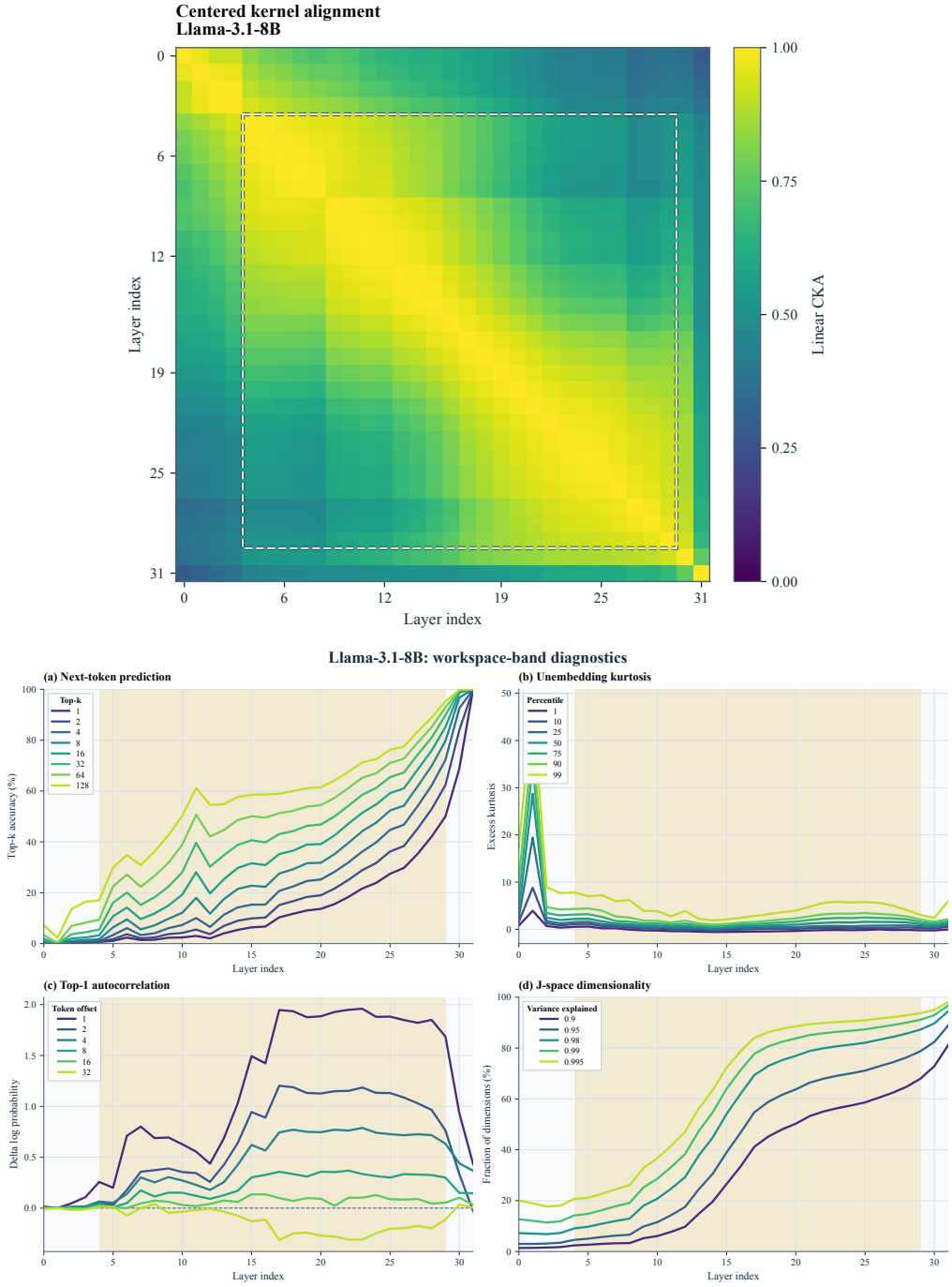

Figure 6: **Workspace diagnostics for Llama-3.1-8B.** The upper panel shows vocabulary-centered linear CKA between layerwise J-lens direction matrices. The lower panels show next-token prediction accuracy, vocabulary-logit kurtosis, top-1 persistence above the shuffled-position null, and effective J-space dimensionality. The dashed square and gold shading mark $\mathcal{W}_m = 4$–29. The early kurtosis spike precedes sustained persistence. The selected band follows this initial lexical regime and ends before the final prediction surge.

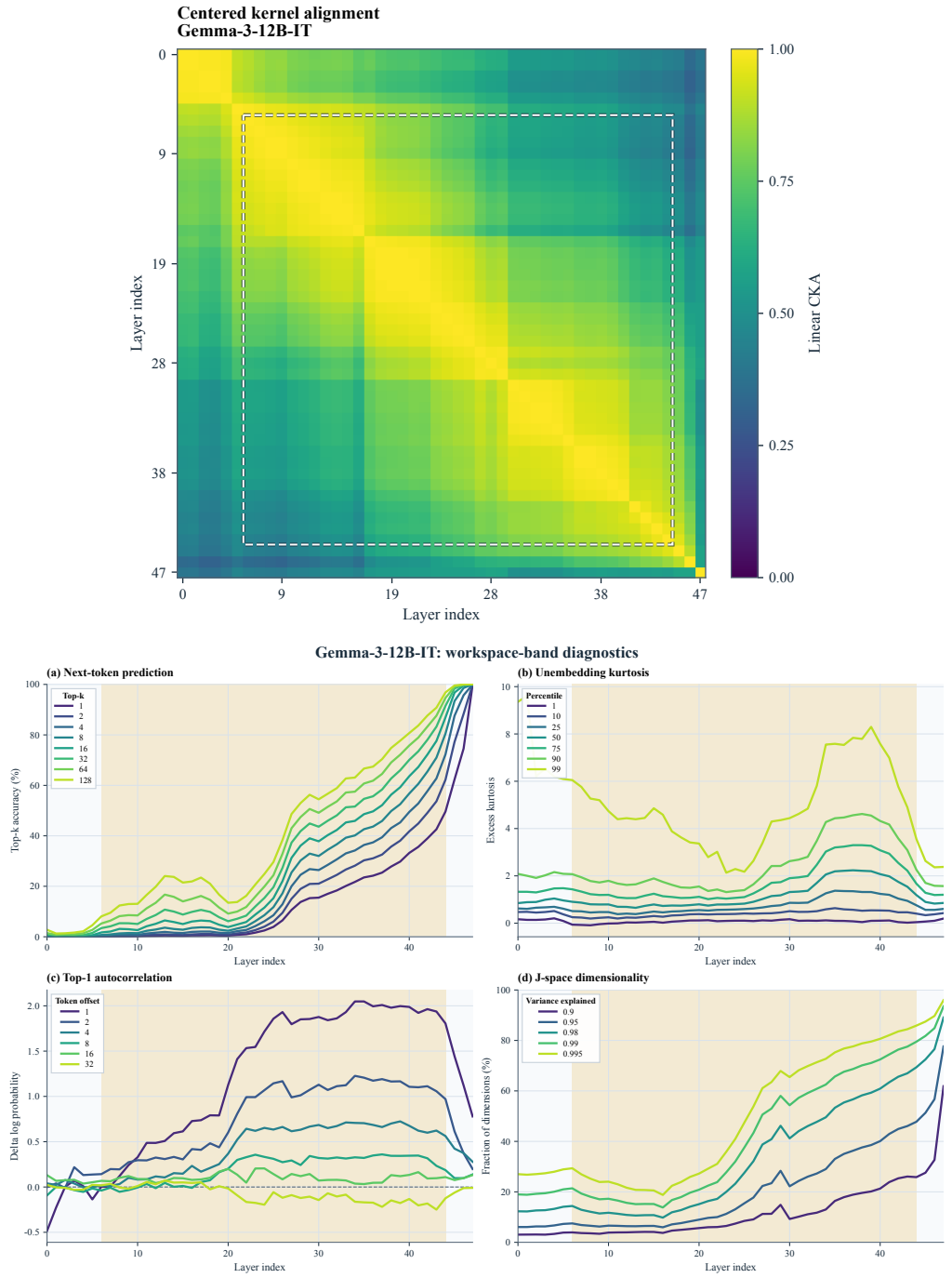

Figure 7: **Workspace diagnostics for Gemma-3-12B.** Panels follow Figure 6, with $\mathcal{W}_m = 6$–44 marked by the dashed square and gold shading. Elevated readout concentration in the initial layers precedes the sustained geometry and persistence of the middle regime. The band begins after that initial transition and ends before the late shift toward next-token decoding.

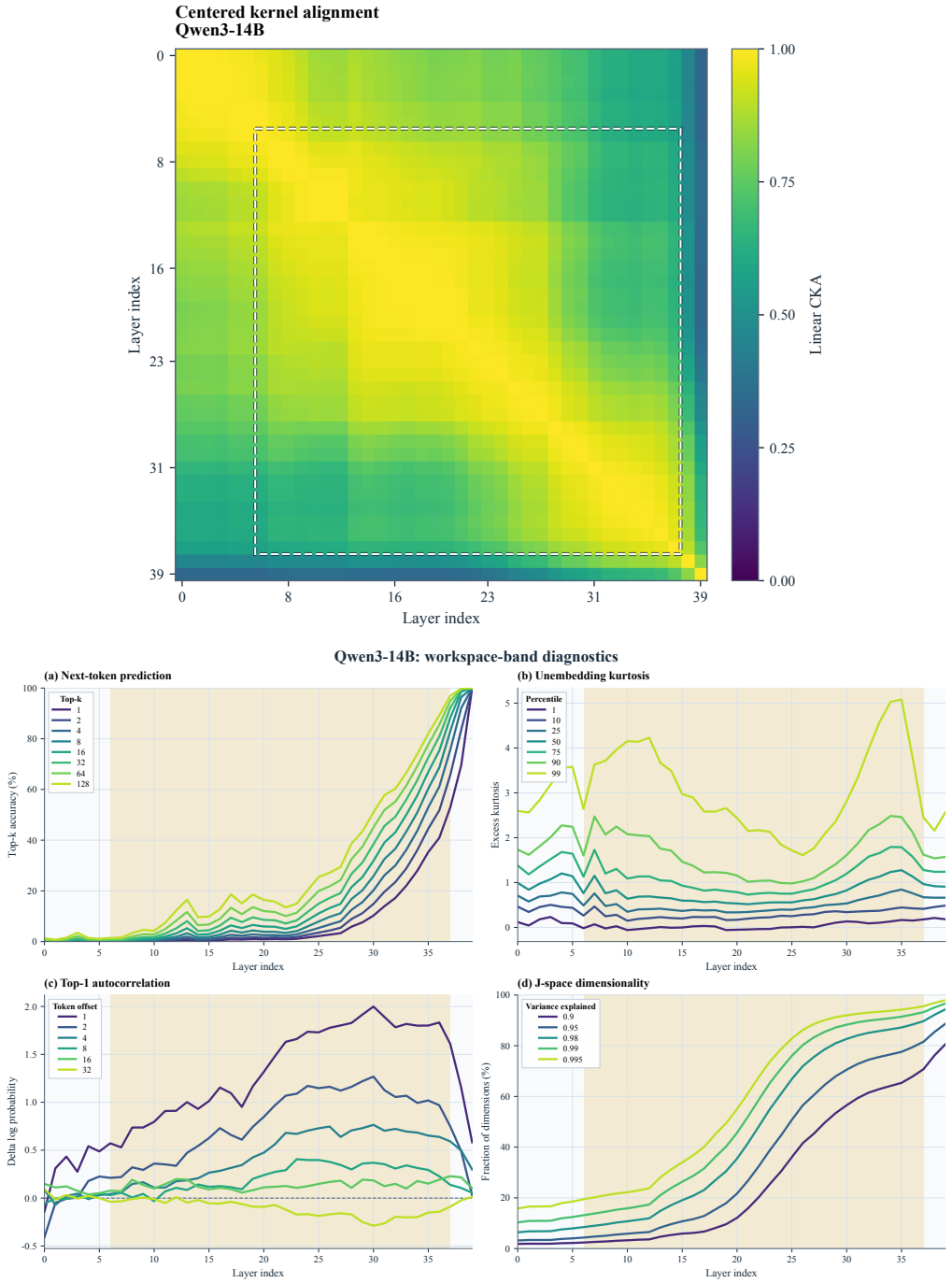

Figure 8: **Workspace diagnostics for Qwen3-14B.** The dashed square and gold shading mark $\mathcal{W}_m = 6$–37. CKA identifies a coherent middle geometry, accompanied by sustained persistence and changes in effective dimensionality. The concentration curve has several local peaks. The boundaries follow the joint depth profile, with the upper boundary preceding the final rise in next-token prediction accuracy.

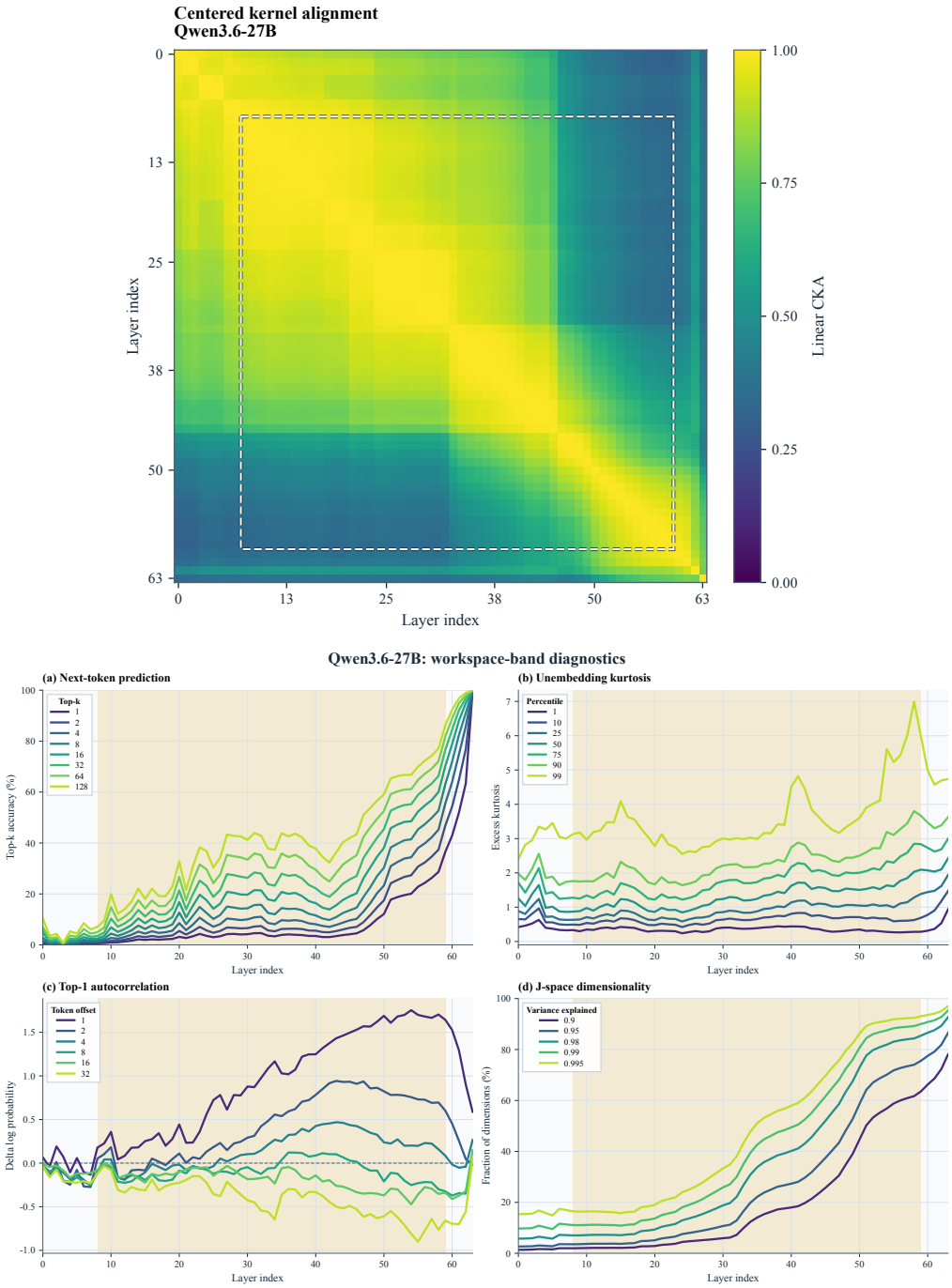

Figure 9: **Workspace diagnostics for Qwen3.6-27B.** The dashed square and gold shading mark $\mathcal{W}_m = 8$–59. A long intermediate regime contains several local changes in readout concentration. CKA, persistence, and dimensionality together support a sustained interval across those changes. The layers after the band show the final transition toward direct next-token decoding.

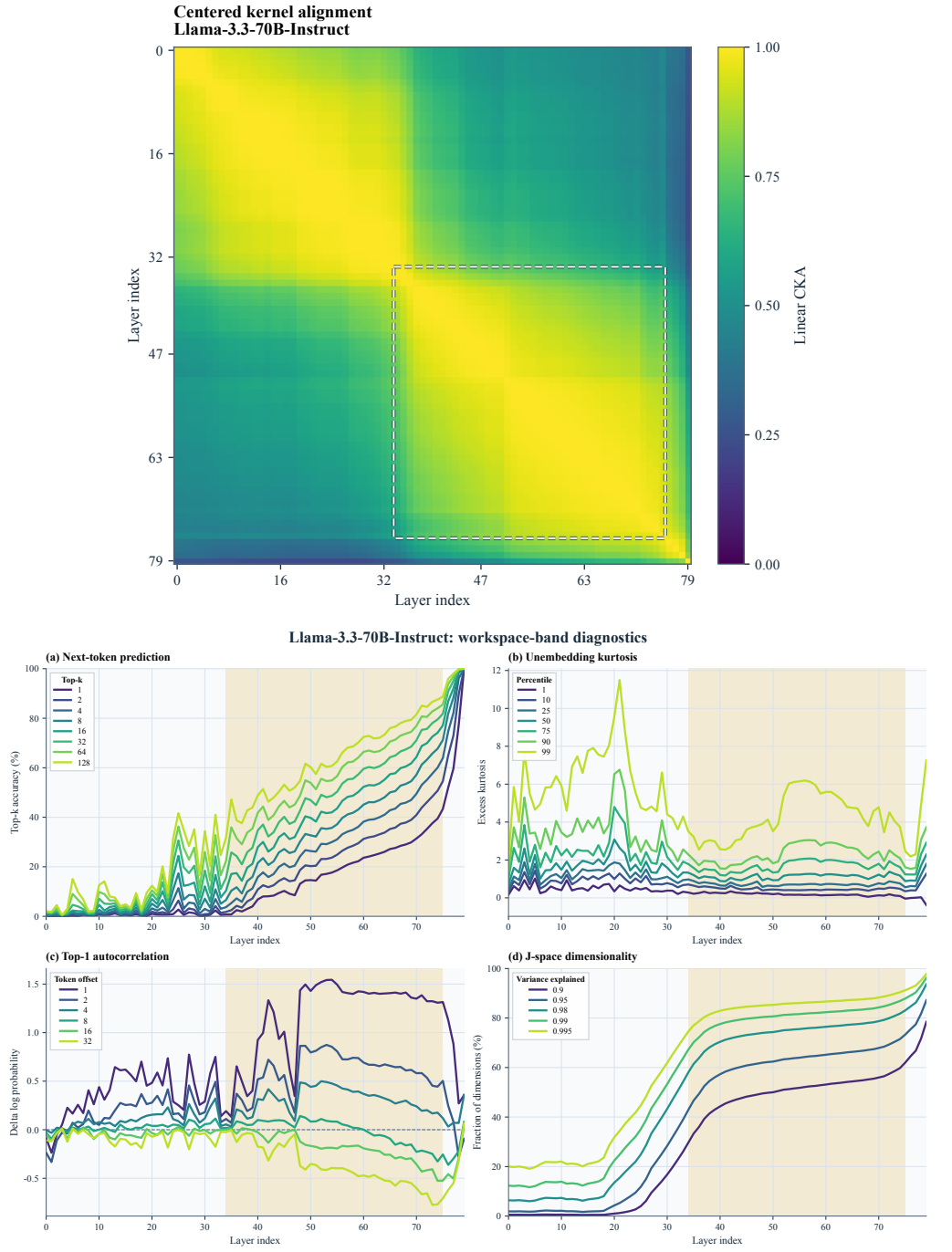

Figure 10: **Workspace diagnostics for Llama-3.3-70B.** The dashed square and gold shading mark $\mathcal{W}_m = 34$–75. The sustained intermediate J-lens regime begins later in depth than in the smaller models. Its onset around layer 34 coincides with a structural transition and stabilization of persistent readout. The upper boundary precedes the final decoding transition.

C TEMPLATE LENS CONSTRUCTION AND VOCABULARY COVERAGE

Fixed-vocabulary readout requires both target coverage and accurate scoring. A concept must belong to the stored inventory before its representation can be compared with the source state. Template Lens therefore separates two properties of a fixed-vocabulary reader: the concepts made available by its inventory and the quality of the representations used to score them.

C.1 TEMPLATE LENS CONSTRUCTION

Template Lens extends J-lens with precomputed representations for a fixed concept vocabulary (Gurnee et al., 2026). We construct templates from a curated inventory of $87{,}157$ lexical concepts. The curation removes adjectival and adverbial expressions, verb phrases, Latin scientific names, and generic or metalinguistic surfaces that do not serve as standalone concept labels. We rank the retained concepts by a frozen frequency proxy and assign consecutive ranks in this final order. The J-lens Template Lens construction uses a $12{,}800$-concept bank, which covers only 10.8% of our directed clozes (Figure 11). We therefore use the first $38{,}400$ concepts, giving the fixed-vocabulary baseline three times as many stored candidates. This larger bank provides a substantially broader candidate set and makes the comparison favorable to Template Lens. Each stored direction summarizes activations from contexts that naturally precede its concept, then remains fixed across source prompts.

For each vocabulary entry w, we generate $P = 200$ ordinary English prefixes whose natural continuation is w. Validation excludes the target concept, its content words and surface variants, naming-style prompts, and completed sentences. The subject model processes each prefix alone. Let $h_{\ell,j}^{(w)}$ be the final-token residual at layer ℓ for accepted prefix j of concept w. Let $\mu_\ell(w)$ and μ_ℓ denote the per-concept and pooled means, and let Σ_ℓ be the pooled covariance. The stored Template Lens direction is

$$t_\ell(w) = \left(\Sigma_\ell + \lambda_\ell^{\mathrm{TL}} I\right)^{-1} \left(\mu_\ell(w) - \mu_\ell\right), \qquad \lambda_\ell^{\mathrm{TL}} = 10^{-3}\ \mathrm{mean}\big(\mathrm{diag}(\Sigma_\ell)\big). \tag{13}$$

The centered mean captures the activation shift associated with a concept, and the regularized covariance adjusts for variation shared across the prefix collection. Template construction does not use the source activations of the evaluation clozes. At evaluation, each stored Template Lens direction and every vocabulary row use the normalized joint score in Equation 8. Template Lens also uses the same workspace bands, target-rank definition, and directed-cloze denominator as in Appendix B. The common protocol fixes the source state and ranking rule, so the comparison reflects how the methods obtain their candidate concepts and construct the representations used to score them.

The curated inventory is drawn from Open English WordNet 2025+. It retains alphabetic surfaces with one to six written words and two to six continuation tokens under the instruction-tuned model tokenizers. Benchmark coverage uses normalized exact surface matching, without aliases or manual additions. The same token-eligibility rule is applied when fitting representations for every model.

C.2 VOCABULARY COVERAGE

Let $C_i^\star$ be the target intermediate concept for directed cloze i, and let $\mathrm{norm}(\cdot)$ apply Unicode normalization, case normalization, and whitespace collapse. For a fixed vocabulary $\mathcal{V}$, its benchmark coverage is

$$\mathrm{Cov}(\mathcal{V}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\mathrm{norm}(C_i^\star) \in \mathcal{V}], \qquad N = 2{,}400. \tag{14}$$

End-to-end Rank@k for model m is

$$\mathrm{Rank@}\,k_m(\mathcal{V}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[r_{i,m}^{\mathrm{TL}}(\mathcal{V}) \le k\right], \tag{15}$$

where a target absent from $\mathcal{V}$ has $r_{i,m}^{\mathrm{TL}}(\mathcal{V}) = +\infty$.

To separate target availability from scoring quality, let $A(\mathcal{V}) = \{i : \mathrm{norm}(C_i^\star) \in \mathcal{V}\}$ be the covered subset. Conditional Rank@k is

$$\text{Conditional Rank@}\, k_m(\mathcal{V}) = \frac{1}{|A(\mathcal{V})|} \sum_{i \in A(\mathcal{V})} \mathbf{1}\left[r_{i,m}^{\mathrm{TL}}(\mathcal{V}) \le k\right]. \tag{16}$$

The end-to-end metric therefore decomposes exactly as

$$\text{Rank@}\, k_m(\mathcal{V}) = \mathrm{Cov}(\mathcal{V}) \times \text{Conditional Rank@}\, k_m(\mathcal{V}). \tag{17}$$

A fixed inventory places a ceiling on end-to-end readout through its coverage. Expanding the vocabulary can raise that ceiling, but available targets must still score above their competitors. Conditional Rank@k captures this second requirement on the covered subset. The factorization therefore distinguishes the availability supplied by the inventory from the quality of the scoring representation.

For the $38{,}400$-entry inventory used in the main comparison, coverage is 25.7% for every model. The corresponding Conditional Rank@10 values implied by Table 2, in the same model order, are approximately 77.4%, 86.4%, 81.7%, 85.2%, and 91.4%. Multiplying each value by the 25.7% coverage reproduces the end-to-end Template Lens Rank@10 values of 19.9%, 22.2%, 21.0%, 21.9%, and 23.5% reported in Table 2.

We characterize the availability supplied by the final inventory using two vocabulary schedules at each size s. Let $\mathcal{U}$ denote the $87{,}157$-concept curated inventory. The random schedule samples $\mathcal{V}_s^{\mathrm{rand}}$ uniformly without replacement from $\mathcal{U}$. Its expected coverage is linear in s,

$$\mathbb{E}\left[\mathrm{Cov}\left(\mathcal{V}_s^{\mathrm{rand}}\right)\right] = \frac{s}{|\mathcal{U}|}\,\mathrm{Cov}(\mathcal{U}), \tag{18}$$

The frequency-ranked schedule $\mathcal{V}_s^{\mathrm{freq}}$ contains the first s concepts in the final frequency order. The comparison tests whether frequency-based selection improves coverage over uniform sampling at the same vocabulary budget.

Figure 11 compares the random expectation with the frequency-ranked schedule. The two curves isolate the inventory component of fixed-vocabulary readout. Frequency-ranked coverage rises rapidly at modest budgets, while the full curated inventory supplies the endpoint under exact surface matching. Each additional concept also requires prefix generation, model forward passes, and representation storage. The vertical guide marks the $38{,}400$-concept bank used for template construction. Even with this threefold expansion over the $12{,}800$-concept J-lens configuration, our method retains the more favorable coverage–preparation trade-off (Appendix H). Equation 17 separates this inventory effect from the quality of the readout representations. The shared-candidate comparison below holds candidate availability fixed and examines the latter component.

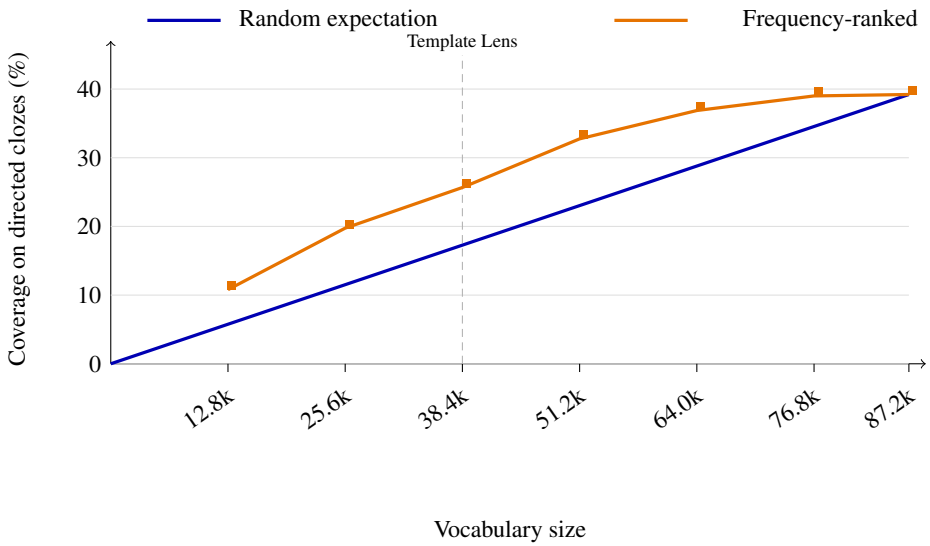

Figure 11: **Static vocabulary coverage as the curated vocabulary grows.** The curves use normalized exact surface matching over all 2,400 directed clozes. "Random expectation" is the analytical expectation for uniform sampling without replacement from the 87,157-concept curated inventory. "Frequency-ranked" uses prefixes of the final frequency order. The leftmost point is the 12,800-concept J-lens configuration; the vertical guide marks the 38,400-concept Template Lens bank.

C.3 Oracle-Candidate Setting

The oracle-candidate setting guarantees that the target is available to both methods. For each cloze i, we construct the target-augmented candidate set

$$\mathcal{C}_i^{\text{oracle}} = \mathcal{C}_i \cup \{C_i^\star\}, \tag{19}$$

where $\mathcal{C}_i$ is the candidate set proposed at each evaluated layer for our method and the stored concept bank for Template Lens. Both methods use the same source activation, workspace band, and rank metric. Coverage is 100% by construction, and all $2{,}400$ directed clozes enter the denominator for every model.

The oracle-candidate setting removes candidate availability as a source of variation. Both methods must rank the target above their remaining candidates using the same source state. Differences in rank therefore reflect representation construction and scoring. End-to-end evaluation adds the requirement that each method supply the target through its ordinary candidate mechanism. The two evaluations answer complementary questions about the complete pipeline and its scoring component. Appendix J separately compares our readout with the learned Oracle Lens method.

Across the five models, Rank@10 in the oracle-candidate setting ranges from 80.2–98.8% for J-lens, 81.8–95.9% for R-lens, and 67.5–80.8% for Template Lens (Table 2). These values quantify the scoring component after candidate availability is held fixed.

D Further Evaluation of Concept Readout

We extend the evaluation of Concept Lens by examining candidate coverage and concept selection from hidden states. **PROPOSE** supplies candidate concepts, and **SCORE** evaluates their agreement with the hidden state being inspected. The experiments compare proposal strategies, evaluate scoring on fixed candidate banks, and test whether concept selection follows changes in the source state. A control using concepts with the same first token examines the distinctions captured by the recovered representations.

These experiments complement the reconstruction task in Table 1 and the causal concept swaps in Section 5.2. Reconstruction examines concept identity, the controls below examine concept selection, and causal swaps test the behavioral effects associated with the readout. Appendix E provides the analysis of representation recovery, transport, and ranking stability.

For a source cloze x at layer ℓ, let $\mathcal{C}(x)$ be the proposed candidate set, v_C the recovered direction for candidate C, and $G_C(h_\ell(x))$ its score against the source hidden state. The controls below vary one of these ingredients at a time while keeping the remaining computation fixed.

D.1 Candidate Generation Strategies

The first experiment isolates the information supplied to **PROPOSE**. Let $C^\star$ be the annotated concept for x, and let $s(x)$ denote the first-token clue selected from the R-lens ranking. Throughout this ablation, any readout seed is taken from R-lens. Table 10 defines the two proposal conditions, and Table 11 compares them quantitatively.

Table 10: **Proposal conditions for the first-token ablation.** All conditions use the same source cloze, frozen subject model, decoding rule, search budget, maximum concept length, candidate cap, and downstream **SCORE**. Only the initial information supplied to **PROPOSE** changes.

Condition	Initial information	Diagnostic interpretation
Lens-seeded	The R-lens-ranked first-token clue $s(x)$	The complete method
Direct proposal	No forced first token; the frozen model proposes directly from x	What the source prompt supplies without a lens clue

The lens-seeded condition is the method used in the main experiments. The direct-proposal condition removes the forced lexical prefix while retaining the source cloze. Their comparison isolates the contribution of a source-aligned first-token clue to candidate discovery.

We hold the source cloze, frozen subject model, decoding rule, expansion budget, maximum concept length, and candidate-set size fixed across conditions. The recovered candidate directions and the downstream **SCORE** are also evaluated with the same settings. The primary metric is proposal *Coverage*,

$$\text{Coverage} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[C_i^\star \in \mathcal{C}_i]\,, \tag{20}$$

which asks whether the target enters the candidate space at all. End-to-end Rank@10 is a secondary metric that measures how changes in candidate availability propagate through the complete readout. The lens-seeded versus direct-proposal comparison tests the contribution of a source-aligned clue.

Table 11: **Ablating the first-token clue during candidate proposal with R-lens.** Coverage measures whether the target enters the candidate set. End-to-end Rank@10 measures the resulting complete readout. Each model reports both metrics for the two proposal conditions.

	Llama-3.1-8B		**Gemma-3-12B**		**Qwen3-14B**		**Qwen3.6-27B**		**Llama-3.3-70B**	
Condition	Coverage	Rank@10	Coverage	Rank@10	Coverage	Rank@10	Coverage	Rank@10	Coverage	Rank@10
Lens-seeded	63.6	61.9	60.8	52.6	65.6	55.7	43.0	40.3	75.5	61.8
Direct proposal	43.0	41.3	43.7	40.6	51.3	46.2	61.7	52.8	70.6	55.3

Lens-seeded proposal has higher Coverage and Rank@10 than direct proposal on four of the five models. Qwen3.6-27B is the exception, where direct proposal performs better on both metrics. This result evaluates the lens-seeded proposer used in the main experiments; the downstream **SCORE** stage is unchanged across conditions. This ablation distinguishes candidate availability from evidence that a concept is represented in the source state. A proposal can be plausible under the source prompt and still fail the subsequent source-conditioned test. The distinction motivates the fixed-proposal-bank scoring comparison below.

The proposal results and the fixed-bank scoring results below use the same model-specific workspace band and workspace-band aggregation as the main evaluation. The subsequent source-mismatch and same-first-token controls instead use fixed source layers, which are reported explicitly to isolate source dependence and representation content.

D.2 Scoring a Fixed Candidate Set

We compare **SCORE** and three alternatives on the same candidate bank generated directly by the frozen model. Continuation likelihood measures each candidate's probability as a textual completion, while **SCORE** measures its agreement with the source hidden state. Holding the candidate bank fixed isolates the contribution of each scoring rule to concept selection. Targets enter the evaluation through ordinary candidate generation.

We retain only examples for which the target $C_i^\star$ is naturally present in its bank $\mathcal{C}_i$. For a candidate $C = (w_1, \ldots, w_n)$, the frozen-LM baseline uses the length-normalized continuation log likelihood

$$S_{\mathrm{LM}}(C \mid x) = \frac{1}{n^\alpha} \sum_{j=1}^{n} \log p_\theta(w_j \mid x, w_{<j}), \tag{21}$$

where x denotes the full direct-proposal context. We set $\alpha = 1$ (the per-token mean log likelihood) and retain the proposal implementation's decoding conventions.

This evaluation is deliberately stricter than ranking only among generated phrases. Each retained target competes against $\mathcal{V} \cup (\mathcal{C}_i \setminus \{C_i^\star\})$, where $\mathcal{V}$ is the complete token vocabulary. No target is injected. For a ranking rule r and $k \in \{1, 10\}$,

$$\mathrm{Rank@}\,k(r) = \frac{\sum_i \mathbf{1}[C_i^\star \in \mathcal{C}_i]\, \mathbf{1}[r_i(C_i^\star) \le k]}{\sum_i \mathbf{1}[C_i^\star \in \mathcal{C}_i]}. \tag{22}$$

For vector methods, every vocabulary token is scored by the same source-conditioned vocabulary score map: *Ours* uses the recovered carrier direction, *First token* uses $W_U[t_1]$, and *Remaining mean* uses the unit-normalized mean of $W_U[t_2], \ldots, W_U[t_n]$. The LLM row uses the exact teacher-forced per-token mean log likelihood for a multi-token candidate and the frozen next-token log likelihood for a vocabulary competitor. For each vector row, we compute ranks across the source layers in the model-specific workspace band $\mathcal{W}_m$ and use the minimum rank for each example, matching the workspace-band aggregation in the main evaluation; this operation is applied equally to *Ours* and both lexical controls. The LLM likelihood has no source-layer axis and therefore contributes one layer-independent score. Ties receive the conservative worst rank, which prevents candidates sharing a first token from receiving an arbitrary advantage. All reported scores are conditional on the model-specific cohort for which the target is naturally present in the frozen direct-proposal bank.

Table 12: **Full-vocabulary fixed-bank ranking with R-lens.** No gold target is injected: a target is retained only if it occurs naturally in the fixed direct-proposal bank, and is then ranked against the complete vocabulary and all other bank candidates. *Ours* uses the source-conditioned recovered-carrier scoring function; *LLM* uses exact teacher-forced length-normalized continuation likelihood; *First token* uses $W_U[t_1]$; and *Remaining mean* uses the mean direction of $W_U[t_2], \ldots, W_U[t_n]$. Vector rows use the same best-over-workspace-band protocol, whereas the LLM row has one layer-independent score; ties receive conservative worst rank.

Method	**Llama-3.1-8B**		**Gemma-3-12B**		**Qwen3-14B**		**Qwen3.6-27B**		**Llama-3.3-70B**	
	Rank@1	Rank@10	Rank@1	Rank@10	Rank@1	Rank@10	Rank@1	Rank@10	Rank@1	Rank@10
Ours	47.9	98.3	24.5	86.6	20.1	85.1	24.3	88.2	26.1	90.0
LLM	13.5	70.6	8.1	72.6	32.7	82.4	23.0	84.4	24.7	88.1
First token	11.8	63.5	14.0	56.7	12.4	54.5	6.1	39.5	16.2	47.3
Remaining mean	2.8	22.4	1.7	17.3	1.6	16.7	1.5	17.3	4.5	19.3

Table 12 shows that the recovered-carrier **SCORE** exceeds both lexical-direction controls on Rank@1 and Rank@10 in all five models. It also exceeds frozen continuation likelihood on both metrics for four of the five models. The result therefore evaluates selection against a large, common competition set within each model, rather than only ranking a small set of already-generated phrases.

D.3 Concept Selection Across Source States

The preceding comparison separates source-conditioned scoring from proposal preference. We next test whether the score changes when the supplied source activation changes, while every candidate and every recovered direction remains fixed. Both controls use counterfactual pairs (x_C, x_D) with the same relation and different intermediate concepts C and D. For these fixed-candidate controls, we use one representative source layer per model within the pre-specified workspace band. We selected the layer from preliminary layer-wise readout diagnostics to obtain a clear concept-readout signal, then fixed it before forming the paired comparisons; the same layer, candidate bank, and recovered directions are used for matched and swapped states. We form the union $\mathcal{U} = \mathcal{C}(x_C) \cup \mathcal{C}(x_D)$ from the ordinary candidate lists at the fixed carrier layer. A pair is retained when each list already contains its own target. The lists use the candidate-generation procedure in Appendix K.1 and contain at most 25 candidates per side, with no gold candidate added. Both sides then share the union, recovered representations, and scoring parameters.

Let $h_C = h_{\ell_S}(x_C)$ and $h_D = h_{\ell_S}(x_D)$ denote the paired source states. The matched condition supplies the query's own state, and the within-pair swap supplies its partner's state. Every candidate is rescored under the supplied state using the same frozen direction. For Qwen3-14B, carrier representations are transported from L37 to L32 before either condition is evaluated. The carrier/source layers are L27/L27 for Llama-3.1-8B, L43/L43 for Gemma-3-12B, L37/L32 for Qwen3-14B, L58/L58 for Qwen3.6-27B, and L60/L60 for Llama-3.3-70B. Carrier readers, layers, and candidate-generation settings are fixed before either control.

Hidden-state mismatch. The first control tests whether target retrieval requires the matched source state. For a directed query i with target $C_i^\star$ and shared union $\mathcal{U}_i$, the target margin is

$$M_i(h) = G_{C_i^\star}(h) - \max_{C \in \mathcal{U}_i \setminus \{C_i^\star\}} G_C(h).$$

A positive margin places the target first within the candidate union. Rank@k measures the fraction of queries whose target appears among the first k members of that union at the fixed source layer. Table 13 reports the diagnostic with vocabulary competitors excluded. Replacing the matched state sharply lowers Rank@10 in every model, providing a direct test that the selected concept depends on the source activation.

Counterfactual discrimination. The second control asks whether each source state favors its own concept over its partner under a fixed candidate bank. The directional margins are $m_C = G_C(h_C) - G_D(h_C)$ and $m_D = G_D(h_D) - G_C(h_D)$. Directional accuracy counts prompt directions whose own concept outscores its partner. Pair accuracy requires both comparisons to be correct,

$$\mathrm{PairAcc} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[m_{C,i} > 0 \ \wedge \ m_{D,i} > 0] \,.$$

We also rank each target in $\mathcal{V} \cup \mathcal{U}$, combining the complete vocabulary with the frozen concept candidates.

Table 13: **Target retrieval depends on the matched source state.** Both conditions use identical candidate unions and representations. Each model uses its strict paired cohort; Table 14 gives the cohort sizes. Rank@k (%) is measured within the candidate union, and Margin is the mean target margin M_i.

Model	Carrier/source	Source state	Rank@1	Rank@10	Margin
Llama-3.1-8B	L27 / L27	Matched	6.9	71.1	-11.78
		Within-pair swap	0.0	1.0	-50.07
Gemma-3-12B	L43 / L43	Matched	13.0	88.2	-7.28
		Within-pair swap	0.0	5.3	-47.98
Qwen3-14B	L37 / L32	Matched	13.3	65.3	-29.01
		Within-pair swap	0.4	10.9	-71.70
Qwen3.6-27B	L58 / L58	Matched	5.1	73.5	-16.54
		Within-pair swap	0.0	7.7	-47.10
Llama-3.3-70B	L60 / L60	Matched	5.9	61.0	-29.40
		Within-pair swap	0.0	4.2	-70.04

Table 14 reports directional accuracy, pair accuracy, and joint Rank@10 for matched and swapped states. Matched directional accuracy is 92.7%–99.2%, and both directions are correct in 85.5%–98.5% of pairs. Joint Rank@10 falls under state swaps in every model, including from 68.9% to 3.6% for Qwen3.6 and from 60.5% to 3.9% for Llama-3.3. The paired preference and vocabulary-level retrieval therefore support the same interpretation. With concept alternatives held fixed, the source state supplies the information that favors the concept associated with the current prompt.

Table 14: **Fixed candidates reveal source-specific selection under full-vocabulary competition.** Each model uses its strict paired cohort, whose two ordinary candidate lists already contain their own targets. Directional and Pair are closed C-versus-D accuracies; joint Rank@10 (%) ranks the target in $\mathcal{V} \cup \mathcal{U}$. No gold candidate is added.

Model	Layers	Cohort	State	Dir.	Pair	Rank@10
Llama-3.1-8B	L27 / L27	204 / 408	Matched	98.3	96.6	71.1
			Swap	1.7	0.0	0.7
Gemma-3-12B	L43 / L43	131 / 262	Matched	99.2	98.5	87.0
			Swap	0.8	0.0	3.8
Qwen3-14B	L37 / L32	124 / 248	Matched	92.7	85.5	64.9
			Swap	7.3	0.0	10.1
Qwen3.6-27B	L58 / L58	98 / 196	Matched	98.0	95.9	68.9
			Swap	2.0	0.0	3.6
Llama-3.3-70B	L60 / L60	204 / 408	Matched	94.9	89.7	60.5
			Swap	5.1	0.0	3.9

D.4 Distinguishing Concepts with the Same First Token

The fixed-candidate controls establish source dependence. We next test whether the recovered carrier direction can be reduced to the first token that initiated proposal. We select naturally occurring counterfactual pairs in which both concepts contain at least two tokenizer pieces and share the same first-token ID under the leading-space tokenization convention. A first-token direction therefore gives the same prototype to both candidates in every selected pair. Any separation in this control must come from information preserved after the first token.

We use one fixed source layer per model, selected within the workspace from preliminary layer-wise readout diagnostics where the carrier reader gives a clear signal: L24, L36, and L30 for Llama-3.1-8B, Gemma-3-12B, and Qwen3-14B, respectively, and L58 and L60 for Qwen3.6-27B and Llama-3.3-70B. The layer is fixed for all 37 pairs and all baselines. This conditional analysis asks whether those baselines reproduce distinctions already made by the carrier reader, while the full-pair rows remain the carrier reader's primary estimate.

For a pair (C_1, C_2), the source prompt for C_1 must strictly prefer C_1 over C_2, and the source prompt for C_2 must strictly prefer C_2 over C_1. A pair succeeds only when both comparisons are correct. We compare the frozen carrier reader with the first-token direction, the mean of all constituent-token directions, and a fixed permutation of carrier-reader prototypes. Every method uses the same direct-

effect score map, source states, and pair set. The comparison therefore isolates whether the carrier suffix contributes information beyond the shared lexical prefix.

The full-pair results are reported in Table 15.

Table 15: **The frozen carrier reader distinguishes concepts that share their first tokenizer token.** The table reports the full 37-pair set for the five instruction-tuned models. Entries count pairs for which both directed source prompts strictly prefer their own concept. The first-token control ties within every selected pair by construction. The constituent mean receives all token pieces but does not use the carrier suffix.

Model / layer	Frozen carrier reader	Constituent mean	Permuted carrier reader	First token
Llama-3.1-8B / L24	**22/37**	18/37	5/37	0/37
Gemma-3-12B / L36	**26/37**	7/37	6/37	0/37
Qwen3-14B / L30	**19/37**	7/37	4/37	0/37

The two larger models each yield 37 same-first-token lexical pairs. The frozen carrier reader strictly separates 26 pairs for Qwen3.6-27B and 25 pairs for Llama-3.3-70B. Table 16 evaluates each lexical baseline on the subset classified correctly by that model's carrier reader. This conditional comparison asks whether a lexical baseline can reproduce distinctions already made by the frozen carrier reader. The full-pair table above remains the estimate of carrier-reader performance on the selected lexical control.

Table 16: **Lexical baselines on same-first-token pairs classified correctly by the carrier reader in two large models.** Each model has 37 lexical pairs. The subset contains pairs whose two source directions are strictly correct under the frozen carrier reader. Entries count pairs for which both directions remain correct under the displayed baseline.

Model / layer	Carrier-reader subset	Constituent mean	Permuted carrier reader	First token
Qwen3.6-27B / L58	26/37	14/26	2/26	0/26
Llama-3.3-70B / L60	25/37	17/25	1/25	0/25

The same-first-token pairs show that the frozen carrier reader retains information that distinguishes complete concepts despite an identical initial token. The constituent mean receives every token piece, making its comparison with the carrier reader informative about the contribution of the carrier suffix. The permuted-carrier-reader condition further tests whether success depends on assigning each recovered direction to the correct concept. These controls complement the state-swap experiment by holding the shared lexical clue fixed.

D.5 Implications for Concept Readout

The controlled comparisons show how Concept Lens selects complete concepts from hidden states. With candidates and representations held fixed, the supplied source state favors the corresponding concept. Carrier representations also distinguish concepts that share their first token. On fixed banks containing naturally proposed targets, **SCORE** achieves higher Rank@10 than continuation likelihood and lexical scoring across all five models. Together, these results support the use of the inspected hidden state to select among candidate concepts.

The proposal comparison identifies candidate generation as a separate avenue for improving coverage while retaining the same representation recovery and scoring procedures. The causal swaps in Section 5.2 and Appendix F.1 complement these readout results by testing whether the recovered representations redirect model behavior as predicted by the concept labels.

E Carrier Recovery, Effect Transport, and Ranking Stability

The analysis follows a candidate representation from carrier recovery to its final rank. We first identify the signal-to-residual condition under which pooling recovers a stable direction. We then show how ridge transport filters that direction in the source lens's effect space. Finally, we connect the transport perturbation to a query-sensitive score interval and a Top-k guarantee. Each step addresses

a different part of the readout, allowing recovery quality and transport stability to be examined separately. The ranking guarantee is defined relative to the direct carrier effect, which provides the reference score before transport.

E.1 Carrier Recovery

Let $\tau(C)$ be the fixed carrier prompt for concept C. For every nonzero vector z, write $\mathrm{unit}(z) = z/\|z\|_2$. At carrier layer ℓ_c, let $h_{\ell_c,t}(\tau(C)) \in \mathbb{R}^d$ be the hidden state at suffix position $t \in \{1, \dots, T\}$. We use the local model

$$h_{\ell_c,t}(\tau(C)) = \mu_t + a_t g_C + \epsilon_t(C), \tag{23}$$

where $\mu_t \in \mathbb{R}^d$ is the position-specific background, $g_C \in \mathbb{R}^d$ is a shared concept-related direction, $a_t \geq 0$ is its strength at position t, and $\epsilon_t(C) \in \mathbb{R}^d$ contains the remaining variation. The local model separates activity associated with the fixed suffix from a component shared across its positions. Under this assumption, centering removes the estimated background and pooling combines evidence for the shared direction. The analysis below states the condition under which the pooled signal is strong enough to determine a stable recovered direction.

Let $\widehat{\mu}_t$ be the background estimated on the calibration set. The carrier reader uses fixed nonnegative weights α_t with $\sum_t \alpha_t = 1$ and forms

$$\Phi(C) = \sum_{t=1}^{T} \alpha_t \left(h_{\ell_c,t}(\tau(C)) - \widehat{\mu}_t\right) = \kappa g_C + \eta_C, \tag{24}$$

where $\kappa = \sum_t \alpha_t a_t$ is the pooled signal strength and $\eta_C = \sum_t \alpha_t(\epsilon_t(C) + \mu_t - \widehat{\mu}_t)$ is the pooled residual.

Proposition E.1 (Conditional carrier recovery)**.** *If $\kappa > 0$ and $\|\eta_C\|_2 < \kappa\|g_C\|_2$, then*

$$\|\mathrm{unit}(\Phi(C)) - \mathrm{unit}(g_C)\|_2 \leq \frac{2\|\eta_C\|_2}{\kappa\|g_C\|_2}. \tag{25}$$

Proof. Apply the standard unit-vector perturbation inequality to κg_C and $\kappa g_C + \eta_C$. The condition ensures that both vectors are nonzero. □

The bound relates directional accuracy to the strength of the shared signal relative to the pooled residual. Stronger agreement across suffix positions improves recovery when the remaining variation stays small. The bound depends on the residual after pooling and therefore accommodates correlated variation across positions. Table 1 tests this premise with held-out singleton concepts whose native directions provide an independent reference. The calibration split fixes the carrier reader before those concepts are evaluated, linking the conditional analysis to a direct test of representation recovery.

Recoverable information. Suppose a local coordinate z_C produces stacked suffix observations $y_C = Az_C$ and a desired effect $e_C^\star = Bz_C$, where A is an observation map and B is an effect map. A linear decoder can recover $e_C^\star$ from y_C for every z_C exactly when

$$\ker(A) \subseteq \ker(B). \tag{26}$$

The condition has a direct interpretation. Any change in the local coordinates that leaves every suffix observation unchanged must also leave the desired effect unchanged. Under that condition, the suffix contains enough information for a linear decoder to recover the effect. The fixed pooling rule in Equation 24 is one concrete carrier reader whose quality is evaluated empirically. Appendix D.4 complements the reconstruction test with pairs that share a first token, examining whether the carrier reader preserves distinctions carried by the complete concept.

E.2 Spectral Effect Transport

Let $J_{\ell_c}, J_{\ell_s} \in \mathbb{R}^{d\times d}$ be the J-lens Jacobians at the carrier and source layers. The carrier effect is $e_C = J_{\ell_c}\Phi(C)$. For $\ell_s \neq \ell_c$, **SCORE** computes a source-layer representation by ridge reconstruc-

tion,

$$v_{C,\ell_s} = \arg\min_{v\in\mathbb{R}^d} \left\{ \|J_{\ell_s} v - e_C\|_2^2 + \lambda_{\ell_s}^{\text{ridge}} \|v\|_2^2 \right\}, \qquad \lambda_{\ell_s}^{\text{ridge}} > 0, \tag{27}$$

and sets $\widehat{e}_{C,\ell_s} = J_{\ell_s} v_{C,\ell_s}$. The coefficient used in the implementation is specified in Appendix K.2. Here I_d denotes the d-dimensional identity. Same-layer transport uses $\widehat{e}_{C,\ell_c} = e_C$.

Proposition E.2 (Spectral effect transport)**.** *Let $J_{\ell_s} = U \operatorname{diag}(\sigma_1, \ldots, \sigma_d) V^\top$ be a singular value decomposition with singular values $\sigma_i \geq 0$. Then*

$$\widehat{e}_{C,\ell_s} = P_{\ell_s} e_C, \qquad P_{\ell_s} = U \operatorname{diag}\left(\frac{\sigma_i^2}{\sigma_i^2 + \lambda_{\ell_s}^{\text{ridge}}} \right)_{i=1}^{d} U^\top. \tag{28}$$

Thus, the i-th left-singular effect direction is retained by the factor $\sigma_i^2/(\sigma_i^2 + \lambda_{\ell_s}^{\text{ridge}})$.

Proof. The ridge solution is $(J_{\ell_s}^\top J_{\ell_s} + \lambda_{\ell_s}^{\text{ridge}} I_d)^{-1} J_{\ell_s}^\top e_C$. Substituting the singular value decomposition and multiplying by J_{ℓ_s} gives Equation 28. □

The filter separates carrier-reader error from ridge bias. Let $e_C^\star$ be an independently specified reference effect and write $e_C = e_C^\star + J_{\ell_c} \delta_C$. Then

$$\widehat{e}_{C,\ell_s} - e_C^\star = \underbrace{P_{\ell_s} J_{\ell_c} \delta_C}_{\text{recovery error after transport}} + \underbrace{(P_{\ell_s} - I_d) e_C^\star}_{\text{ridge bias}}. \tag{29}$$

The first term propagates recovery error through the source-layer filter. The second term measures the part of the reference effect removed by regularization. The decomposition connects the final discrepancy to two choices in the method, the quality of the carrier reader and the ridge coefficient. Directions with large singular values are retained more strongly, while directions with small singular values are damped. Transport therefore favors effects that the source lens can represent without requiring a large residual-stream representation. All quantities in the comparison are defined relative to the specified reference $e_C^\star$.

The FP64 implementation reproduces the spectral residual identity to numerical precision. For this numerical check, we use one fixed representative source layer per model within the workspace band, selected from preliminary layer-wise diagnostics where concept readout is readily measurable. The resulting layers are L24, L36, L30, L53, and L53 for Llama-3.1-8B, Gemma-3-12B, Qwen3-14B, Qwen3.6-27B, and Llama-3.3-70B, respectively; the layers are fixed before computing the interval statistics. At these source layers, the median residual-energy fractions are 1.52×10^{-4}, 2.89×10^{-3}, 4.07×10^{-4}, 1.97×10^{-4}, and 1.71×10^{-4}, respectively. The maximum FP64 SVD-identity discrepancies are 1.12×10^{-11} for Qwen3.6 and 6.42×10^{-12} for Llama-3.3. These are the fixed cross-layer points used below.

E.3 Query-Sensitive Top-k Stability

The final score depends only on normalized direction. Let $N(e) = \Gamma e / r_N(e)$ be the model's final normalization, where Γ is a fixed diagonal gain and $r_N(e) > 0$ is the RMS scale. Define

$$F(e) = \operatorname{unit}(N(e)) = \operatorname{unit}(\Gamma e), \qquad \operatorname{score}(e; q) = q^\top F(e), \tag{30}$$

where h_{ℓ_s} is the source hidden state and $q = N(J_{\ell_s} h_{\ell_s})$ is the source query. The positive shared unembedding scale in the implementation does not change a ranking and is omitted from this analysis. The remaining score is therefore a directional comparison between the candidate effect and the source query.

Fix a source query and a candidate i. Let e_i be its direct carrier effect, let $\widehat{e}_i$ be its transported effect, and define the measured implementation-coordinate perturbation

$$x_i = N(e_i), \qquad \widehat{x}_i = N(\widehat{e}_i), \qquad u_i = \widehat{x}_i - x_i. \tag{31}$$

Write $r_i = \|x_i\|_2$, $p_i = x_i / r_i$, and decompose

$$u_i = a_i p_i + b_i, \qquad b_i \perp p_i, \qquad r_i' = \|\widehat{x}_i\|_2 = \sqrt{(r_i + a_i)^2 + \|b_i\|_2^2}. \tag{32}$$

The reference score is $S_i^0 = q^\top p_i$, and the transported score is $\widehat{S}_i = q^\top \operatorname{unit}(\widehat{x}_i)$. The signed tangent term is

$$\ell_i = \frac{q^\top b_i}{r_i}. \tag{33}$$

Normalization makes the distinction between radial and tangent changes useful. The radial term $a_i p_i$ changes the vector's length and has no first-order effect on its normalized score. The tangent term b_i changes the direction, and its projection onto q determines the first-order score change. Retaining the sign of that projection distinguishes score increases from decreases. A bound based only on perturbation norm discards this information, motivating the query-sensitive interval below.

Lemma E.3 (Query-sensitive score interval)**.** *If $r_i + a_i > 0$, then the transported score satisfies*

$$\widehat{S}_i = S_i^0 + \ell_i + \varepsilon_i, \qquad |\varepsilon_i| \le R_i^{\mathrm{geom}}, \tag{34}$$

where

$$R_i^{\mathrm{geom}} = |q^\top p_i| \frac{\|b_i\|_2^2}{r_i'(r_i' + r_i + a_i)} + |q^\top b_i| \frac{|r_i - r_i'|}{r_i r_i'}. \tag{35}$$

When $\|u_i\|_2 \le r_i/2$, the additional valid remainder is

$$R_i^{\mathrm{quad}} = 4\|q\|_2 \frac{\|u_i\|_2^2}{r_i^2}. \tag{36}$$

We use $R_i = \min\{R_i^{\mathrm{geom}}, R_i^{\mathrm{quad}}\}$ whenever both conditions hold, and otherwise use $R_i = R_i^{\mathrm{geom}}$. Therefore,

$$L_i = S_i^0 + \ell_i - R_i \le \widehat{S}_i \le S_i^0 + \ell_i + R_i = U_i. \tag{37}$$

Proof. The exact normalized direction is $\operatorname{unit}(\widehat{x}_i) = ((r_i + a_i)p_i + b_i)/r_i'$. Subtracting ℓ_i from the exact score change gives two terms. The identities

$$1 - \frac{r_i + a_i}{r_i'} = \frac{\|b_i\|_2^2}{r_i'(r_i' + r_i + a_i)}, \qquad \left|\frac{1}{r_i'} - \frac{1}{r_i}\right| = \frac{|r_i - r_i'|}{r_i r_i'}$$

bound them by Equation 35. The quadratic remainder follows from the standard unit-vector expansion under its stated small-perturbation condition. □

Theorem E.4 (Interval-count Top-k stability)**.** *Let g be the gold candidate in a finite candidate bank $\mathcal{B}$, and let $\operatorname{rank}_{\widehat{S}}(g)$ be one plus the number of candidates with a strictly larger transported score. If*

$$\#\{j \in \mathcal{B} \setminus \{g\} : U_j \ge L_g\} < k, \tag{38}$$

then $\operatorname{rank}_{\widehat{S}}(g) \le k$.

Proof. Every candidate with $U_j < L_g$ has $\widehat{S}_j < \widehat{S}_g$. Fewer than k remaining candidates can tie or exceed the gold lower bound. Thus fewer than k candidates can strictly exceed the transported gold score. □

The theorem converts score intervals into a ranking guarantee by counting candidates that could outrank the target. Candidates whose upper endpoints lie below the target's lower endpoint can be excluded immediately. The signed tangent term makes each interval sensitive to the source query and the direction of the score change. A perturbation orthogonal to the query contributes no first-order score change, even when its norm is appreciable. The certificate combines that geometric information with the separation between candidates, providing a decision-level test of transport stability.

E.4 Empirical Validation

We evaluate the score intervals on 1,200 counterfactual pairs, providing 2,400 directed source queries. Carrier templates, carrier readers, ridge coefficients, selected source layers, and the candidate bank are fixed before evaluation. Every selected-layer candidate satisfies $\|u_i\|_2 \le r_i/2$ and $r_i + a_i > 0$, so both remainder bounds are available. Table 17 compares intervals that use progressively more information about the perturbation. The common candidate bank and counting rule ensure that changes in certification coverage reflect the score bounds used in the same ranking test.

Table 17: **Finite score intervals certify transported Top-10 decisions.** Each value is the fraction of direct-reference Top-10 queries certified by the interval-count condition. All columns use the same candidate bank and candidate-count rule. "Prototype norm" bounds score change by post-normalization prototype distance. The two tangent columns use the quadratic remainder with symmetric or signed centers, and "Signed geometric" uses R_i^{geom} at the signed center. Every interval contains the observed transported score, and every certified query retains the observed transported gold concept in Top-10 within 10^{-8} numerical tolerance. Higher coverage is better.

Model / layer	Prototype norm	Symmetric tangent	Signed tangent	Signed geometric
Llama-3.1-8B / L24	88.6%	99.3%	99.3%	**99.7%**
Gemma-3-12B / L36	25.5%	75.8%	78.1%	**99.7%**
Qwen3-14B / L30	47.4%	90.7%	93.3%	**98.7%**
Qwen3.6-27B / L53	69.3%	96.1%	97.9%	**99.0%**
Llama-3.3-70B / L53	83.6%	98.1%	99.3%	**99.9%**

The geometric interval provides the highest certification coverage in all five models, and the signed center improves on the symmetric tangent interval in four. All evaluated score intervals have finite endpoints and contain the observed transported scores. Every certified Top-10 decision agrees with the transported reference within the stated numerical tolerance. These checks test both parts of the argument, containment of each score and preservation of the resulting ranking. The improvement in coverage shows the value of retaining the query-dependent direction and sign of the perturbation.

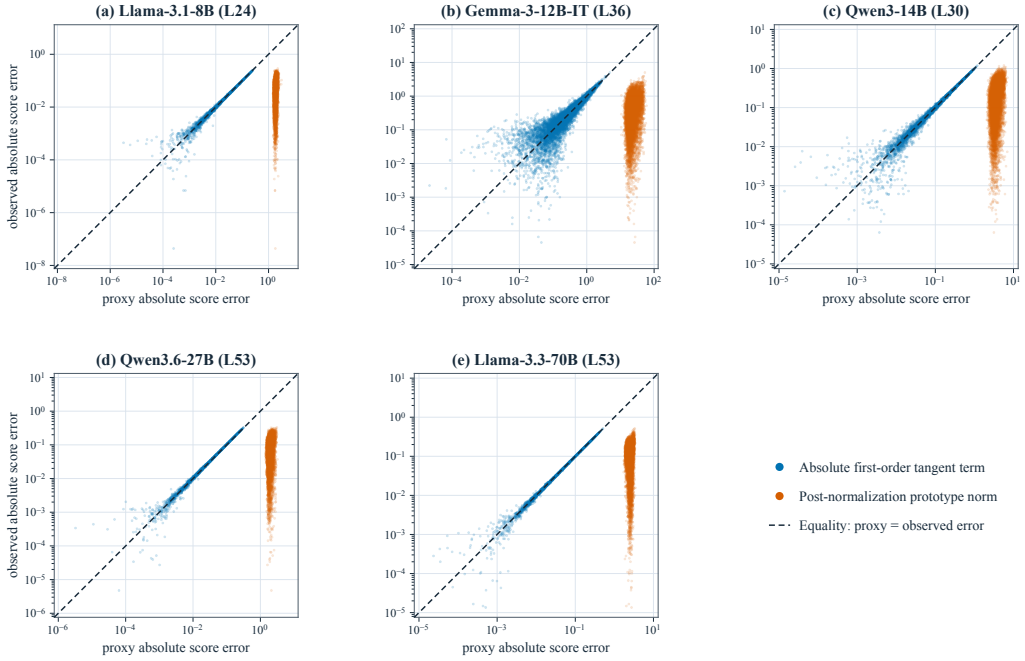

Figure 12: **Query-sensitive error separates observed score changes from a full prototype norm.** The five panels use fixed late source layers. The horizontal axis is a proxy for absolute score error, and the vertical axis is the observed absolute difference between direct and transported scores. Blue points use the absolute first-order tangent term from Equation 33. Orange points use the post-normalization prototype-norm bound. The displayed points are sampled from the fixed candidate bank.

Figure 12 connects the interval analysis to the observed score changes. The tangent term has absolute-error Spearman correlations of 0.9999, 0.9708, 0.9991, 0.9999, and 0.9999 for Llama-3.1, Gemma, Qwen3, Qwen3.6, and Llama-3.3, respectively. The corresponding prototype-norm correlations are 0.2690, 0.2281, 0.3075, 0.2385, and 0.1742. The tangent term closely tracks the score change across all five models because it measures the component observed by the query. The finite intervals in Table 17 turn this geometric agreement into a check of individual ranking decisions.

The certificate guarantees agreement with the direct-effect reference on the evaluated candidate bank. Its role is to establish that transport preserves a specified readout decision. Concept identity and source dependence are assessed through held-out reconstruction and the controls in Ap-

pendix D.4. The interval-count rule also applies to a joint concept–vocabulary bank when valid intervals are supplied for every competitor. The present analysis uses realized prototype pairs and fixed queries, giving a deterministic stability check that complements the semantic and causal evaluations elsewhere in the paper.

F Causal Intervention Protocol

This appendix specifies the protocol for the causal concept-swap experiment in Section 5.2. Each counterfactual pair contains two clozes with the same relation: a source cloze P_C whose answer is A_C and a partner cloze P_D whose answer is A_D. The associated intermediate concepts are C and D, respectively. We edit the residual stream while running P_C and test whether its continuation moves from A_C toward the answer A_D predicted by the partner. Both directions of each pair are evaluated, yielding $2{,}400$ directed clozes from $1{,}200$ pairs.

F.1 Intervention Setup

For each concept $X \in \{C, D\}$, **SCORE** recovers a carrier-layer direction $v_X^{(\ell_c)}$ from its concept text. The causal protocol uses a fixed multi-layer intervention set. At each intervention layer ℓ in this set, we transport and normalize this direction as

$$\hat{v}_X^{(\ell)} = \mathrm{unit}\Big(T_{\ell_c \to \ell} v_X^{(\ell_c)}\Big), \tag{39}$$

where $T_{\ell_c \to \ell}$ is the ridge transport defined in the main method. Thus, the directions are recovered before the causal trial and independently of the source residual state.

Let $h_{\ell,j}$ denote the residual stream at position j, and let j_C and j_D be the final positions of the source and partner prompts. We apply

$$\begin{aligned} \widetilde{h}_{\ell,j} = h_{\ell,j} &- d\,\mathbf{1}[j \geq j_C - 11]\left\langle h_{\ell,j}, \hat{v}_C^{(\ell)} \right\rangle \hat{v}_C^{(\ell)} \\ &+ a\,\mathbf{1}[j > j_C - n_m]\,\beta_D^{(\ell)} \hat{v}_D^{(\ell)}, \qquad \beta_D^{(\ell)} = \left|\left\langle h_{\ell,j_D}(P_D), \hat{v}_D^{(\ell)} \right\rangle\right|. \end{aligned} \tag{40}$$

The deletion term removes the source-concept component at the final 12 source-prompt positions. The addition term writes the partner direction at the final n_m positions, where n_m is the number of continuation tokens in the partner concept D under the subject model's tokenizer. Its magnitude $\beta_D^{(\ell)}$ is measured on the unedited partner prompt; hence no steering coefficient is fitted from the outcome of a causal trial.

The binary switches define the intervention arms: $(d, a) = (1, 1)$ is full replacement, $(0, 1)$ is add-only, and $(1, 0)$ is delete-only. The random-direction control also uses $(d, a) = (0, 1)$, but substitutes a seeded random unit direction for $\hat{v}_D^{(\ell)}$ while retaining the matched loading $\beta_D^{(\ell)}$. All other choices—the source and partner prompts, selected layers, write positions, and loading rule—are held fixed across arms.

F.2 Evaluation

For an edited source cloze i, let $r_i(A)$ be the rank of answer A in the edited continuation. We record whether the partner answer enters the top 10 and whether the source answer remains there:

$$\begin{aligned} (\mathrm{P@10}, \mathrm{S@10}, \mathrm{CS@10}) = \frac{100}{N} \sum_{i=1}^{N} \Big(&\mathbf{1}[r_i(A_D) \leq 10],\ \mathbf{1}[r_i(A_C) \leq 10], \\ &\mathbf{1}[r_i(A_D) \leq 10 \ \wedge\ r_i(A_C) > 10] \Big), \qquad N = 2{,}400. \end{aligned} \tag{41}$$

P@10 measures partner promotion, S@10 measures source retention, and CS@10 measures clean replacement: the partner appears while the source disappears. Every arm is evaluated on the same directed clozes and with the same endpoint definition.

G Relationship to Existing Readout Methods

Methods for interpreting hidden states differ in how they define concepts, construct concept representations, and use the source activation. These choices determine what a readout can explain and how the explanation can be tested. Our method connects three operations within a common lens geometry. A vocabulary ranking supplies first-token clues, the frozen model recovers directions for the completed concepts, and a source-state score evaluates those directions alongside vocabulary tokens. The comparisons below explain how this combination relates to activation verbalization, feature dictionaries, and causal intervention.

G.1 Template Lens and Oracle Lens

Vocabulary lenses provide the starting point for our approach. Logit Lens projects hidden states through the unembedding, Tuned Lens learns layer-specific affine maps, and J-lens uses averaged Jacobians to account for downstream processing (nostalgebraist, 2020; Belrose et al., 2025; Gurnee et al., 2026). R-lens modifies the J-lens construction through relevance propagation to improve early-layer readouts (Blank et al., 2026). Each lens assigns scores to vocabulary tokens. Our method uses those scores to identify promising first tokens, then extends the readout to complete concepts that span several tokens. Appendix I describes the shared interface for J-lens and R-lens.

Template Lens extends the readout with a fixed vocabulary of multi-token concepts (Gurnee et al., 2026). Each entry receives a representation before evaluation, and inference scores the source state against the stored collection. Our method constructs the candidate set for each query. **PROPOSE** completes first-token clues using the source prompt, and **SCORE** recovers the resulting concept representations from carrier hidden states. Transport places those representations in the source layer's readout geometry. Appendix C compares the two approaches through vocabulary coverage and through scoring on a shared candidate set, separating target availability from representation quality.

Oracle Lens learns to propose concepts from activations and reconstruct concept directions from text (Gurnee et al., 2026). The learned components allow the method to generate descriptions outside a fixed inventory. Our method obtains both candidate text and candidate directions from the frozen subject model, using a calibrated carrier reader for representation recovery. The source activation enters through the lens ranking and the final score, while recovery receives the candidate text alone. Appendix J compares the generated readouts at a shared activation interface, and Appendix H describes the corresponding preparation and inference costs.

G.2 Patchscopes and SelfIE

Patchscopes and SelfIE generate natural-language interpretations by supplying a source activation to a language model during generation (Ghandeharioun et al., 2024; Chen et al., 2024). This interface offers flexible multi-token descriptions with little preparation. Reliable readout requires checking whether these descriptions reflect information in the source activation.

Faithful-Patchscopes shows that familiar linguistic associations can override contextual information in the patched activation, and mitigates this bias through contrastive logit recalibration (Gong et al., 2026). In related introspection tasks, Lindsey (2026) finds that detection of injected concepts is unreliable and depends on the injection layer and strength. Singh et al. (2026, Section 5.3) further reports sensitivity to prompt wording and difficulty distinguishing activation interventions from textual manipulations. These findings motivate explicit checks that a readout is supported by the source activation.

Concept Lens separates candidate generation from source-conditioned scoring. **PROPOSE** uses token clues and the source context to propose concepts, while **SCORE** recovers each representation from candidate text alone. The final score measures agreement with the source activation in the chosen lens's geometry and is independent of the candidate's generation probability. Holding candidates and representations fixed lets us test whether concept selection follows changes in the source activation (Appendix D.3). Controlled reconstruction tests concept identity (Section 4.2), and concept swaps test the predicted behavioral effects (Appendix F).

G.3 Natural Language Autoencoders

Natural language autoencoders pair an activation verbalizer with an activation reconstructor (Fraser-Taliente et al., 2026). The verbalizer maps a source activation to a natural-language description, and the reconstructor maps that description back to an activation. Joint training encourages the description to preserve information needed for reconstruction. The resulting text can express extended explanations of model behavior and has been evaluated in model auditing. The reconstruction objective gives the description an activation-level target, providing a way to assess how much of the original state the generated language retains.

Our readout evaluates the support for individual multi-token concepts. Candidate completion uses source text and a first-token clue, while the carrier reader recovers each direction from candidate text alone. **SCORE** then compares the direction with the source activation under vocabulary competition. This concept-level formulation provides two natural checks. Held-out singleton reconstruction tests whether the carrier reader preserves concept identity, and fixed-candidate state swaps test whether the ranking responds to the source. Both checks use the frozen subject model and the same calibrated carrier reader used for multi-token readout.

G.4 Sparse Autoencoders and Feature Dictionaries

Sparse autoencoders (SAEs) learn a dictionary that represents a hidden state as a sparse combination of learned feature directions (Huben et al., 2024). The feature vocabulary is the set of learned latent coordinates, and interpretability is studied by examining the examples and behavior associated with each coordinate. Recent SAE studies scale this dictionary approach to larger models and use sparse features to recover causal graphs (Bricken et al., 2023; Templeton et al., 2024; Marks et al., 2025). This setup addresses decomposition of an activation into simultaneously active features.

Our unit is a natural-language concept C whose identity is specified by candidate text. Placing the text in a carrier prompt yields subsequent hidden states from which the carrier reader recovers v_C. Transport maps the direction into the chosen lens's geometry for scoring alongside native vocabulary directions. The candidate inventory can therefore vary with the query while the readout space remains fixed. SAE features and our concept representations both support causal tests. Their respective units are learned coordinates in a feature dictionary and linguistically specified concepts recovered from the subject model.

G.5 Activation Steering and Concept Representations

Activation addition and contrastive activation addition construct steering directions from behavioral or prompt contrasts (Turner et al., 2024; Rimsky et al., 2024). A common construction takes the difference between mean positive and negative activations,

$$v_{\text{contrast}} = \mathbb{E}[h_{\text{positive}}] - \mathbb{E}[h_{\text{negative}}],$$

and adds a scaled version during generation. Inference-time intervention, representation engineering, and function vectors similarly use behavioral contrasts or task demonstrations to obtain directions (Li et al., 2023; Zou et al., 2025; Todd et al., 2024). Related work studies factual associations through knowledge neurons (Dai et al., 2022), targeted weight updates in ROME and MEMIT (Meng et al., 2022; 2023), and analyses of factual recall (Geva et al., 2023). These approaches connect internal representations with changes in model behavior through different choices of intervention target.

Our method recovers v_C by centering and pooling the frozen model's subsequent hidden states for candidate text. The chosen lens then transports the representation to the source layer for scoring and intervention. A candidate's direction is thus defined before the causal trial and independently of its outcome. The paired tasks supply a concrete prediction for the edit. Replacing the source concept with its partner should promote the answer associated with that partner under the same relation. Appendix F tests this prediction with deletion, addition, and direction-specific controls.

G.6 Scope of the Contribution

Several lines of work show that semantic units extend across tokenizer boundaries. Studies of token erasure and inner lexicons identify word-level structure across subword tokens (Feucht et al., 2024; Kaplan et al., 2025), while latent-reasoning studies identify intermediate entities before the final answer (Yang et al., 2024; Biran et al., 2024). Future Lens predicts later tokens from a single hidden state (Pal et al., 2023), and Entity Lens reconstructs multi-token entity mentions from entity representations (Morand et al., 2025). These findings motivate readouts whose semantic units can contain several tokens, with a representation that can be examined as a whole.

Our contribution connects open-vocabulary concept identification, representation recovery, joint vocabulary scoring, and causal replacement in one pipeline. Each proposed concept receives a direction recovered from the frozen model and a score in the selected lens's geometry. The same direction can then be used to test the behavioral prediction implied by the concept label. This connection gives a readout more than a textual interpretation. The output supplies both a ranked hypothesis about the source state and an intervention target whose effect can be evaluated on the paired task.

H Computational Footprint and Online Efficiency

A readout method requires both one-time preparation and computation for each new query. Separating these costs makes the comparison useful across different deployment settings, from a small number of inspections to repeated analysis of many prompts. We use Qwen3.6-27B as the common subject model. Table 18 reports preparation in accelerator time, external API spending, and persistent storage, while Table 19 reports online inference. Model pretraining and weight loading are excluded from both comparisons. A100-hours sum device-hours across all GPUs used in a preparation run.

H.1 One-Time Preparation

The preparation comparison covers the model-specific artifacts needed to run each method. Our method calibrates the carrier reader on singleton concepts once a J-lens is available. Template Lens uses the same lens and additionally constructs a fixed inventory of concept representations. We treat the J-lens fit as a shared prerequisite and exclude its cost from both method-specific rows. This accounting separates the cost of obtaining the underlying vocabulary lens from the additional work required to extend that lens to multi-token concepts.

The Oracle Lens configuration uses the public Qwen3.6-27B proposer, reconstructor, and dictionary for one source layer. We estimate the cost of constructing this configuration by running a subset of the public training procedure, measuring accelerator usage, and extrapolating to the full training configuration. The resulting estimate is $2{,}000$–$4{,}000$ A100-hours for one layer. The table marks this quantity as an estimate and states the associated layer coverage. Additional source layers require their own layer-specific artifacts, so coverage is part of the preparation comparison.

On Qwen3.6-27B, our singleton-based carrier-reader calibration takes approximately 0.05–0.1 A100-hours and supports all 52 source layers. Calibration fixes the carrier positions, background means, and pooling weights; the frozen model supplies candidate completions and carrier states at inference time. The procedure requires no external API calls or separately trained proposer and reconstructor. A single calibrated carrier reader can therefore serve many source layers through lens-based transport. The small additional preparation cost makes the method practical for the multi-model and multi-layer evaluation used in this study.

Table 18: **One-time preparation for multi-token readout on Qwen3.6-27B.** Our method and Template Lens share an existing J-lens. Singleton-based carrier-reader calibration supports 52 source layers. The Oracle Lens row uses the public single-layer configuration, with training cost estimated from trial runs and extrapolated to the full configuration.

Method	Source layers	A100-hours	API cost (USD)	Storage (GiB)
Ours	52	0.05–0.1	0	5
Template Lens	52	20–30	1,500–2,000	43
Oracle Lens	1 (released)	2,000–4,000[†]	0	169

[†]Estimated by measuring accelerator usage on trial runs of the public Oracle Lens implementation and extrapolating to its full training configuration.

H.2 Online Readout Cost

One online query evaluates a prompt over the source layers listed in Table 19. Our method and Template Lens cover the source layers used in their accuracy experiments, while the public Oracle Lens configuration reads one source layer. Reporting the layer count makes the unit of work explicit. A request that returns rankings across a layer interval supports a different analysis from a request at one residual position. The latency and memory measurements describe each reported configuration together with the amount of readout it provides.

The p50 and p95 latencies summarize typical and slower requests after model warm-up. Each measurement includes source-state extraction and all subsequent online operations, with setup, model loading, and tokenization excluded. Peak VRAM is reported per GPU for the corresponding configuration. Together, the layer count, latency, and peak memory characterize the scope, interactive delay, and hardware requirements of each online configuration. Appendix K.3 describes how our implementation shares work across candidates and source layers.

Table 19: **Online readout cost on Qwen3.6-27B at batch size one.** End-to-end latencies include source-state extraction and all online readout operations after warm-up. Source layers gives the coverage of one request, and peak VRAM is reported per GPU. Oracle Lens uses the public single-layer configuration.

Method	Source layers	p50 E2E (s)	p95 E2E (s)	Peak VRAM / GPU (GiB)
Ours	52	4.4	4.7	70
Template Lens	52	0.56	0.68	55
Oracle Lens	1 (released)	5.10	5.32	55

The preparation and online measurements reflect different ways of allocating computation. Template Lens performs concept-representation construction in advance and uses the stored inventory at query time. Our method recovers representations for the candidates proposed for each query, allowing the concept inventory to adapt to the prompt. Shared-prefix decoding, cross-layer batching, and vectorized execution reduce the repeated work in this process. Table 23 measures their cumulative effect. The cost comparison therefore connects the choice of concept inventory with the work performed before and during a readout.

I A Pluggable Proposal and Lens Interface

I.1 Separating Candidate Proposal from Readout Geometry

The extension has two orthogonal interfaces. A candidate proposer Π maps a source prompt and an optional ranked vocabulary clue to a candidate set,

$$\Pi : \big(x, \rho_A(h_{\ell_s}); \mathcal{B}\big) \longmapsto \mathcal{C}_\Pi(x),$$

where $\rho_A(h_{\ell_s})$ is the vocabulary ranking supplied by a lens and $\mathcal{B}$ denotes the fixed search budget. A readout lens supplies a layerwise linear map A_ℓ used by **SCORE**; throughout this appendix $A_\ell = J_\ell$ for J-lens and $A_\ell = R_\ell$ for R-lens. The lens-seeded proposer used in our experiments reads $\rho_A(h_{\ell_s})$, whereas a direct proposer can produce $\mathcal{C}_\Pi(x)$ without a lens clue (Appendix D). Both candidate sets are then scored with the same carrier-reader and lens interface.

The frozen subject model and decoder are unchanged when either module is replaced. Rank ranges, decoding budgets, and filtering rules remain fixed as specified in Appendix K.1. End-to-end J/R rows therefore compare a complete proposer–lens composition and can include a proposal effect. Rows in the oracle-candidate setting add the annotated target to each proposer's own candidate set $\mathcal{C}_\Pi(x)$, which removes proposal failures and isolates the lens-dependent scoring geometry.

I.2 R-LENS READOUT GEOMETRY

R-lens changes the backward rules used to construct the readout map while preserving the subject model's forward values (Blank et al., 2026). Let $\mathcal{H}_\ell = \mathbb{R}^d$ be the layer-ℓ residual space and $\mathcal{H}_L = \mathbb{R}^d$ the final-residual space. Fix a forward trace ξ. Writing $D_k(\xi)$ for an ordinary local Jacobian, J-lens uses

$$J_{\ell,\xi} = D_L(\xi)\cdots D_{\ell+1}(\xi), \qquad J_\ell = \mathbb{E}_\xi[J_{\ell,\xi}]. \tag{42}$$

R-lens keeps the same forward trace but replaces the local backward factors with the relevance-propagation factors $\widetilde{D}_k(\xi)$:

$$R_{\ell,\xi} = \widetilde{D}_L(\xi)\cdots \widetilde{D}_{\ell+1}(\xi), \qquad R_\ell = \mathbb{E}_\xi[R_{\ell,\xi}]. \tag{43}$$

The LN, identity, and half rules used by R-lens replace local propagation coefficients (for example, by detaching an LN denominator or reweighting the two branches of a gated MLP); they do not introduce a new hidden state, unembedding, or coordinate system. Equivalently, $\widetilde{D}_k = D_k + E_k$ for a rule-dependent path reweighting E_k.

Consequently, both maps have the same domain and codomain,

$$J_\ell, R_\ell : \mathcal{H}_\ell \longrightarrow \mathcal{H}_L,$$

and both readouts use the same final normalization $N(z) = \Gamma z/r(z)$ and unembedding W_U. Let $u_w \in \mathcal{H}_L$ denote the column form of the unembedding row for vocabulary token w. For either lens,

$$[W_U N(A_\ell h)]_w = \frac{\langle d_\ell^A(w), h\rangle}{r(A_\ell h)}, \qquad d_\ell^A(w) = A_\ell^\top \Gamma^\top u_w \in \mathcal{H}_\ell, \quad A \in \{J, R\}. \tag{44}$$

Because $r(A_\ell h) > 0$ is common to all w, the corresponding vocabulary ranking is determined by the inner products in Equation 44. Thus the J- and R-lens token directions are covectors in the same source residual space, while $A_\ell h$ and the native rows u_w use the same final residual coordinates. A carrier state is likewise an element of the unchanged $\mathcal{H}_{\ell_c}$. R-lens can change which directions are salient, especially on early layers, but it does not create a separate carrier, vocabulary, or output space.

The following relation makes the distinction between a common representation space and a common metric explicit.

Proposition I.1 (Coordinate relation between the two lenses)**.** *If J_ℓ and R_ℓ are nonsingular, define $B_\ell = J_\ell^{-1} R_\ell$. Then*

$$R_\ell = J_\ell B_\ell, \qquad d_\ell^R(w) = B_\ell^\top d_\ell^J(w), \qquad G_\ell^R = B_\ell^\top G_\ell^J B_\ell, \qquad G_\ell^A = A_\ell^\top \Gamma^\top \Gamma A_\ell. \tag{45}$$

Hence B_ℓ is an isometry from $(\mathcal{H}_\ell, G_\ell^R)$ to $(\mathcal{H}_\ell, G_\ell^J)$.

Proof. The first equality follows from the definition of B_ℓ. Substitution into $d_\ell^R(w) = R_\ell^\top \Gamma^\top u_w$ and $G_\ell^R = R_\ell^\top \Gamma^\top \Gamma R_\ell$ gives the remaining equalities. Therefore $h^\top G_\ell^R h' = (B_\ell h)^\top G_\ell^J (B_\ell h')$ for all h, h'. □

This is an exact coordinate-space relation under the stated rank condition; it does not say that B_ℓ is close to the identity or that J-lens and R-lens assign identical scores. If a map is singular, its metric is defined on the corresponding quotient $\mathcal{H}_\ell/\ker(A_\ell)$, while the shared ambient residual and vocabulary interfaces in Equation 44 remain unchanged. In particular, the relation maps every native J-lens token covector to its R-lens counterpart by $B_\ell^\top$, but a frozen J-calibrated carrier reader does not explicitly apply that transform. Its transfer to R is therefore an empirical compatibility question addressed in Appendix I.4.

For source layer ℓ_s and carrier layer ℓ_c, we use the final normalization $N(z) = \Gamma z/r(z)$ from Appendix E.3 and define

$$\begin{aligned} q^A_{\ell_s} &= N(A_{\ell_s} h_{\ell_s}), \\ v^A_{C,\ell_s} &= (A^\top_{\ell_s} A_{\ell_s} + \lambda^{\text{ridge}}_{\ell_s} I_d)^{-1} A^\top_{\ell_s} A_{\ell_c} \Phi_A(C), \\ p^A_{C,\ell_s} &= \text{unit}\big(N(A_{\ell_s} v^A_{C,\ell_s})\big)\,. \end{aligned} \tag{46}$$

Here $q^A_{\ell_s}$ is the source query, v^A_{C,ℓ_s} is the transported concept representation, and p^A_{C,ℓ_s} is its normalized scoring direction. The ridge coefficient and same-layer convention follow Appendix K.2. **SCORE** compares concept directions and normalized unembedding rows against the same query, retaining a common scale for concept and vocabulary competition. The carrier hidden states can be reused because the model's forward computation is unchanged. The pooling weights in Φ_A are calibrated against the selected lens's native directions $d^A_{\ell_c}(w)$ using the fixed singleton calibration split.

The lens map induces a metric on residual-stream directions. For nonzero lens effects,

$$G^A_{\ell_s} = A^\top_{\ell_s} \Gamma^\top \Gamma A_{\ell_s}, \qquad \cos(N(A_{\ell_s} v), N(A_{\ell_s} h)) = \frac{v^\top G^A_{\ell_s} h}{\sqrt{v^\top G^A_{\ell_s} v}\sqrt{h^\top G^A_{\ell_s} h}}. \tag{47}$$

A change from J to R therefore changes the metric through which source and candidate representations are compared, while leaving the underlying vector space and unembedding interface fixed. Writing $\Delta_{\ell_s} = R_{\ell_s} - J_{\ell_s}$ makes both the token-direction and metric changes explicit:

$$d^R_{\ell_s}(w) - d^J_{\ell_s}(w) = \Delta^\top_{\ell_s} \Gamma^\top u_w, \qquad G^R_{\ell_s} - G^J_{\ell_s} = J^\top_{\ell_s} \Gamma^\top \Gamma \Delta_{\ell_s} + \Delta^\top_{\ell_s} \Gamma^\top \Gamma J_{\ell_s} + \Delta^\top_{\ell_s} \Gamma^\top \Gamma \Delta_{\ell_s}. \tag{48}$$

The equality of spaces is exact at the interface level; whether the two metrics are close on the directions used by this method is an empirical, task-local question. When the carrier reader is calibrated for R-lens, calibration, transport, and scoring all use R; the singleton directions used to calibrate the carrier reader and the vocabulary directions that compete with a recovered concept then share one geometry. We distinguish this matched R-calibrated setting from a fixed J-calibrated/R-lens ablation, rather than attributing a difference between them to scoring alone.

I.3 Matched J-lens and R-lens Evaluation

For a seeded proposer, write Π^A_{seed} for the same frozen decoder, scaffold, rank ranges, and search budget, supplied with the ranking $\rho_A(h_{\ell_s})$. The reported end-to-end rows compare the two compositions (Π^J_{seed}, J) and (Π^R_{seed}, R) on the same subject-model forward pass, source positions, candidate budget, decoding rule, and model-specific workspace band. The R-lens row replaces the lens-dependent seed ranking, transport, and scoring maps with R. Where the frozen J carrier reader is reused, it is labeled J-calibrated/R-lens rather than an R-calibrated carrier reader.

Candidate lists may differ because Π^J_{seed} and Π^R_{seed} receive different rankings. The end-to-end comparison therefore measures the complete proposer–lens composition. The direct-proposal setting in Appendix D changes candidate construction while leaving the carrier reader and scoring geometry unchanged. The oracle-candidate setting adds the annotated target to each lens's own candidate set and gives the cleaner J/R comparison of **SCORE**; the cross-lens carrier-reader check is reported separately in Appendix I.4.

Table 2 evaluates the interface at two levels. End-to-end readout measures proposal and scoring together, while the oracle-candidate setting fixes the available concepts and isolates scoring. An omitted target receives infinite rank, and recovered concepts compete with the complete vocabulary under the selected lens. Each matched J/R pair uses a common target layer and fitting corpus. We train all R-lens maps ourselves rather than use an official release. For Llama-3.1-8B, Gemma-3-12B, and Qwen3-14B, we use the same 500 datasets as the matched J-lens fit and change only the lens operator. For Llama-3.3-70B and Qwen3.6-27B, we use 100 examples because of the larger model scale; the same held-out evaluation protocol applies to all five models. These controls connect differences in the final readout to the lens choice while preserving the task, model, workspace band, and evaluation budget.

Table 20: **Held-out cross-lens compatibility check.** Native@10 is strict retrieval of the matching leading-space token from the complete normalized vocabulary; $E_{B\to R}$ uses the fixed cached R-lens source queries at the carrier layer (lower is better) and is reported only for this auxiliary geometry diagnostic. "Shuffled/R" is a no-fixed-point permutation of held-out recovered R-calibrated prototypes relative to their native tokens. No **PROPOSE** step is run.

Model	Arm	Native@10 (%)	Median cosine	$E_{B\to R}$
Llama-3.1-8B	J-cal./R	99.6	0.342	0.0377
	R-cal./R	99.6	0.346	0.0382
	Shuffled/R	0.0	-0.002	0.0439
Gemma-3-12B	J-cal./R	99.8	0.453	0.0494
	R-cal./R	99.8	0.454	0.0495
	Shuffled/R	0.0	-0.004	0.0592
Qwen3-14B	J-cal./R	99.6	0.371	0.0746
	R-cal./R	99.6	0.373	0.0738
	Shuffled/R	0.0	-0.003	0.0764

I.4 Held-Out Cross-Lens Compatibility Check

The held-out check tests cross-lens compatibility independently of candidate proposal. We use the 500-concept singleton partition and retrieve each recovered direction's matching token from the complete native vocabulary. The fully matched R-cal./R arm serves as the positive control. The J-cal./R arm reaches 99.6–99.8% Native@10 across the three models, matching R-cal./R at the reported precision. The key cross-lens question is whether a carrier reader calibrated to J native directions still preserves token identity when its recovered state is read in R geometry.

For a held-out singleton w_i, let Φ_B denote the carrier pooling rule calibrated against native directions from lens $B \in \{J, R\}$. Source and carrier are both at layer c, so the explicit same-layer transport convention gives the R-scored prototype

$$p_i^{B\to R} = \mathrm{unit}(N(R_c \Phi_B(w_i))) \,. \tag{49}$$

The J-cal./R arm is $p_i^{J\to R}$: its pooling weights were chosen using J native directions, but it is retrieved and scored with R-lens. The R-cal./R arm is the fully matched $p_i^{R\to R}$ baseline. Both use the same split and unchanged carrier forward states. The query-side diagnostic uses a fixed cached bank of source residuals; it is an auxiliary geometry check, separate from the 2,400-cloze paired benchmark used for end-to-end evaluation. The negative control is a fixed no-fixed-point permutation of the 500 R-calibrated prototypes relative to their held-out native tokens. It preserves the carrier reader, R-lens map, prototype distribution, and complete vocabulary competition, but destroys the intended concept-to-direction assignment.

For $e_i^{B\to R} = p_i^{B\to R} - \mathrm{unit}(u_{w_i})$, we also measure normalized score error on the fixed R-lens source queries,

$$\begin{aligned} E_{B\to R}^2 &= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left((e_i^{B\to R})^\top \widehat{q}_j^R \right)^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} (e_i^{B\to R})^\top M_R e_i^{B\to R}, \qquad M_R = \frac{1}{m} \sum_{j=1}^{m} \widehat{q}_j^R (\widehat{q}_j^R)^\top . \end{aligned} \tag{50}$$

Here $\widehat{q}_j^R = \mathrm{unit}(q_j^R)$. Native-token retrieval tests concept identity; $E_{B\to R}$ weights the remaining recovery error by its effect on the source-query scores used by **SCORE**.

Table 20 reports the result. The cross-lens J-cal./R arm retains 99.6–99.8% Native@10, matching R-cal./R at the reported precision. Refitting in R geometry changes median native cosine by at most 0.004 and $E_{B\to R}$ by at most 0.0008; it does not uniformly reduce the score error. Thus the result supports local compatibility of the J- and R-induced geometries on recovered carrier directions and the source queries used here, rather than a claim that their metrics are globally identical. In contrast, the assignment control retrieves no matching token in the top 10 and has near-zero median cosine in every model. No **PROPOSE** step is run: candidate strings, carrier states, and held-out source queries are fixed in advance.

J Comparison with Oracle Lens

Oracle Lens provides a close comparison because it verbalizes hidden activations with open-vocabulary, multi-token phrases (Gurnee et al., 2026). Its oracle model generates descriptions from an injected activation, and the full pipeline can map those descriptions back to directions for activation reconstruction. Our benchmark gives the readout a specific semantic target, the intermediate concept required by the cloze. We compare the generated concepts directly at a shared residual state. The evaluation asks whether the different readout interfaces recover the same task-relevant concept from the same point in the model's computation.

J.1 Activation Interface

We use the public Qwen3.6-27B implementation, which provides the oracle model together with the activation-injection and generation procedure. The implementation reads a raw residual activation at the input to decoder block 45. Under our post-block indexing convention, the same representation is the L44 residual state. Matching the representation itself removes ambiguity caused by different layer labels. The public interface supplies a reproducible comparison point at which Oracle Lens and our two lens variants can read exactly the same subject-model activation.

The comparison follows the layer coverage of this implementation. The original study uses a trained Haiku-4.5 oracle that is not publicly released, and other Oracle-Lens-style checkpoints expose different layer configurations. We use the self-contained Qwen3.6-27B inference interface to keep activation extraction and generation reproducible. The resulting experiment evaluates concept recovery at the shared block-45 input. The main multi-layer experiments address the complementary question of where concepts become readable across the subject model, using the model-specific workspace bands.

J.2 Matched Direct Comparison

The direct comparison uses the same $2{,}400$ directed clozes, subject model, and source position for all three methods. Our J-lens and R-lens variants read the post-block L44 state, and Oracle Lens receives the identical representation at the input to block 45. Each method returns up to ten outputs for a prompt. Matching the residual state fixes the information available to the readout, while matching the output budget fixes the number of opportunities to recover the target concept. Table 21 reports the resulting comparison.

Oracle Lens follows its released decoding procedure and generates natural-language phrases, while our variants return ranked concept labels. We use *Mention@10* as the primary metric to accommodate the descriptive form of Oracle Lens outputs. A prediction is correct when the normalized gold concept appears as a complete, word-bounded span in any of the ten outputs. *Exact@10* additionally requires an exact normalized match to the canonical concept string. The two metrics coincide for our ranked concept outputs. Reporting both metrics preserves a common semantic target while making the role of output format explicit.

Table 21: **Concept readout at a shared Qwen3.6-27B residual state.** All methods read the same activation on the same $2{,}400$ benchmark examples. Mention@10 accommodates free-form descriptions, and Exact@10 measures exact recovery of the canonical concept string. Input to block 45 and post-block L44 denote the same residual position.

Method	Residual position	Mention@10 (%)	Exact@10 (%)
Ours (J-lens)	Post-block L44	1.08	1.08
Ours (R-lens)	Post-block L44	2.79	2.79
Oracle Lens	Input to block 45	2.17	0.38

This is a deliberately fixed-layer comparison using the shared-interface Mention@10/Exact@10 protocol, whereas Table 2 reports end-to-end joint-vocabulary Rank@10 aggregated over the workspace band. For Qwen3.6-27B, Figure 5 shows that dense first-token readout occurs mainly around layers 50–60, whereas the shared L44 interface lies just before that dense region and has sparse readout. The low single-layer rates in Table 21 are therefore expected and are not inconsistent with the higher band-level rates in the main table.

J.3 Readout Design and Model-Specific Preparation

The two methods allocate model-specific preparation differently. Oracle Lens trains a concept-to-activation reconstructor, constructs a dictionary of phrase directions, and decomposes source activations to obtain supervision for the oracle model (Gurnee et al., 2026). The oracle is then trained and refined with a reconstruction objective. Applying the pipeline to an additional subject model involves activation collection, reconstruction training, teacher decomposition, and oracle training. These stages produce a learned activation-to-language interface tailored to the subject model and residual position.

Our method uses the frozen subject model for candidate completion and carrier-state extraction. Once the underlying lens is available, lightweight singleton-based calibration fixes the carrier reader, and lens-based transport supports the selected source layers. This design makes the same procedure usable across the model-specific workspace bands and subject models in the main experiments. The shared-state comparison evaluates concept recovery at a common interface, while Appendix H compares the preparation and online work associated with each configuration. Together, the evaluations relate readout quality to the resources required to obtain it.

K Implementation Details

This appendix specifies candidate generation, carrier calibration, and efficient execution of the readout in Section 3. The evaluation protocol is given in Appendix B.3, and the recovery and transport analysis is given in Appendix E.

K.1 Candidate Generation

PROPOSE uses the Scaffold Prompt in Section 3.1. At each source layer, we divide the first 100 entries of the selected lens ranking into three disjoint rank ranges. Each rank range starts from at most 20 lexical seeds and maintains an independent active frontier. Table 22 gives the search widths.

Table 22: **Search budgets for candidate generation.** Rank ranges are inclusive. Each rank range keeps its own active prefixes during continuation.

First-token rank	Maximum lexical seeds	Active width
1–20	20	12
21–50	20	6
51–100	20	6

The seed fixes the first concept token. Continuation uses the frozen model's next-token distribution with nucleus probability 0.85, at most 10 expansions per step, and at most four concept tokens including the seed. Each rank range retains high-probability prefixes within its active width. A candidate is completed at a valid termination or the length limit. We discard special and formatting tokens except those used to terminate a candidate.

We merge completed candidates across rank ranges, remove duplicates and strict prefixes of longer retained candidates, and keep at most 25 candidates. Proposal probabilities guide the search and candidate retention; the final readout uses the source-state scores in Equation 8. These search settings are fixed across the lens comparisons in Appendix I.1. End-to-end Rank@10 counts a cloze as successful when the complete target concept is placed in the joint top 10. Proposal searches first-token ranks 1–100, so a first token outside the lens top 10 is not itself a failure.

K.2 Carrier Calibration and Vector Construction

The Carrier Prompt in Section 3.2 places the concept before fixed text that includes `The capital of France is Paris.` We extract hidden states at positions following the complete concept and align extraction positions to this fixed suffix. The carrier layer ℓ_c, extraction window $\mathcal{T}$, position means, and pooling weights are selected using the 300-concept calibration partition in Appendix A.4. The carrier layers are L27, L43, L37, L58, and L60 for Llama-3.1-8B, Gemma-3-12B, Qwen3-14B, Qwen3.6-27B, and Llama-3.3-70B, respectively. The carrier reader remains fixed for held-out reconstruction, multi-token readout, and causal evaluation.

For the J-lens instantiation, let $\mathcal{S}$ be the calibration concepts and let $d_{\ell_c}(w)$ be the native J-lens vector of singleton w. We center carrier hidden states by suffix position and center the reference token vectors over the same calibration set,

$$\mu_t = \frac{1}{|\mathcal{S}|}\sum_{w\in\mathcal{S}} h_{\ell_c,t}(\tau(w)), \qquad \bar{d}_{\ell_c} = \frac{1}{|\mathcal{S}|}\sum_{w\in\mathcal{S}} d_{\ell_c}(w). \tag{51}$$

For each extraction position, we measure mean cosine agreement with the centered reference vectors. The positive part of this agreement determines the pooling weight,

$$\rho_t = \max\left\{0, \frac{1}{|\mathcal{S}|}\sum_{w\in\mathcal{S}} \cos\big(h_{\ell_c,t}(\tau(w)) - \mu_t, d_{\ell_c}(w) - \bar{d}_{\ell_c}\big)\right\}, \qquad \alpha_t = \frac{\rho_t}{\sum_{u\in\mathcal{T}}\rho_u}. \tag{52}$$

The selected window has positive total agreement. Equation 4 applies these fixed means and weights to each candidate's carrier hidden states. Candidate text is the only input that changes during vector recovery. Appendix E.1 analyzes the signal and residual in the pooled direction. For the R-lens instantiation, the same procedure is calibrated against the corresponding R-lens directions as described in Appendix I.2.

For different carrier and source layers, the J-lens transport in Equation 6 has the ridge solution

$$v_C^{(\ell_s)} = \left(J_{\ell_s}^\top J_{\ell_s} + \lambda_{\ell_s}^{\text{ridge}} I_d\right)^{-1} J_{\ell_s}^\top J_{\ell_c} v_C^{(\ell_c)}, \qquad \lambda_{\ell_s}^{\text{ridge}} = 10^{-2}\frac{\|J_{\ell_s}\|_F^2}{d}, \tag{53}$$

where d is the residual dimension, I_d is the identity matrix, and $\|\cdot\|_F$ is the Frobenius norm. When $\ell_s = \ell_c$, we use the recovered vector directly. **SCORE** applies the model's final normalization to the transported effect and uses the shared scale $m_U = \text{median}_{w\in\mathcal{V}}\|u_w\|_2$ for concept and vocabulary scores. Appendix E.2 derives the spectral action of transport, and Appendix E.3 analyzes the resulting normalized scores. The R-lens instantiation replaces the corresponding J-lens maps with R-lens maps consistently for calibration, transport, and scoring, as specified in Appendix I.

K.3 Efficient Inference

An online readout returns concept rankings over the selected source layers for one prompt. **PROPOSE** repeatedly processes the same scaffold and overlapping candidate prefixes. Shared-prefix tree decoding reuses the corresponding key–value (KV) cache, and cross-layer batching combines decoder calls for several source layers. Candidate selection remains independent within each source layer and rank range.

Algorithm 1 computes the scaffold's KV cache once and reuses the cache across source layers. The frontier F contains candidate prefixes of equal length, tagged by source layer and rank range. Each decoding step processes a node's final token using its parent's KV cache. The resulting logits support both the termination check and the next expansion. Identical child prefixes share a decoder computation within each batch. ExpandPrune selects children independently within each layer and rank range and reuses their parents' KV caches.

Algorithm 1 Shared-prefix tree decoding

Input Scaffold $\text{Scaffold}(x)$, tagged seeds $\{S_\ell\}_{\ell\in\mathcal{L}}$, maximum depth D.
Output Retained candidates for each source layer.

$(K_0, z_0) \leftarrow$ Prefill($\text{Scaffold}(x)$); $\mathcal{C}_\ell \leftarrow \varnothing$ for all $\ell \in \mathcal{L}$
for each source layer $\ell \in \mathcal{L}$ **do**
  $F \leftarrow$ SeedFrontier(S_ℓ, K_0, z_0)
  for $d = 1, \ldots, D$, while $F \neq \varnothing$, **do**
    $(Z, K) \leftarrow$ DecodeStep(F) ▷ one new token per node
    $\mathcal{C}_\ell \leftarrow$ UpdateCompleted($\mathcal{C}_\ell, F, Z$)
    if $d < D$ **then** $F \leftarrow$ ExpandPrune(F, Z, K)
  end for
end for
return $\{$SelectCandidates($\mathcal{C}_\ell$)$\}_{\ell\in\mathcal{L}}$

For a fixed scaffold and candidate prefix, the decoder's continuation distribution is independent of the source-layer label. Algorithm 2 advances frontiers at the same depth from up to B source layers in one model call. UPDATECOMPLETED assigns each completed prefix to the candidate set for its tagged source layer. For L source layers and depth limit D, the upper bound on incremental decoder calls falls from LD to $D\lceil L/B \rceil$. The request batch size remains one; B controls how many source layers share a decoder batch.

Algorithm 2 Cross-layer batching of shared-prefix trees

Input Same as Algorithm 1, with layer batch size B.
Output Retained candidates for each source layer.

$(K_0, z_0) \leftarrow \textsc{Prefill}(\text{Scaffold}(x))$; $\mathcal{C}_\ell \leftarrow \varnothing$ for all $\ell \in \mathcal{L}$
for each group $G \subseteq \mathcal{L}$ in a partition with $|G| \le B$ **do**
  $F \leftarrow \bigcup_{\ell \in G} \textsc{SeedFrontier}(S_\ell, K_0, z_0)$
  for $d = 1, \ldots, D$, while $F \neq \varnothing$, **do**
    $(Z, K) \leftarrow \textsc{DecodeStep}(F)$ ▷ one call across source layers
    $(\mathcal{C}_\ell)_{\ell \in G} \leftarrow \textsc{UpdateCompleted}((\mathcal{C}_\ell)_{\ell \in G}, F, Z)$
    if $d < D$ **then** $F \leftarrow \textsc{ExpandPrune}(F, Z, K)$
  end for
end for
return $\{\textsc{SelectCandidates}(\mathcal{C}_\ell)\}_{\ell \in \mathcal{L}}$

The optimized implementations batch 16 carrier candidates per recovery pass. The combined implementation uses $B = 8$ source layers per decoder batch. The measured effect of these choices is reported below.

The third optimization is sequence-level (Seq) vectorization, called the *Vectorized policy* below. At each decoding depth it packs the active candidate sequences into one tensorized frontier, applies termination and pruning masks in parallel, and avoids per-candidate GPU–CPU synchronization. The resulting CARRY directions remain on the GPU until **SCORE** (GPU CARRY), so the vectorized policy is applied after the shared KV-tree and cross-layer batching stages.

All implementations use BF16 inference. Batching and vectorization can change the order of floating-point operations, so the optimized and sequential executions are not bitwise identical even though they use the same search budgets, pruning rules, and scoring definition. Small finite-precision differences matter primarily when candidate scores lie close to a pruning or ranking boundary, where a small numerical perturbation can change which prefix is retained. Such discrepancies therefore concentrate on marginal cases rather than changing the underlying readout procedure; final readout quality is evaluated by the task-level metrics in the main experiments, while Appendix K.4 reports execution agreement separately as an implementation diagnostic.

K.4 Measured Effect of Efficient Execution

We apply the three optimizations cumulatively while keeping the search budgets, expansion rules, scoring definition, and BF16 numerical precision fixed. Table 23 reports those medians linearly rescaled from the measured layer counts to each model's workspace-band width. Each request includes source-state extraction, lens readout, proposal, vector recovery, and joint scoring over the selected layers. Both the sequential reference and optimized implementations use BF16 inference. Measurements use one warm-up request and CUDA synchronization; tokenization and one-time preparation are excluded. The latency comparison therefore measures execution optimizations under the same numerical precision rather than a change in model precision.

The shared-prefix tree accounts for most of the reduction, while cross-layer batching and sequence-level vectorization provide further gains. Across the five evaluated models, the final stack reduces median latency by 8.73–$15.05\times$. With $B = 8$, top-1 agreement with sequential execution is 90.95%, 94.84%, and 93.26% for Llama-3.1-8B, Gemma-3-12B, and Qwen3-14B, respectively, over nonempty prompt–layer outputs. The optimized BF16 execution is therefore not numerically identical to the sequential reference. The remaining disagreements occur when small finite-precision differences cross discrete pruning or ranking boundaries, so they primarily affect near-boundary candidates and have limited effect on the aggregate readout. We report agreement to characterize

this numerical sensitivity; the task-level experiments remain the relevant evaluation of final readout quality.

Table 23: **Cumulative optimizations reduce online readout latency.** Entries are median end-to-end seconds per prompt on the same ten inputs. Columns apply the optimizations cumulatively; the final implementation uses $B = 8$ and the Vectorized policy. Parentheses give the workspace-band layer count. The speedup is the ratio of those rescaled times. All variants use BF16; optimized execution preserves the same search policy but is not bitwise identical to the sequential reference.

Model (layers)	Full forward	+ KV-tree	+ Cross-layer batching	+ Vectorized	Speedup
Llama-3.1-8B (26)	19.24	5.77	3.94	**1.54**	12.49×
Gemma-3-12B (39)	53.10	14.13	7.31	**3.61**	14.71×
Qwen3-14B (32)	33.26	8.40	4.66	**2.21**	15.05×
Qwen3.6-27B (52)	62.19	20.23	11.81	**4.27**	14.56×
Llama-3.3-70B (42)	117.1	29.88	24.11	**13.41**	8.73×