SHOWING VS. TELLING: STEERING WITH FEEDBACK VECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Input–output demonstrations can induce activation vectors that execute a task when injected into a language model. Prior work extends this idea to task instructions. We study *feedback vectors* extracted from rules, explanations, and critiques that specify a mapping without providing input–output examples. Across a suite of seven models spanning four architectures and 1B–9B parameters, we compare textual feedback with three vector constructions, and examine prompt format, representation, and out-of-distribution behavior. Under a Q/A prompt on the primary 1B model, the best of the three vector methods, selected on the training pool, outperforms textual critiques on five of six tasks, and a single fixed method does so on four; across all seven models, that fixed method exceeds critique tokens in 21 of 42 model–task pairs. Feedback-vector accuracy tracks the model's ability to use feedback as text across 146 conditions ($r = .86$). Across three critique prompt templates, however, best-of-methods vector accuracy varies less than accuracy from feedback presented as text. Feedback and demonstration vectors also generalize differently: feedback vectors can outperform function vectors when a regularity in the demonstrations breaks, while function vectors can be stronger on rare factual retrieval. Together with their largely different selected attention heads, these findings are consistent with task representations shaped differently by demonstrations and linguistic feedback.

1 INTRODUCTION

A large language model can be told what to do in two very different ways. It can be *shown*: a few input–output demonstrations placed in context induce the model to continue the pattern (Brown et al., 2020; Min et al., 2022). Or it can be *told*: a rule ("output the antonym of the input word"), an explanation ("the answer is the opposite because this task collects antonym pairs"), or a critique ("that was a synonym; it should be the opposite") specifies the same mapping in natural language, with no examples at all.

Showing is known to compress into a single direction in activation space: summing the outputs of a small set of causally identified attention heads yields a *function vector* (FV) that, added to a zero-shot forward pass, executes the demonstrated task (Hendel et al., 2023; Todd et al., 2024). Davidson et al. (2025) recently showed that task *instructions* yield such vectors too, through largely different heads. But an instruction is only one form of telling; people correct each other with explanations and critiques far more often than with example sets. Telling is also a puzzling channel: explanations added to prompts help only marginally (Lampinen et al., 2022), which leaves open whether the information in feedback is weak or the token channel that delivers it is lossy.

We define the *feedback vector* (FBV): a vector extracted from natural-language feedback containing no input–output pairs and injected into a zero-shot pass (Figure 1). Our causal-mediation construction applies the head scoring and vector assembly used for FVs to feedback-conditioned prompts. The conditioning text and corruption intervention differ between the two sources; we also test two residual-state constructions that do not select heads. We ask whether feedback vectors execute the specified task, how their selected heads compare with demonstration-derived heads, and when prompt format or distribution shift changes their effectiveness. Building on Davidson et al. (2025)'s findings for instructions, we examine rules, explanations, and critiques across models and prompt formats.

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

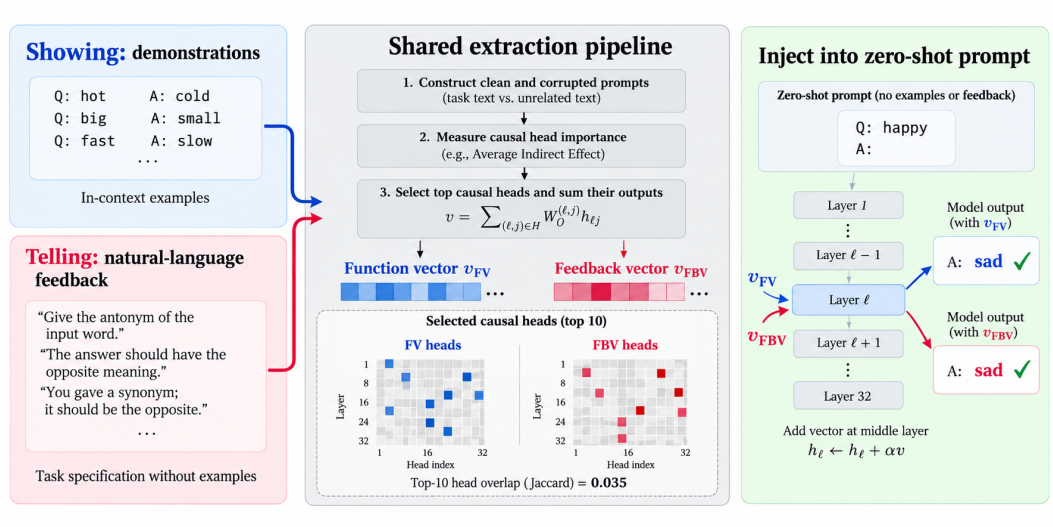


Figure 1: Method-C feedback vectors and function vectors share head scoring, vector assembly, and injection, but use different conditioning text and corruption interventions. The conditioning text is either input–output demonstrations (*showing*) or natural-language feedback containing no examples (*telling*). Causal mediation selects the attention heads that carry the task, their outputs are summed into a single vector, and that vector is added at a middle layer of a zero-shot prompt that contains no task information.

Our contributions are fourfold:

- **Task execution from feedback.** We extract vectors from example-free rules, explanations, and critiques and evaluate three constructions across seven models. The causal-mediation construction selects heads with little overlap with demonstration-derived FV heads, extending the instruction-side finding of Davidson et al. (2025).
- **Template sensitivity.** Across three critique prompt templates in two Llama models, the best-of-methods feedback-vector pipeline changes less in accuracy ($+.02$ to $+.12$) than critiques presented as tokens ($+.10$ to $+.24$); the ranking of a fixed extractor against tokens can reverse.
- **Complementary generalization.** On irregular past tenses at 1B, a contrastive FBV exceeds the FV (.76 vs. .63); on rare-country retrieval, FVs are more robust in the Llama models.
- **Token–vector association.** Across 146 model–task–flavor conditions, token and best-FBV accuracy correlate at r=.86. In the six conditions with token accuracy below .10, no method yields vector accuracy above .19.

2 Related Work

Task representations from demonstrations. Hendel et al. (2023) show that the hidden state at the end of an ICL prompt acts as a *task vector* that executes the demonstrated task when patched into a zero-shot pass; Todd et al. (2024) use causal mediation (Vig et al., 2020) to find attention heads whose summed outputs form a *function vector*. Related work compresses demonstrations via PCA (Liu et al., 2024), traces the circuit that assembles task information (Bakalova et al., 2025), learns task representations from attention heads (Saglam et al., 2025) or by optimization (Yang et al., 2025b) (our Method D), aligns task vectors with full ICL (Kwon et al., 2026), and decomposes function vectors into per-example sub-vectors that inherit the biases of their examples (Wang et al., 2026). All condition on input–output examples.

Steering from text. Inference-time interventions steer models with optimized latent vectors, activation addition, and contrastive directions (Subramani et al., 2022; Turner et al., 2023; Li et al., 2023; Rimsky et al., 2023; Zou et al., 2023), including instruction-derived directions that reinforce a constraint still present in the prompt (Stolfo et al., 2025). Our vectors must instead replace the task specification; Method B adapts contrastive steering to this setting. Closest to our work, Davidson et al. (2025) find that instruction- and demonstration-derived vectors both raise zero-shot accuracy but "mostly engage different attention heads." We extend this to explanations and critiques, which they leave to future work, add calibrated head-overlap baselines, and test feedback written after an observed error (Section 5.5).

Learning from explanations and feedback. Natural-language feedback supports prompting, revision, and training (Akyürek et al., 2023; Borges et al., 2024; Avsian et al., 2026; Madaan et al., 2023; Shinn et al., 2023; Luo et al., 2025); our grounded-feedback pipeline adapts the generate–verify–regenerate loop of Avsian et al. (2026). Yet explanations added to few-shot prompts yield only marginal gains (Lampinen et al., 2022), which motivates our question: is the bottleneck the information in feedback, or the token channel that delivers it? The "critique vector" of Phan et al. (2026) captures a reasoning model's own self-correction rather than a task specification distilled from external critique text.

3 Feedback Vectors

3.1 Preliminaries: function vectors

Let a decoder-only transformer have L layers, H attention heads per layer, and residual width d. Following Todd et al. (2024), for a task t with demonstration prompts $p_1, \ldots, p_N$ (each k input–output pairs followed by a query), let $\bar{\boldsymbol{h}}_{\ell j} \in \mathbb{R}^{d_h}$ be the mean output of head (ℓ, j) at the final token across clean prompts. Each head's *average indirect effect* is measured on corrupted prompts $\tilde{p}_i$ (demonstration labels shuffled):

$$\mathrm{AIE}(\ell, j) \;=\; \frac{1}{N} \sum_{i=1}^{N} \Big[P\big(y_i \,\big|\, \tilde{p}_i;\, \boldsymbol{a}_{\ell j} \leftarrow \bar{\boldsymbol{h}}_{\ell j}\big) - P\big(y_i \,\big|\, \tilde{p}_i\big) \Big], \tag{1}$$

the change in probability of the correct first answer token when head (ℓ, j)'s final-token output is patched to its clean mean. With $\mathbb{A}$ the top-m heads by AIE and $W_O^{(\ell j)}$ the head's slice of the output projection, the function vector is $\mathrm{FV}_t = \sum_{(\ell,j)\in\mathbb{A}} W_O^{(\ell j)}\, \bar{\boldsymbol{h}}_{\ell j} \in \mathbb{R}^d$, added to the residual stream at a middle layer of a zero-shot pass.

3.2 The feedback modality

A *feedback statement* for task t is a natural-language sentence that constrains the task mapping without exhibiting it. We author three flavors, distinguished pragmatically:

- **Rule**: an imperative statement of the mapping ("Output the antonym of the input word.");
- **Explanation**: a declarative statement with a rationale ("The correct answer is the word with the opposite meaning, because this task pairs each word with its antonym.");
- **Critique**: a second-person corrective, framed as a response to a prior wrong answer ("Wrong: you gave a synonym, but the task wants the antonym.").

These authored critiques *presuppose* an error: they are written per task, not in response to an answer the model actually produced. All experiments use authored statements except one analysis in Section 5.5, which uses a fourth source, *grounded feedback*: a critique written by an external model in response to a specific wrong answer the evaluated model produced, and kept only if it makes that model correct its answer as text.

A strict constraint governs authoring: *no feedback statement may contain an input–output pair*. Explanations in the style of Lampinen et al. (2022) attach to particular examples; including them would smuggle demonstrations into the feedback channel and contaminate the modality comparison.

Our explanations therefore justify the rule, not an instance of it. Each task receives ten paraphrases per flavor, and k sampled statements play the role that k demonstrations play for an FV.

3.3 Extraction methods

Feedback has no input–output structure to exploit, so the extraction itself must be defined. We implement four methods behind one interface and report which method each result uses.

Method A: separator readout (Hendel-style). Present k feedback statements followed by a query template ("⟨feedback⟩ Q: x' A:") for dummy queries x' drawn from the task's input distribution; read the final-token residual state at each layer, averaged over prompts. Injection replaces (or adds to) the residual state at one layer. Cheapest; no causal selection.

Method B: contrastive mean (CAA/Stolfo-style). The layerwise difference of mean final-token residual states between matched prompts with and without the feedback prefix, over neutral queries (Rimsky et al., 2023; Stolfo et al., 2025). Risk: captures a generic "an instruction is present" direction rather than task content. Matched queries hold the query fixed, although this contrast alone cannot separate task content from a generic feedback-prefix effect.

Method C: causal mediation (Todd-style; primary). Method C shares head scoring, vector assembly, and injection with the FV pipeline of Section 3.1, but uses different conditioning text and a different corruption intervention: the clean run conditions on feedback text ("⟨k statements⟩ Q: x A:") and the corrupted run replaces the statements with an unrelated task's feedback, matched in flavor and count (word-shuffling, the naive analogue of label shuffling, leaves the rule's content words intact and collapses the causal contrast; Appendix E). AIE selects heads, yielding $\mathrm{FBV}_t = \sum_{(\ell,j)\in\mathbb{A}'} W_O^{(\ell j)} \bar{\boldsymbol{h}}_{\ell j}^{\mathrm{fb}}$. Because Method C otherwise matches the FV procedure, head and geometric comparisons are aligned, though differences may still reflect conditioning and corruption.

Method D: optimized readout (supervised reference). Initialize from Method A (or B) and optimize the injected vector by gradient descent to maximize likelihood of correct outputs on a small labeled probe set, in the spirit of Yang et al. (2025b). This is not a feedback-only construction (it sees labels). We ran it for rule feedback on Llama-3.2-1B and Llama-3.1-8B; it improves on its initialization in some conditions but not all, so we report it as a supervised reference rather than a ceiling (Appendix F.3).

3.4 Injection

At evaluation, we inject a vector at the final prompt token of a query-only prompt, "Q: x A:", which contains neither feedback nor demonstrations. Method A replaces the residual state; Methods B and C, as well as FVs, use additive injection. Method D uses the injection mode of the method it is initialized from. We select the injection layer and, for additive interventions, the scale $\alpha \in \{1, 2, 4\}$ using first-token accuracy on the training/extraction pool. Additive FV and FBV comparisons use the same layer–scale search grid. There is no separate development set. Reported accuracies use full generation on evaluation queries disjoint from the training pool. For OOD experiments, settings are selected on the in-distribution training pool and frozen before ID and OOD test evaluation.

4 Experimental Setup

Tasks and data. We study six short-answer mappings from the Todd et al. (2024) suite: *antonym*, *present→past*, *singular→plural*, *capitalize*, *country→capital*, and *english→french*. Each task has 50 held-out evaluation queries, disjoint from the pool used to extract vectors and select injection settings. The evaluation queries remain fixed across extraction seeds. The irregular-verb OOD split instead contains 35 queries; all splits are described in Appendix D.

Models. We evaluate seven checkpoints from four model families: Llama-3.2-1B, Llama-3.1-8B, Llama-3.1-8B-Instruct (Llama Team, AI @ Meta, 2024), Qwen3-1.7B (Yang et al., 2025a), Gemma-2-2B, Gemma-2-9B (Gemma Team, Google DeepMind, 2024), and GPT-J-6B (Wang & Komat-

suzaki, 2021). Six are base models and one is instruction-tuned. All have function-vector reference and five-seed feedback-vector runs; additional analyses use subsets detailed in Appendix A.

Baselines and conditioning budgets. We compare (i) zero-shot generation without an intervention, (ii) *feedback-as-tokens*, which prepends the sampled feedback statements to the query, (iii) a demonstration-derived function vector (FV), and (iv) full in-context learning (ICL). The main feedback-channel comparison uses five statements for both textual feedback and Methods A/B/C. Its FV and ICL columns are ten-demonstration references from the separate reproduction experiment (Table 11); they are not matched-count controls for the five-statement results. In the OOD vector comparison, FVs and FBVs each use ten conditioning items, while feedback-as-tokens uses five statements.

Selection and reporting. Injection settings are selected by first-token accuracy on the training/extraction pool, as described in Section 3. There is no separate development set. We report exact-match accuracy from greedy full generation on held-out queries, truncating answers at the first newline and ignoring case except for casing tasks. Extraction seeds resample conditioning material, not evaluation queries. "Best FBV" is the extractor among Methods A/B/C with the highest training-pool accuracy, alongside its training-selected layer and scale; it excludes the supervised Method D. Table 1 selects one method per condition over seed-averaged training-pool accuracy; the template and correlation analyses select within each seed before averaging.

5 RESULTS

We first evaluate whether feedback-derived vectors induce the specified mapping on held-out inputs. We then examine sensitivity to prompt format, compare the heads and vector representations obtained from feedback and demonstrations, and evaluate their behavior under distribution shifts. Methods A/B/C are compared for task execution; the head-level analysis uses Method C, which shares the FV assembly procedure.

5.1 TASK EXECUTION FROM FEEDBACK

Table 1 compares feedback as tokens with Methods A–C. Across the 18 task–flavor combinations, the highest seed-averaged vector accuracy ranges from .19 to .88 (median .60), compared with zero-shot accuracy of at most .04. Thus, every combination admits an intervention that induces the intended mapping without demonstrations or a textual task specification in the evaluated prompt.

No extractor is uniformly strongest. Method B performs well on singular→plural (.72–.88) and country→capital (.76–.80). Method A is competitive on these tasks but weak on present→past critiques (.01). Method C is strongest on antonym across all three flavors (.48–.50) and on capitalize rules (.67), but substantially weaker on several other combinations. All three methods trail tokens on english→french. These differences motivate reporting the extractors individually rather than treating FBV as a single uniformly reliable procedure.

The Q/A advantage is largest for critiques. The training-selected best vector mean exceeds tokens on five of six critique tasks, four of six explanation tasks, and two of six rule tasks. This count reflects training-pool selection among three extractors; for a single fixed extractor, Method B exceeds tokens on four of six critique tasks, three of six explanation tasks, and one of six rule tasks. The best FBVs also exceed the ten-demonstration FV references on four of six tasks, although these references are not conditioning-count matched. Useful task information can thus be extracted from feedback; the next experiment tests how strongly its advantage over tokens depends on the prompt.

The pattern varies across models. Figure 2 shows every model and task, averaged over flavors and seeds; Appendix Table 9 separates the flavors. For critiques, Method B exceeds tokens in 21 of 42 model–task pairs, and the training-selected best method does so in 28. Averaged over flavors, Method B exceeds tokens in 17 of 42 pairs, Method A in 11, and Method C in 6; Method C is weak except on GPT-J-6B. Feedback vectors and FVs also fail on different tasks: the FV is at most .04 on english→french in six of seven models, where Method B reaches .24–.66, while on capitalize the Gemma FVs reach .50 and 1.00 and no FBV exceeds .30.

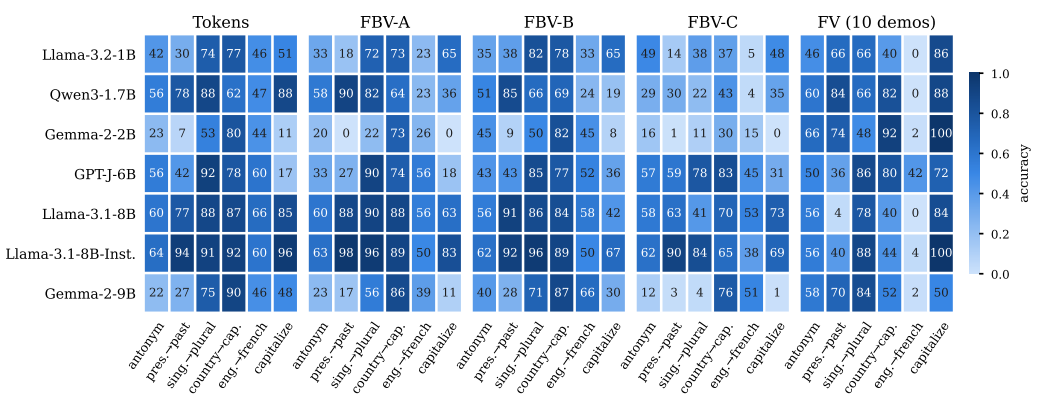


Figure 2: Task execution across seven models under Q/A. Tokens and FBV-A/B/C use five feedback statements, averaged over three flavors and five extraction seeds (50 held-out queries per task). FV is the ten-demonstration reference run, not a conditioning-count-matched control. Cells show accuracy $\times 100$. Per-flavor values appear in Table 9.

Task	Flavor	Tokens	A	B	C	FV	ICL
antonym	rule	.45	.31	.38	**.50**	.46	.60
	expl.	.45	.36	.36	**.48**		
	crit.	.36	.34	.32	**.49**		
present→past	rule	**.41**	.31	.40	.21	.66	.98
	expl.	.42	.23	**.54**	.18		
	crit.	.07	.01	**.19**	.02		
singular→plural	rule	**.94**	.86	.88	.44	.66	.98
	expl.	.79	**.86**	**.86**	.65		
	crit.	.50	.43	**.72**	.06		
country→capital	rule	**.78**	.73	.77	.36	.40	.90
	expl.	**.77**	.72	.76	.49		
	crit.	.78	.72	**.80**	.25		
english→french	rule	**.41**	.15	.24	.04	.00	.72
	expl.	**.49**	.26	.29	.06		
	crit.	**.49**	.26	.45	.06		
capitalize	rule	.48	.57	.55	**.67**	.86	1.00
	expl.	.49	**.67**	.64	.42		
	crit.	.54	.71	**.77**	.34		

Table 1: Task-execution accuracy on Llama-3.2-1B under Q/A. Tokens and A/B/C use five feedback statements; entries average five seeds over 50 fixed queries. FV and ICL are ten-demonstration references, not count-matched controls. Bold marks the best feedback-channel mean. Seed results are in Table 8.

5.2 Sensitivity to prompt format

Prompt formatting is known to substantially affect language-model performance (Sclar et al., 2024; Braun et al., 2025). Table 2 compares critique feedback under Q/A, "x -> y", and `Input:`/`Output:`. For each template, we rerun extraction and reselect the injection layer and scale α on the training pool. Results are averaged across the six tasks, with means and standard deviations over five clean extraction seeds for Q/A and three for each alternative. We report Method B separately from the best result over Methods A/B/C for each task.

The advantage over tokens depends on the template. The Q/A advantage does not persist under stronger token templates: best-FBV margins fall from $+.11$ and $+.12$ under Q/A to approximately zero under the alternatives (Table 2). Method B can also reverse from outperforming to underperforming tokens. Thus, the results support lower sensitivity of the adapted best-of-methods pipeline, not a template-invariant advantage over textual feedback.

Model	Template	Tokens	FBV-B	Best FBV	Best − Tokens
Llama-3.2-1B	Q/A	$.46 \pm .01$	$.54 \pm .01$	$.57 \pm .01$	$+.11$
	x -> y	$.56 \pm .01$	$.47 \pm .02$	$.59 \pm .01$	$+.02$
	Input/Output	$.65 \pm .00$	$.61 \pm .01$	$.65 \pm .00$	$.00$
Llama-3.1-8B	Q/A	$.60 \pm .03$	$.64 \pm .02$	$.72 \pm .03$	$+.12$
	x -> y	$.83 \pm .01$	$.75 \pm .04$	$.80 \pm .03$	$-.02$
	Input/Output	$.84 \pm .01$	$.75 \pm .00$	$.84 \pm .01$	$.00$

Table 2: Critique-flavor accuracy under three prompt templates, averaged over six tasks (mean $\pm$ s.d. over extraction seeds; Q/A: five clean seeds; alternatives: three). Extraction and selection of injection layer and α are repeated under each template. FBV-B denotes Method B. Best FBV uses the A/B/C method selected on the training pool for each task and seed, then averages tasks and seeds. Margins are computed from unrounded means. Rule-flavor results are described in the text.

Rule-flavor sensitivity differs across models. For rules, the Method-B-minus-token margins under Q/A, arrow, and Input/Output are $-.11$, $-.12$, and $-.10$ at 8B, remaining within .01 of the Q/A margin. At 1B the corresponding margins are $-.04$, $-.15$, and $-.05$, indicating a larger change under the arrow template. Each alternative-template rule result uses only one seed; the replicated alternative-template runs evaluate critiques. These rule comparisons are therefore descriptive.

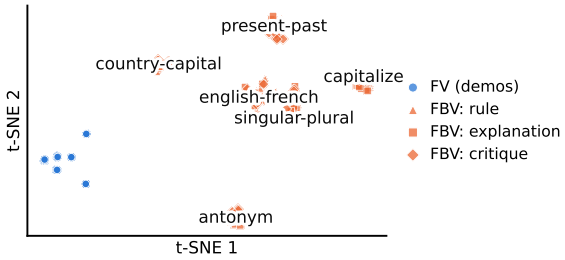
5.3 Comparing demonstration- and feedback-derived representations

Davidson et al. (2025) found that instruction- and demonstration-derived function vectors primarily involve different selected attention heads. We examine this relationship for rules, explanations, and critiques using per-task head sets, and compare cross-modality overlap with cross-task FV overlap and a chance reference. The head analysis concerns Method C; behavioral results obtained with Methods A or B are not directly localized by this analysis.

The selected head sets overlap little. Across the task–flavor comparisons in Table 13, the top-10 FBV and FV head sets have Jaccard overlap 0.00–0.111 (mean .035). The reported random-selection reference is approximately .01, whereas FV head sets from different tasks overlap at a mean of .462. Thus, feedback and demonstrations identify substantially different head sets under the respective causal-mediation procedures. The procedures share the AIE scoring and vector assembly steps, but use different corruptions: shuffled demonstration labels for FVs and another task's feedback for FBVs. The overlap therefore characterizes the heads identified under these interventions. It does not establish that the complete task-execution circuits are separate, and overlap in conditions where Method C fails to induce the task must be interpreted cautiously.

Vector geometry retains task-related structure. For each extraction seed, we obtain one FV from k=10 demonstrations and one Method-C FBV per flavor from k=5 statements. Both constructions use the top-10 AIE heads, with cross-task corruption for FBVs. Seeds resample the conditioning demonstrations or statements; held-out queries stay fixed. Six tasks, four conditioning types, and ten seeds yield 240 vectors per model. All vectors are L_2-normalized. The statistics in Figure 3 use the original vector space: cosine similarity for retrieval and cosine distance for silhouettes. The t-SNE projection uses the cosine metric, PCA initialization, and perplexity 30, and serves only as a visualization. At 1B, it shows separated FBV task groups and a more compact FV region.

Cross-modality retrieval is the more informative task test. The nearest-FV analysis assigns each of the 180 FBVs the task of its most cosine-similar vector among the 60 FVs. Accuracy is .750 at 1B and .844 at 8B, compared with chance .167. By flavor, the 1B/8B accuracies are .68/.88 for critiques, .80/.82 for explanations, and .77/.83 for rules. These results show task-related alignment between demonstration- and feedback-derived vectors. Leave-one-out task 1-NN over all 240 vectors is much higher (.996/.992), but a vector's nearest neighbor is typically from the same task, route, and flavor under another extraction seed. This score therefore largely reflects repeatability within conditions and should not be interpreted as cross-modality generalization. Route 1-NN reaches 1.00; its majority-class reference is .75, because 180 of the 240 vectors are feedback-derived. Qualitative unembedding projections are provided in Appendix G.1.



(a) Vector geometry

	1B	8B
Silhouette, task	.353	.251
Silhouette, route	.281	.171
LOO 1-NN, task	.996	.992
LOO 1-NN, route	1.00	1.00
FBV → nearest FV	.750	.844

(b) Reported statistics

Figure 3: FV and Method-C FBV geometry. (a) t-SNE (Van der Maaten & Hinton, 2008) of L_2-normalized vectors for six tasks and ten extraction seeds on Llama-3.2-1B, using cosine distance, PCA initialization, and perplexity 30. (b) All statistics are computed in the original vector space, not the t-SNE projection. Silhouettes use cosine distance; nearest-neighbor tests use cosine similarity. LOO 1-NN evaluates all 240 vectors and is sensitive to same-condition neighbors across seeds. FBV → nearest FV evaluates task retrieval across modalities (chance .167). Route denotes FV versus FBV; its majority-class reference is .75. The 8B visualization appears in Appendix F.

5.4 Out-of-distribution generalization

Table 3 compares three-seed ID and OOD accuracy on Llama-3.2-1B. We select settings on the ID training pool and freeze them before evaluation. FVs and FBVs both use ten conditioning items and search the same layer–scale grid. The splits test rare-word antonyms, irregular past tenses, and rare-country capital retrieval (Appendix D). Both Methods B and C are reported.

Irregular past tenses produce a crossover at 1B. With demonstrations drawn only from regular verbs, the FV drops from .97 on regular ID verbs to .63 on irregular OOD verbs, while Method B rises from .33 to .76; tokens rise from .41 to .72. Because the same 35 irregular verbs are evaluated across seeds, replication measures extraction variability rather than independent test-set uncertainty, and we do not claim statistical significance. The crossover is consistent with sensitivity to demonstration regularities, but does not show that FV errors apply an incorrect *-ed* rule. Method C does not reproduce the crossover (.07 → .22), and at 8B the effect disappears near ceiling. Thus, the evidence supports a specific transfer difference, not a general advantage on structural shifts.

Frequency shifts depend on the task and model. For antonym, Method C changes from .49 to .47, compared with .48 to .38 for the FV; Method B is weaker on both splits (.31 to .21). For country→capital, the FV changes from .76 to .67, while Method B falls from .83 to .31 and Method C from .66 to .29. Tokens also decline substantially, from .94 to .53. The rare-country ordering is not universal across models. At Llama-3.1-8B, the three-seed ID–OOD loss is .09 for Method B versus .03 for the FV. Method C has zero accuracy on both splits there, so its zero gap reflects a floor. Gemma-2-9B reverses the gap ordering: Method B loses .11 and the FV .31, using three seeds. We therefore do not claim that knowledge shifts consistently favor demonstration vectors. These experiments establish task- and method-dependent transfer profiles. Task identity and shift type are confounded, and test sets are small. The rule-versus-sample-regularity interpretation is a hypothesis supported by the past-tense example, not an identified general mechanism. Additional OOD details appear in Appendix H; conditioning-count sweeps appear in Appendix G.2.

5.5 When does feedback yield an effective vector?

Across 126 main-suite conditions (seven models, six tasks, three flavors; five seeds) and 20 single-seed breadth conditions, token accuracy and best-FBV accuracy correlate at Pearson $r = .86$. On the main suite alone, $r = .86$; removing model means gives .81, and removing both model and task effects gives .74. The association is also present for each extractor separately: pooled $r = .89$, .76, and .58 for A, B, and C. All six conditions with token accuracy below .10 have vector accuracy at most .19 for every method. These results link vector effectiveness to the model's ability to use feedback in the tested format, but do not establish a comprehension bound or a label-free diagnostic. Full aggregation definitions and robustness analyses appear in Appendix G.3.

Task (shift)	ICL	Tokens	FV	FBV-B	FBV-C
antonym (rare)	.56 → .69	.41 → .34	.48 → .38	.31 → .21	.49 → .47
present→past (irregular)	1.00 → .86	.41 → .72	.97 → .63	.33 → .76	.07 → .22
country→capital (rare)	.95 → .85	.94 → .53	.76 → .67	.83 → .31	.66 → .29

Table 3: ID → OOD accuracy on Llama-3.2-1B (three-seed means, rule feedback). Layer and scale are selected on the ID training pool. FV and FBVs use ten conditioning items; Tokens uses five statements and ICL uses ten demonstrations. All baseline columns come from the Method-C run files. Frequency OOD sets contain 50 queries; irregular verbs number 35. Both FBV methods are shown without OOD-based method selection.

Feedback grounded in observed errors. We also test feedback written in response to errors the models actually made (Appendix C). After each zero-shot error, an external model (gpt-oss-120b; OpenAI, 2025) writes one sentence about that specific wrong answer, without revealing the correct answer or giving an example. The sentence is kept only if, as text, it makes the model correct its answer. We extract a Method-B vector from this single sentence in the revision context (the input and the model's own wrong answer), then inject it with the sentence removed. Feedback was verified for 833 of 2,088 test errors (40%). The rate is again lowest for models that use authored feedback poorly as text (9% on Gemma-2-2B, 84% on Llama-3.1-8B-Instruct). Over 822 errors in 33 evaluable cells, the vector corrects $.708$ of errors, compared with $.007$ without feedback and $.999$ with the text. We also compare vectors from accepted and rejected sentences on the 258 errors that have a logged rejected candidate. The accepted sentence's vector corrects $.558$ of these errors, compared with $.264$ for the rejected one (McNemar $p = 2 \times 10^{-19}$). A vector from feedback about a *different* error type of the same task still corrects $.575$, compared with $.636$ for matched feedback. Critiques of different errors thus share largely task-level content.

6 Limitations

Our experiments focus on short, objectively scored task mappings, so the results do not yet establish that feedback vectors can steer open-ended generation or more complex reasoning behavior. While Section 5.5 extends the analysis to feedback generated in response to actual model errors, those experiments evaluate feedback that is first verified to correct the model when presented as text; performance therefore characterizes the compressibility of effective feedback rather than arbitrary feedback. Several analyses also have deliberately narrower scope. The prompt-format experiment adapts extraction and hyperparameters separately to each format, so it measures robustness of the extraction pipeline rather than transfer of a single frozen vector. Mechanistic comparisons of attention heads concern Method C specifically and should not be interpreted as identifying the full circuits responsible for feedback-based task execution.

7 Conclusion

Rules, explanations, and critiques can condition vectors that induce simple tasks without a textual task specification at evaluation. Their usefulness depends on the extractor and comparison protocol: the advantage over textual critiques under Q/A largely disappears with better-performing templates, even though the best-of-methods vector pipeline changes less. Method-C feedback vectors use largely different selected heads from FVs while retaining cross-source task alignment, extending prior results on instructions. Transfer behavior also differs in specific settings, most clearly on irregular past tenses at 1B, but does not consistently favor feedback. Verified critiques of real model errors also compress into vectors that correct most of those errors. These findings make conditioning source, extraction procedure, and prompt format central variables in evaluating activation-based task steering.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, concision, organization, and to provide feedback on experimental methodology and interpretation of results. All AI-assisted suggestions were independently reviewed and revised by the authors.

REFERENCES

Afra Feyza Akyürek, Ekin Akyürek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7716–7733, 2023.

Adar Avsian, Christopher Richardson, Anirudh Sundar, and Larry Heck. Flex: Language modeling with few-shot language explanations. *arXiv preprint arXiv:2601.04157*, 2026.

Aleksandra Bakalova, Yana Veitsman, Xinting Huang, and Michael Hahn. Contextualize-then-aggregate: Circuits for in-context learning in gemma-2 2b. *arXiv preprint arXiv:2504.00132*, 2025.

Beatriz Borges, Niket Tandon, Tanja Käser, and Antoine Bosselut. Let me teach you: Pedagogical foundations of feedback for language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12082–12104, 2024.

Joschka Braun, Carsten Eickhoff, David Krueger, Seyed Ali Bahrainian, and Dmitrii Krasheninnikov. Understanding (un) reliability of steering vectors in language models. *arXiv preprint arXiv:2505.22637*, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Guy Davidson, Todd M. Gureckis, Brenden M. Lake, and Adina Williams. Do different prompting methods yield a common task representation in language models? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.12075.

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

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Jihoon Kwon, Jiwon Choi, and Jy-yong Sohn. Distributional alignment as a criterion for designing task vectors in in-context learning. *arXiv preprint arXiv:2605.20730*, 2026.

Andrew K. Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane X. Wang, and Felix Hill. Can language models learn from explanations in context? *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: Making in-context learning more effective and controllable through latent space steering. In *International Conference on Machine Learning (ICML)*, 2024.

Llama Team, AI @ Meta. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Renjie Luo, Zichen Liu, Xiangyan Liu, Chao Du, Min Lin, Wenhu Chen, Wei Lu, and Tianyu Pang. Language models can learn from verbal feedback without scalar rewards. *arXiv preprint arXiv:2509.22638*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Hoang Phan, Quang H. Nguyen, Hung T. Q. Le, Xiusi Chen, Heng Ji, and Khoa D. Doan. Decoding the critique mechanism in large reasoning models. *arXiv preprint arXiv:2603.16331*, 2026.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Baturay Saglam, Xinyang Hu, Zhuoran Yang, Dionysis Kalogerias, and Amin Karbasi. Learning task representations from in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6634–6663, 2025.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pp. 25055–25083, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Robyn Speer. rspeer/wordfreq: v3.0, September 2022. URL https://doi.org/10.5281/zenodo.7199437.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. In *International Conference on Learning Representations (ICLR)*, 2025.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. *URL https://arxiv. org/abs/2205.05124*, 1, 2022.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model, 2021.

Entang Wang, Yiwei Wang, Aleksandra Bakalova, and Michael Hahn. How few-shot examples add up: A causal decomposition of function vectors in in-context learning. *arXiv preprint arXiv:2605.16591*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Haolin Yang, Hakaze Cho, Kaize Ding, and Naoya Inoue. Task vectors, learned not extracted: Performance gains and mechanistic insight. *arXiv preprint arXiv:2509.24169*, 2025b.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Model inventory and experiment coverage

Table 4 gives exact checkpoints. All seven models have function-vector reference runs and five-seed task-execution results. Head-overlap experiments cover Llama-3.2-1B, Qwen3-1.7B, and Llama-3.1-8B with five seeds each, and GPT-J-6B and Gemma-2-9B with three seeds each. The ten-seed geometry analyses cover the two base Llama models. Template experiments also cover those two models: Q/A uses five seeds, alternate-template critiques three, and alternate-template rules one.

Model	Checkpoint	Type
Llama-3.2-1B	`meta-llama/Llama-3.2-1B`	Base
Qwen3-1.7B	`Qwen/Qwen3-1.7B-Base`	Base
Gemma-2-2B	`google/gemma-2-2b`	Base
GPT-J-6B	`EleutherAI/gpt-j-6b`	Base
Llama-3.1-8B	`meta-llama/Llama-3.1-8B`	Base
Llama-3.1-8B-Instruct	`meta-llama/Llama-3.1-8B-Instruct`	Instruct
Gemma-2-9B	`google/gemma-2-9b`	Base

Table 4: Models in the main five-seed task suite. Six checkpoints are base models and one is instruction-tuned.

OOD and auxiliary analyses. The OOD experiments use three seeds for both Methods B and C on the two base Llama models. Gemma-2-9B has three seeds for Method B and one for Method C. The k-sweep figures use seed 0, Method C, and rule feedback at all three model sizes. Composition probes cover the two base Llama models. Paraphrase, negation, misleading-feedback, and third-person probes use Llama-3.2-1B. The word-shuffle corruption ablation also uses Llama-3.2-1B; the attention-normalization assembly check uses Gemma-2-9B.

Task-breadth extension. The 20 additional conditions all use Llama-3.1-8B and a single extraction seed. Synonym, english→german, english→spanish, and capitalize-first-letter each have all three feedback flavors (12 cells). Landmark→country, lowercase-first-letter, person→occupation, and product→company each have rules and explanations (eight cells). These runs broaden the correlation analysis but are not five-seed replications of the main suite.

B The feedback dataset

Each task has ten authored paraphrases per feedback flavor. Statements describe the mapping without giving input–output pairs. We aim to preserve the intended task information across flavors while varying its pragmatic form; their wording and length are not identical. Explanations justify a rule rather than explain a particular answer. Authored critiques presuppose an error without referring to an actual sampled model response; critiques of real errors are described in Appendix C. The following are examples from the antonym sets, not the complete dataset.

Rules. "Output the antonym of the input word." "For each word given, respond with a word that means the opposite." "Reply with the word whose meaning is the reverse of the prompt word."

Explanations. "The correct answer is the word with the opposite meaning, because this task pairs each word with its antonym." "Each answer negates the meaning of the prompt word, since the mapping here is word to opposite."

Critiques. "That's not right — you repeated the word back; the answer should be a word meaning the opposite." "Wrong: you gave a synonym, but the task wants the antonym."

For controls, misleading sets state an incorrect mapping, such as synonym instead of antonym. Negation sets express the correct mapping through a negative instruction, such as "Do not output a synonym; output the word with the opposite meaning." These controls probe the effect of changing the description; their outcomes appear in Appendix G.4.

C FEEDBACK GROUNDED IN OBSERVED ERRORS: DETAILS

Pipeline. (1) Each model answers the zero-shot prompt. Test errors come from the 50 held-out queries; development errors come from the extraction pool and are used only to select layer and scale. (2) The generator writes one sentence about the specific wrong answer w. (3) A candidate f is accepted only if the model, shown f after its wrong answer, produces the correct answer; the generator gets up to five attempts. (4) Method B is applied to the single accepted sentence: the revision prompt with the feedback line is contrasted with the same prompt without it, averaged over 24 revision contexts (development inputs paired with the model's own wrong answers). The contexts are shared by every item in a model–task cell, so two items' vectors differ only in feedback text, and no extraction prompt contains a reference answer. (5) The vector is injected into the revision prompt with the feedback line removed. Nine of 42 cells have fewer than five verified test errors and are excluded, which leaves 822 errors in 33 cells (Table 5).

Feedback source	Extract	Inject	n	Accuracy [95% CI]
None (revision cue only)	–	rev.	822	.007 [.003, .016]
Verified feedback as text	–	rev.	822	.999 [.993, 1.00]
Verified, grounded	pre	rev.	822	.491 [.457, .526]
Verified, grounded	rev.	rev.	822	**.708** [.676, .738]
Verified, grounded	rev.	pre	822	.625 [.592, .658]
Authored critiques (k=5)	pre	pre	822	.607 [.573, .640]
Authored critiques (k=5)	rev.	rev.	822	.629 [.595, .661]
Verified, grounded (paired)	rev.	rev.	258	**.558** [.497, .617]
Rejected candidate, same error	rev.	rev.	258	.264 [.214, .321]
Matched: this error's feedback	rev.	rev.	478	**.636** [.592, .678]
Mismatched: other error type	rev.	rev.	478	.575 [.531, .619]

Table 5: Feedback vectors from critiques of real model errors, pooled over 33 model–task cells (seven models). "Pre" is the pre-answer prompt "Q: x A:"; "rev." is the revision prompt containing the input and the model's own wrong answer. Grounded vectors use one Method-B sentence. Authored vectors use five of the paper's critique statements. Layer and scale are selected on development errors for each row. Intervals are Wilson binomial intervals from a single seed.

Additional findings. The vector is more effective at the revision point than before the attempt (.708 vs. .625). In the verified-vs.-rejected comparison, the two vectors disagree on 84 items, 80 of which favor the verified vector, and the verified vector wins in 20 of 25 cells. Rejected candidates have the same source and style, concern the same error, and reached the model but failed to correct it. The matched condition tests each vector on the input that produced it; the mismatched condition is the test on new inputs. Matched feedback is better in 12 of 17 cells, worse in 2, and tied in 3. That contrast covers only the 17 cells with at least two recurring error types (478 instances), because echoing and defining dominate zero-shot errors. Under the same context and injection site, grounded feedback exceeds five authored critiques (.708 vs. .629). Country→capital is the exception to revision-point gains (.27 at revision vs. .59 before the attempt, Table 7). Once a wrong answer is in context, a last-token edit rarely changes a retrieved fact.

Prompts. Initial answers use the paper's zero-shot prompt, "Q: $x\backslash$nA:". Verification, the with-feedback side of extraction, and the text condition use the first prompt below. The no-feedback baseline, the other side of extraction, and vector injection use the second prompt, which is identical except that the feedback line is removed:

```
Q: {x}                          Q: {x}
A: {wrong}                      A: {wrong}

Feedback: {f}                   Try again.
Try again.                      Revised A:
Revised A:
```

The feedback generator (gpt-oss-120b, temperature 0, low reasoning effort) receives the system prompt "You are a terse tutor. Reply with ONE sentence of corrective feedback. Never state the correct answer." Its user prompt contains the zero-shot prompt, the task rule, the model's answer, and the reference answer, which is marked secret. It asks for one sentence "that tells the model what it did wrong and what kind of answer is required, WITHOUT revealing the reference answer and WITHOUT giving any example input-output pair." On retries, the prompt also lists each rejected candidate and the reason it was rejected. Before any candidate reaches the evaluated model, a filter rejects whole-word occurrences of the reference answer (case-sensitive for casing tasks) and literal demonstration forms ("→", "Q:"/"A:"). This filter rejected 14 candidates across the run, and no accepted sentence contains the reference answer.

Example. Gemma-2-2B, singular→plural. For input "cup", the model answers "a small container for holding liquid". Two candidates ("You gave a definition instead of converting the given noun into its required plural form.") fail verification, and the model again produces a definition. The accepted sentence is "Your reply provided a definition, but the task requires you to output the noun's plural form instead." With it, the model answers "cups". Without it, the model repeats its definition.

Protocol. Decoding is greedy, uses at most eight new tokens, and cuts at the first newline. The evaluator is the same as in the main experiments. Layer and $\alpha \in \{1, 2, 4\}$ are selected separately for each condition by first-token accuracy on verified *development* errors. These errors come from up to 110 extraction-pool inputs and never include test queries. The paired verified-vs.-rejected comparison reuses the layer and scale of the verified revision-context vector. For each error, the rejected candidate is the first one that reached the model and failed to correct it; candidates removed by the leak filter are excluded. Error-type labels come from a per-task taxonomy of at most six labels, proposed by the generator from the observed errors. The most frequent labels are echoed input (365), definition instead of answer (185), and description instead of capital (92). Verification and final evaluation disagree on one of 822 items. The cause is batch-dependent nondeterminism in half-precision greedy decoding, and we report the item as is.

Order of experiments. We first ran the pipeline with vectors extracted from the pre-answer prompt, as in the main experiments. That version already showed compression (revision-point accuracy .491), but pre-answer injection scored higher (.644) and matched feedback did not beat mismatched feedback (.439 vs. .464). Its extraction context matched the pre-answer site and not the revision site. The revision-context extraction in Table 5 was specified in a plan frozen before it was run, after the first pass exposed this mismatch. It reverses both results; the first-pass numbers remain as the "pre/rev." row.

Geometry. Cosines are computed for each item and then averaged, at each cell's injection layer. Pre-answer grounded vectors have cosine .81 with the authored-critique vector and .71 with the authored-rule vector. Revision-context grounded vectors have cosine .39 and .43 with the same two vectors. The two extraction contexts of the same sentence have cosine .44. Verified and rejected revision-context vectors have cosine .85 and nearly equal norms (ratio 1.06). The component of the verified vector orthogonal to the rejected vector has norm .54 relative to the full verified vector.

Model	Errors	Verified	Rate	Cells	Rev.-ctx vector @ rev.
Llama-3.2-1B	298	126	.42	6	.71
Qwen3-1.7B	299	92	.31	6	.50
Gemma-2-2B	300	28	.09	2	.27
GPT-J-6B	294	32	.11	2	.84
Llama-3.1-8B	299	144	.48	6	.73
Gemma-2-9B	300	162	.54	5	.71
Llama-3.1-8B-Instruct	298	249	.84	6	.69
All	2088	833	.40	33	.65

Table 6: Verification and vector accuracy by model. "Errors" and "Verified" count held-out test errors. The last column gives the unweighted mean over evaluated cells of revision-context vectors injected at the revision point; the pooled value in Table 5 is .708. Cells with fewer than five verified test errors are excluded: Gemma-2-2B (4 tasks), GPT-J-6B (4), and Gemma-2-9B (capitalize).

			Grounded, rev. ctx		Authored crit.	
Task	Cells	Pre-ctx @ rev.	@ rev.	@ pre	pre @ pre	rev. @ rev.
antonym	5	.54	.73	.40	.68	.71
present→past	5	.62	.98	.70	.62	.77
singular→plural	7	.55	.71	.60	.62	.72
country→capital	6	.10	.27	.59	.82	.25
english→french	5	.31	.46	.40	.50	.36
capitalize	5	.59	.82	.46	.44	.68
All	33	.45	.65	.53	.62	.58

Table 7: Unweighted per-task means over evaluated cells. No-feedback revision accuracy is at most .02, and verified feedback as text is at least .99, for every task.

D Out-of-distribution splits

For antonym, english→french, and country→capital, inputs are ranked by wordfreq Zipf frequency (Speer, 2022). The 50 rarest inputs form the OOD test set; the extraction/training pool and ID test are disjoint subsets drawn from the most frequent half of the remaining inputs. Examples include *assail*, *transgress*, and *disclaim*, and countries such as Eswatini, Tuvalu, and Comoros. Input frequency provides a reproducible distribution shift, but it is not a direct measurement of a model's factual knowledge or of the frequency of the corresponding answer.

For present→past, the training pool and ID test contain regular verbs, including regular spelling changes such as *y→ied* and consonant doubling. The OOD set contains 35 irregular verbs, such as *hide→hid* and *speak→spoke*. Demonstrations therefore exhibit regular morphology while the OOD test requires irregular forms. The feedback instruction to convert to past tense remains valid on both splits. All injection settings are selected on the ID training pool and frozen before test evaluation.

E Corruption design for causal mediation

An AIE corruption should disrupt the task-conditioning signal sufficiently to reveal the effect of restoring clean head outputs. For FVs, we shuffle demonstration labels. For FBVs, we replace the prefix with an unrelated task's feedback, matched in flavor and statement count. This changes task content without equating every surface property of the two prefixes.

In an initial diagnostic, shuffling words within a rule was ineffective: task-identifying words remained in the prompt, and the clean-versus-corrupt contrast produced weak AIE scores. On singular→plural, changing to cross-task corruption raised Method-C accuracy from .06 to .78 in that diagnostic run. This is an individual ablation result, not the five-seed main-table estimate. All main-text Method-C results use cross-task corruption. The ablation motivates the corruption choice but does not eliminate corruption as a factor in FV–FBV head differences.

F ADDITIONAL TASK-EXECUTION RESULTS

F.1 FIVE-SEED RESULTS FOR THE PRIMARY MODEL

Table 8 reports every task-execution cell for seeds 0–4. An earlier seed-2 run was invalidated after numerical corruption under MPS memory pressure: all conditions, including the token baseline, flatlined at zero. That file is excluded and a clean seed-2 rerun is included. A flatline canary now guards queued runs. Method C on antonym-rule ranges from .46 to .54. The less stable cells include Method C on singular→plural rule (.14–.66) and Method B on capitalize-rule, where seed 0 is .24 and the other four seeds range from .58 to .68. Across Table 1, population seed standard deviations are at most .05 for tokens, .09 for Method A, and .05 for Method B except capitalize-rule (.16); Method C reaches .18.

Table 8: Per-seed Q/A accuracies on Llama-3.2-1B with five feedback statements. Each row evaluates the same 50 held-out queries. T denotes feedback as tokens.

Task	Flavor	Method	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4
antonym	rule	T	.46	.42	.42	.46	.48
		A	.30	.32	.32	.32	.28
		B	.40	.34	.38	.40	.38
		C	.52	.52	.48	.46	.54
antonym	expl.	T	.38	.46	.46	.46	.50
		A	.34	.36	.38	.38	.34
		B	.32	.36	.32	.40	.38
		C	.50	.48	.44	.48	.50
antonym	crit.	T	.28	.34	.44	.36	.38
		A	.36	.30	.34	.32	.36
		B	.32	.30	.32	.32	.36
		C	.48	.50	.48	.50	.50
present→past	rule	T	.44	.36	.48	.36	.42
		A	.22	.44	.32	.38	.20
		B	.34	.42	.38	.46	.40
		C	.08	.20	.28	.24	.26
present→past	expl.	T	.40	.42	.38	.44	.44
		A	.32	.16	.22	.26	.18
		B	.50	.58	.56	.52	.52
		C	.02	.16	.18	.34	.22
present→past	crit.	T	.08	.08	.06	.08	.04
		A	.00	.02	.02	.00	.00
		B	.20	.18	.18	.20	.20
		C	.00	.02	.02	.06	.02
singular→plural	rule	T	.92	.92	.94	.96	.94
		A	.86	.86	.86	.86	.88
		B	.86	.90	.86	.90	.86
		C	.40	.40	.14	.66	.60
singular→plural	expl.	T	.78	.80	.76	.82	.80
		A	.88	.84	.88	.84	.86
		B	.88	.84	.86	.90	.80
		C	.62	.40	.66	.86	.72
singular→plural	crit.	T	.46	.48	.56	.52	.48
		A	.40	.28	.52	.42	.54
		B	.76	.66	.74	.72	.72
		C	.06	.06	.04	.06	.06
country→capital	rule	T	.78	.76	.80	.76	.78
		A	.74	.74	.74	.72	.72
		B	.78	.78	.78	.74	.78
		C	.20	.40	.52	.42	.28
country→capital	expl.	T	.76	.70	.84	.76	.78
		A	.78	.70	.72	.64	.76

Continued on next page

Table 8 continued

Task	Flavor	Method	Seed 0	Seed 1	Seed 2	Seed 3	Seed 4
		B	.78	.72	.76	.76	.78
		C	.56	.60	.34	.50	.46
country→capital	crit.	T	.78	.78	.80	.78	.76
		A	.76	.70	.72	.72	.72
		B	.82	.80	.80	.80	.78
		C	.32	.40	.00	.20	.32
english→french	rule	T	.40	.38	.42	.40	.44
		A	.16	.12	.14	.12	.22
		B	.28	.24	.16	.20	.30
		C	.00	.00	.22	.00	.00
english→french	expl.	T	.48	.48	.48	.48	.52
		A	.26	.24	.30	.20	.32
		B	.26	.38	.26	.26	.28
		C	.00	.00	.16	.02	.12
english→french	crit.	T	.50	.54	.44	.50	.48
		A	.26	.36	.22	.16	.30
		B	.44	.50	.48	.42	.42
		C	.00	.12	.16	.00	.00
capitalize	rule	T	.46	.50	.56	.46	.44
		A	.48	.60	.60	.62	.54
		B	.24	.60	.58	.66	.68
		C	.72	.58	.72	.66	.68
capitalize	expl.	T	.44	.44	.56	.52	.48
		A	.68	.64	.70	.68	.66
		B	.60	.64	.68	.66	.62
		C	.36	.38	.52	.50	.36
capitalize	crit.	T	.60	.54	.50	.52	.56
		A	.76	.70	.66	.68	.76
		B	.74	.76	.76	.82	.76
		C	.18	.48	.46	.34	.22

F.2 SEVEN-MODEL TASK-EXECUTION GRID

Table 9 reports every main-suite cell summarized in Figure 2 and used in the correlation analysis (Appendix G.3): seven models, six tasks, three flavors, and four feedback channels, each averaged over five extraction seeds. The Llama-3.2-1B rows reproduce Table 1.

Table 9: Q/A task-execution accuracy for all seven models (means over five extraction seeds; five feedback statements; 50 fixed held-out queries per task). T denotes feedback as tokens.

		Rule				Explanation				Critique			
Model	Task	T	A	B	C	T	A	B	C	T	A	B	C
Llama-3.2-1B	antonym	.45	.31	.38	.50	.45	.36	.36	.48	.36	.34	.32	.49
	present→past	.41	.31	.40	.21	.42	.23	.54	.18	.07	.01	.19	.02
	singular→plural	.94	.86	.88	.44	.79	.86	.86	.65	.50	.43	.72	.06
	country→capital	.78	.73	.77	.36	.77	.72	.76	.49	.78	.72	.80	.25
	english→french	.41	.15	.24	.04	.49	.26	.29	.06	.49	.26	.45	.06
	capitalize	.48	.57	.55	.67	.49	.67	.64	.42	.54	.71	.77	.34
Qwen3-1.7B	antonym	.58	.55	.47	.24	.53	.60	.52	.34	.56	.59	.54	.29
	present→past	.86	.95	.80	.39	.74	.86	.90	.27	.73	.90	.86	.23
	singular→plural	.96	.84	.66	.28	.86	.84	.64	.22	.82	.76	.67	.16
	country→capital	.53	.64	.66	.42	.60	.54	.68	.38	.74	.73	.74	.50
	english→french	.49	.25	.23	.06	.45	.22	.22	.00	.48	.23	.26	.06
	capitalize	1.00	.34	.13	.41	.82	.42	.21	.35	.82	.32	.23	.30
Gemma-2-2B	antonym	.46	.51	.60	.20	.10	.04	.33	.18	.12	.07	.40	.10

Continued on next page

Table 9 continued

Model	Task	Rule				Explanation				Critique			
		T	A	B	C	T	A	B	C	T	A	B	C
	present→past	.19	.00	.28	.02	.03	.00	.00	.00	.00	.00	.00	.00
	singular→plural	.85	.46	.78	.26	.30	.06	.28	.01	.45	.13	.43	.05
	country→capital	.90	.82	.83	.46	.73	.69	.82	.22	.77	.67	.82	.21
	english→french	.59	.42	.47	.30	.22	.07	.33	.03	.51	.27	.54	.12
	capitalize	.25	.00	.06	.00	.07	.00	.17	.00	.02	.00	.03	.00
GPT-J-6B	antonym	.57	.44	.43	.58	.60	.32	.47	.57	.51	.22	.38	.56
	present→past	.41	.29	.26	.61	.56	.41	.67	.55	.28	.12	.38	.62
	singular→plural	.91	.92	.84	.86	.94	.88	.85	.78	.92	.90	.87	.69
	country→capital	.80	.74	.78	.81	.78	.74	.78	.85	.76	.74	.76	.84
	english→french	.65	.56	.52	.49	.58	.59	.48	.44	.58	.55	.56	.41
	capitalize	.22	.20	.38	.36	.12	.17	.28	.27	.18	.16	.42	.30
Llama-3.1-8B	antonym	.63	.60	.59	.62	.62	.61	.59	.62	.55	.59	.49	.49
	present→past	.95	1.00	.95	.58	.94	1.00	.96	.90	.41	.63	.83	.42
	singular→plural	.98	.94	.92	.68	.97	.92	.92	.47	.70	.83	.73	.07
	country→capital	.86	.88	.82	.62	.86	.88	.89	.70	.87	.88	.80	.79
	english→french	.71	.58	.50	.43	.76	.61	.55	.54	.49	.48	.69	.61
	capitalize	1.00	.82	.67	.91	.99	.72	.28	.91	.55	.36	.32	.36
Llama-3.1-8B-Inst.	antonym	.65	.63	.63	.62	.65	.62	.61	.63	.63	.63	.64	.63
	present→past	.98	.98	.93	.94	.99	1.00	.95	.89	.85	.97	.87	.88
	singular→plural	.98	.96	.96	.80	.98	.96	.96	.88	.78	.96	.96	.84
	country→capital	.93	.88	.88	.69	.91	.90	.90	.58	.92	.90	.90	.69
	english→french	.61	.52	.52	.08	.74	.54	.58	.38	.46	.43	.39	.69
	capitalize	.98	.67	.64	.87	.99	.88	.65	.79	.90	.93	.72	.42
Gemma-2-9B	antonym	.39	.48	.57	.23	.12	.09	.29	.04	.14	.12	.35	.09
	present→past	.50	.51	.48	.06	.24	.00	.30	.02	.07	.00	.06	.00
	singular→plural	.93	.72	.94	.09	.74	.61	.65	.02	.59	.36	.54	.02
	country→capital	.88	.89	.88	.56	.90	.84	.88	.87	.91	.84	.86	.86
	english→french	.62	.64	.70	.64	.24	.14	.67	.39	.51	.37	.60	.49
	capitalize	.79	.24	.57	.02	.38	.06	.15	.01	.27	.02	.18	.00

F.3 METHOD D: OPTIMIZED READOUT

Method D was run once (seed 0) for rule feedback on Llama-3.2-1B and Llama-3.1-8B, with two arms. The replacement arm starts from the Method-A readout and the additive arm from the α-scaled Method-B vector. Each arm keeps the injection layer selected for its initialization and optimizes only the injected vector; model weights stay frozen. Each task uses 70 labeled training-pool examples: 50 probe examples for the loss and 20 dev examples for checkpoint selection. Optimization uses Adam for 150 steps with batch size 16 and learning rate $0.005\,\|\mathbf{v}_0\|$, minimizing cross-entropy on the first gold-answer token. The checkpoint with the highest dev first-token accuracy, evaluated every 15 steps, is kept. The model is loaded in bfloat16. Initialization accuracies come from this run and can differ slightly from the five-seed task-execution values.

Table 10 shows that optimization helps most where the training-free readout is weak. At 1B, mean accuracy rises from .45 to .68 in the replacement arm and from .47 to .62 in the additive arm, largely because present→past rises from .20 to .98 and from .32 to .94. At 8B, the additive arm rises from .75 to .79, but the replacement arm falls from .80 to .74. It decreases on antonym, country→capital, and capitalize, despite dev accuracy at or above initialization. Method D is therefore a supervised reference, not an upper bound on training-free extraction.

F.4 FUNCTION-VECTOR REFERENCE RESULTS

Table 11 reports FV accuracy at 44–86% of ten-shot ICL accuracy on five of six tasks. For translation, the FV remains at zero despite nonzero TV and ICL accuracy. This is a failure of the evaluated FV construction, not evidence that translation cannot be represented by any task vector.

		Replacement arm		Additive arm	
Model	Task	A (init)	D	B (init)	D
Llama-3.2-1B	antonym	.26	.46	.38	.38
	present→past	.20	.98	.32	.94
	singular→plural	.86	.88	.86	.94
	country→capital	.74	.72	.78	.80
	english→french	.16	.36	.26	.22
	capitalize	.48	.68	.24	.44
	mean	.45	.68	.47	.62
Llama-3.1-8B	antonym	.60	.46	.58	.60
	present→past	1.00	1.00	.96	1.00
	singular→plural	.94	.98	.92	.96
	country→capital	.86	.78	.90	.88
	english→french	.52	.56	.44	.54
	capitalize	.88	.68	.68	.78
	mean	.80	.74	.75	.79

Table 10: Method D on rule feedback (one seed; 50 held-out queries per task). Each arm reports held-out accuracy of its initialization and of the optimized vector. Labeled budget per task: 50 probe and 20 dev training-pool examples.

Task	Zero-shot	ICL (k=10)	FV	TV	FV / ICL
antonym	.00	.60	.46	.48	.77
present→past	.00	.98	.66	.94	.67
singular→plural	.04	.98	.66	.80	.67
country→capital	.00	.90	.40	.56	.44
english→french	.00	.72	.00	.22	.00
capitalize	.00	1.00	.86	.66	.86

Table 11: Function-vector reproduction on Llama-3.2-1B: exact-match accuracy on 50 held-out queries. FV: top-10 AIE heads, additive injection with training-selected settings; TV: training-selected state replacement. FV and ICL use ten demonstrations. These are reference-run values, not five-seed task-execution means.

F.5 Cross-model reference comparison

Table 12 retains the original Gemma-2-2B reference run and updates the primary-model column to five seeds. In this Gemma run, the best FBV exceeds tokens on antonym, present→past, and singular→plural; it trails tokens on country→capital and capitalize. The main-suite correlation analysis separately uses five-seed results for all seven models. These two summaries have different scopes.

The Gemma attention-normalization assembly check did not revive the collapsed Method-C conditions when the vectors were assembled without the normalization approximation. This excludes that tested modification as a remedy; it does not establish a complete explanation of the failures.

F.6 Head-overlap detail

The reference head sets in Table 13 have mean cross-source Jaccard overlap .035, versus approximately .01 for random selection and .462 for FV–FV overlap across tasks. The attribution interventions differ between sources, as specified in Section 3.1.

	Llama-3.2-1B			Gemma-2-2B		
Task	Tokens	FBV	FV	Tokens	FBV	FV
antonym	.45	**.50**	.46	.36	**.58**	.66
present→past	.42	**.54**	.66	.20	**.24**	.74
singular→plural	**.94**	.88	.66	.88	**.96**	.48
country→capital	.78	**.80**	.40	**.88**	.82	.92
english→french	**.49**	.45	.00	**.56**	**.56**	.02
capitalize	.54	**.77**	.86	**.30**	.16	1.00

Table 12: Descriptive cross-model summary: maximum over flavors for Tokens and over flavors/methods for FBV; FV reference runs. Llama entries maximize five-seed means. Gemma entries retain the single-seed reference run and are not the five-seed Gemma aggregate. Bold compares Tokens with FBV, including displayed ties.

Task	Flavor	Jaccard@10	AIE Spearman
antonym	criti.	0.053	0.125
antonym	expla.	0.053	0.199
antonym	rule.	0.053	0.102
capitalize	criti.	0.053	0.207
capitalize	expla.	0.053	0.212
capitalize	rule.	0.053	0.223
country–capital	criti.	0.111	0.024
country–capital	expla.	0.053	0.118
country–capital	rule.	0.000	0.067
english–french	criti.	0.000	0.047
english–french	expla.	0.000	-0.025
english–french	rule.	0.000	0.001
present–past	criti.	0.000	0.054
present–past	expla.	0.053	-0.009
present–past	rule.	0.000	-0.011
singular–plural	criti.	0.053	0.073
singular–plural	expla.	0.000	0.031
singular–plural	rule.	0.053	0.027

Table 13: Llama-3.2-1B task–flavor head-overlap detail from the reference extraction. These entries are not averages over the multi-seed inventory. Jaccard compares the top ten AIE heads; Spearman compares full AIE maps.

G Additional feedback-vector analyses

G.1 Qualitative vocabulary projections

Table 14 projects FVs and rule-conditioned Method-C FBVs through the unembedding. Several FBV projections emphasize words associated with the task description: *reversed* for antonym, *French* for translation, and *capitalized* for capitalize. The corresponding FV projections differ and are not uniformly interpretable. For example, the plural FV surfaces *airs*, *erals*, and *aries*, while other FV token lists have no obvious connection to the intended outputs.

These observations are compatible with differences in represented task information, but the projection is not a causal decomposition. In particular, task-related FBV tokens may reflect words retained from the conditioning text. The examples do not establish that FBVs encode meaning while FVs encode only output statistics, or that projection quality predicts intervention effectiveness.

G.2 Sensitivity to the number of conditioning items

Figure 4 varies the number of demonstrations or feedback statements at $k \in \{1, 2, 5, 10\}$ while keeping hyperparameters selected on the ID training pool at k=10 fixed. The FBVs use Method C, rule feedback, and cross-task corruption. All plotted runs use seed 0 and four tasks. The curves show that performance depends on the task, conditioning source, and number of items. Some feedback-vector conditions induce task behavior with few statements, whereas FV performance can increase substantially with additional demonstrations. This comparison does not match token counts or in-

Task	Function vector	Feedback vector (rule)
antonym	chez, -cloud, elsewhere	reversed, Reverse, reversing, 逆
present→past	stil, erland, plier	did, verbs, sentences
singular→plural	airs, erals, aries	ret, understanding, _plural
country→capital	stadiums, arbon, Stadium	cities, city, country
english→french	airs, nict, lets	French, langue, franc
capitalize	YES, eters, Unsupported	capitalized, capitalize

Table 14: Qualitative unembedding projections of FVs and Method-C FBVs (Llama-3.2-1B; rule flavor; cross-task corruption for FBVs). These token lists describe the projections and do not establish a semantic-versus-output decomposition.

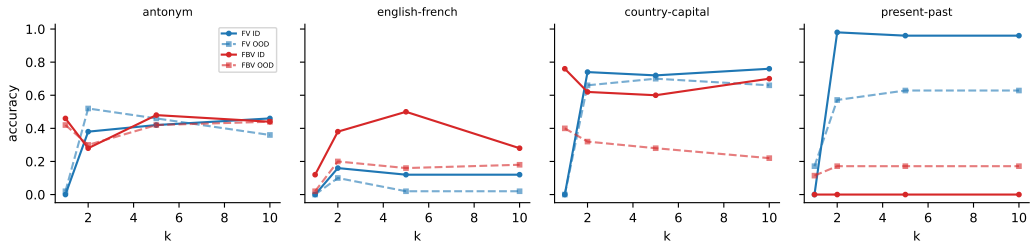


Figure 4: Accuracy versus the number of conditioning items on Llama-3.2-1B, using seed 0 on four tasks. FVs use demonstrations; FBVs use Method C with rule feedback and cross-task corruption. Both use $k \in \{1, 2, 5, 10\}$. Solid lines show ID performance and dashed lines show OOD performance. Layer and α are selected on the ID training pool at k=10 and frozen across the sweep.

formation content: a single rule can specify a mapping more directly than a single input–output pair. Moreover, hyperparameters selected at k=10 may be suboptimal at smaller k.

G.3 Association between token and vector accuracy

The 146 conditions comprise 126 main-suite cells (seven models × six tasks × three flavors, averaged over five seeds) and 20 single-seed Llama-3.1-8B breadth cells. The latter are enumerated in Appendix A. Let a_{cms} be method m's test accuracy for condition c and extraction seed s, and let $\hat{m}_{cs}$ be the method with the highest training-pool accuracy. The best-FBV statistic used here and in the template analysis is

$$b_c = \frac{1}{S_c} \sum_s a_{c\hat{m}_{cs}s}. \tag{2}$$

Table 1 and its win counts instead select one method per condition from seed-averaged training-pool accuracy. The two selection orders differ by at most .10 per condition, with mean difference .008. Using the latter order gives pooled $r = .847$ over 146 conditions and $r = .842$ over the 126 main conditions, compared with .857 and .855 for Eq. 2.

Within-model and within-task associations. Table 15 shows that the association remains after removing model means, and after jointly regressing token and vector accuracies on model and task indicators and correlating the residuals. The .81 model-adjusted result uses the 126 main cells. The corresponding 146-cell result is .83, treating the breadth extension as a separate group; merging it into the Llama-3.1-8B group also rounds to .83. Per-model main-suite correlations range from .76 to .92, except Qwen3-1.7B (.51). These adjustments reduce explanations based solely on average model or task difficulty without isolating a causal mechanism.

The association is present without method selection. Table 16 reports each method separately. Method A has a stronger pooled correlation than the best-of-methods summary; Method C has a weaker one. For Method B, pooled Spearman correlation is .75 and per-model Pearson correlations are .79–.92 except Llama-3.1-8B (.36) and Qwen3-1.7B (.16). Thus, method selection is not necessary for an association, but its magnitude depends on the extractor and model.

Conditions / adjustment	Pearson r	Spearman ρ
All conditions (146)	.86	.85
Main suite (126)	.86	.84
Main suite, model means removed	.81	–
Main suite, model and task effects removed	.74	–
Breadth extension (20)	.88	–

Table 15: Association of token and best-FBV accuracy. Main-suite results use 126 five-seed-averaged cells; breadth results use 20 single-seed cells. The joint adjustment removes model and task effects by least squares. Dashes indicate coefficients not reported.

Vector	All 146	Main 126	Within-model	Floor maximum
Best A/B/C	.86	.86	.81	.19
A	.89	.89	.84	.01
B	.76	.76	.72	.19
C	.58	.64	.56	.02

Table 16: Pearson correlations with token accuracy by extractor. Within-model correlations remove model means over the 126 main cells. The final column gives maximum vector accuracy among the six conditions with token accuracy below .10.

Observed floor behavior. All six token-floor conditions have best-FBV accuracy at most .19. Of the remaining 140 conditions, 139 yield best-FBV accuracy above .10; the exception is Gemma-2-2B capitalize-rule (tokens .25, best FBV .06). These are properties of the evaluated conditions, not a general bound. Both accuracy axes require labels, and Method C uses correct-answer probability for AIE. Moreover, token accuracy itself changes with prompt format. The results therefore do not provide a label-free test of feedback comprehension.

G.4 Paraphrase, negation, and misleading-feedback controls

For antonym, Method-B vectors extracted from two disjoint halves of the paraphrase set have cosine similarity .981. Their downstream accuracies are .34 and .40, compared with .38 for the full set. A negatively phrased correct rule ("do not give a synonym; give the opposite") yields accuracy .40. These results suggest limited sensitivity to the tested wording changes, without establishing invariance to paraphrase or general handling of negation.

A vector extracted from the incorrect instruction "output a synonym" reduces antonym accuracy to .00, with an input echo rate of .02. However, it does not produce consistent synonym outputs; responses instead drift toward definition-like elaborations. Thus, this control shows that changing the feedback can disrupt the original behavior, but does not demonstrate reliable redirection to the alternative task.

G.5 Corrective framing

In a three-seed Llama-3.2-1B probe, restating critiques in the third person under Q/A raises present→past token accuracy from .07 to .25, but does not improve antonym or capitalize. For antonym, Method-B vectors extracted from second- and third-person critiques reach .34 and .28 in a single run. The vector comparison concerns a different task from the reported token improvement and does not establish invariance to corrective framing.

H Additional OOD results and diagnostic figures

H.1 ICL seed variation

The B and C runs recompute their baselines. Tokens and ICL agree across the two sets of files, while FV means can differ by .01; the antonym ID FV mean is .48 in the C runs and .49 in the B runs.

Task (shift)	ID	OOD
antonym (rare)	$.56 \pm .05$	$.69 \pm .03$
present→past (irregular)	$1.00 \pm .00$	$.86 \pm .05$
country→capital (rare)	$.95 \pm .02$	$.85 \pm .02$
english→french (rare)	$.77 \pm .01$	$.57 \pm .02$

Table 17: Three-seed ICL means and reported seed standard deviations on Llama-3.2-1B, using the baselines from the Method-C OOD runs. The same queries are used across extraction seeds.

To avoid mixing baseline sources, Table 3 uses all baseline columns from C runs. Only aggregate accuracies were retained for these runs, so paired per-query confidence intervals are not reported.

H.2 SINGLE-SEED TRANSLATION REFERENCE

Table 18 lists the single-seed english→french OOD reference run.

Condition	ID	OOD	ID − OOD
Zero-shot	.00	.00	.00
ICL ($k = 10$)	.78	.54	.24
Tokens ($k = 5$)	.42	.24	.18
FV ($k = 10$)	.12	.02	.10
FBV-B ($k = 10$)	.28	.06	.22
FBV-C ($k = 10$)	.28	.18	.10

Table 18: Historical single-seed english→french OOD reference. These values are separate from the three-seed ICL summary in Table 17 and are not presented as three-seed vector means.

H.3 LAYER SWEEPS

Figures 5 and 6 show the reference FV and task-vector layer sweeps.

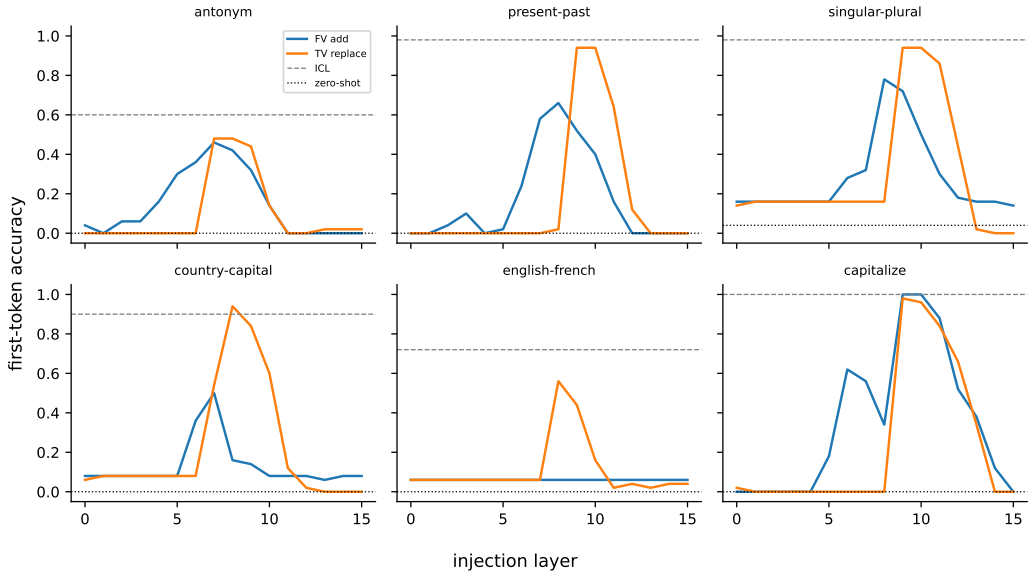


Figure 5: Reference FV-addition and task-vector-replacement layer sweeps on Llama-3.2-1B, evaluated with the first-token proxy. Final reported task accuracies use full generation.

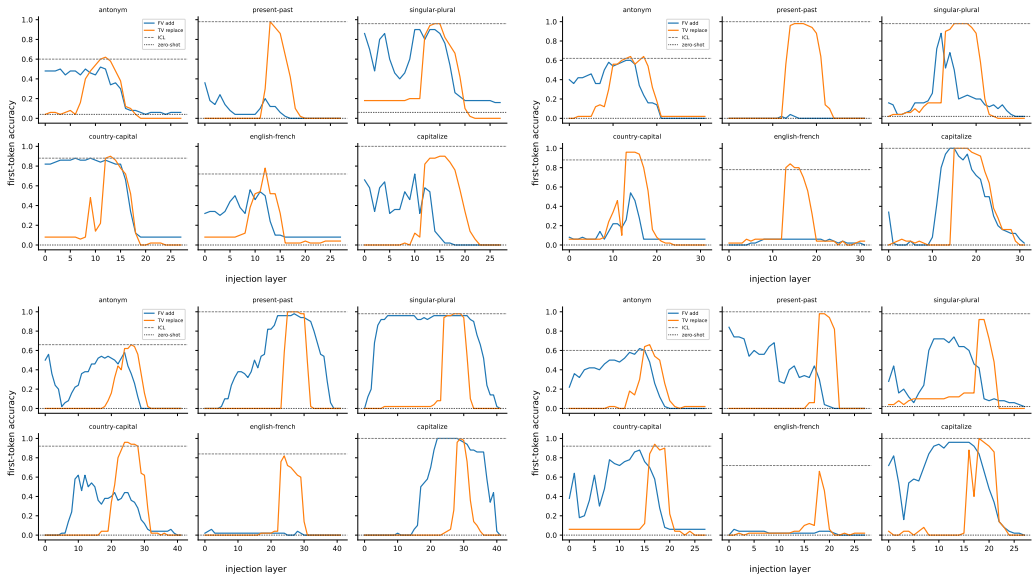


Figure 6: Reference first-token layer sweeps for GPT-J-6B and Llama-3.1-8B (top), and Gemma-2-9B and Qwen3-1.7B (bottom).

H.4 Additional geometry visualizations

Figures 7–9 give additional t-SNE views of the FV and FBV geometry.

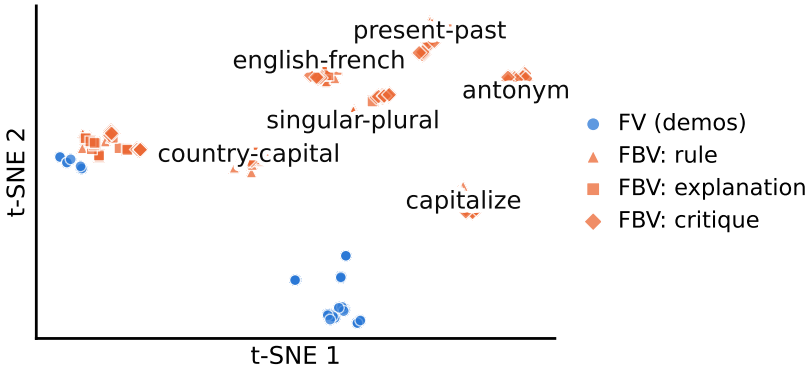


Figure 7: Llama-3.1-8B t-SNE visualization corresponding to Figure 3: six tasks, ten seeds, one FV and one Method-C FBV per flavor and seed. All numerical geometry statistics use the original vector space; distances in this projection are for visualization only.

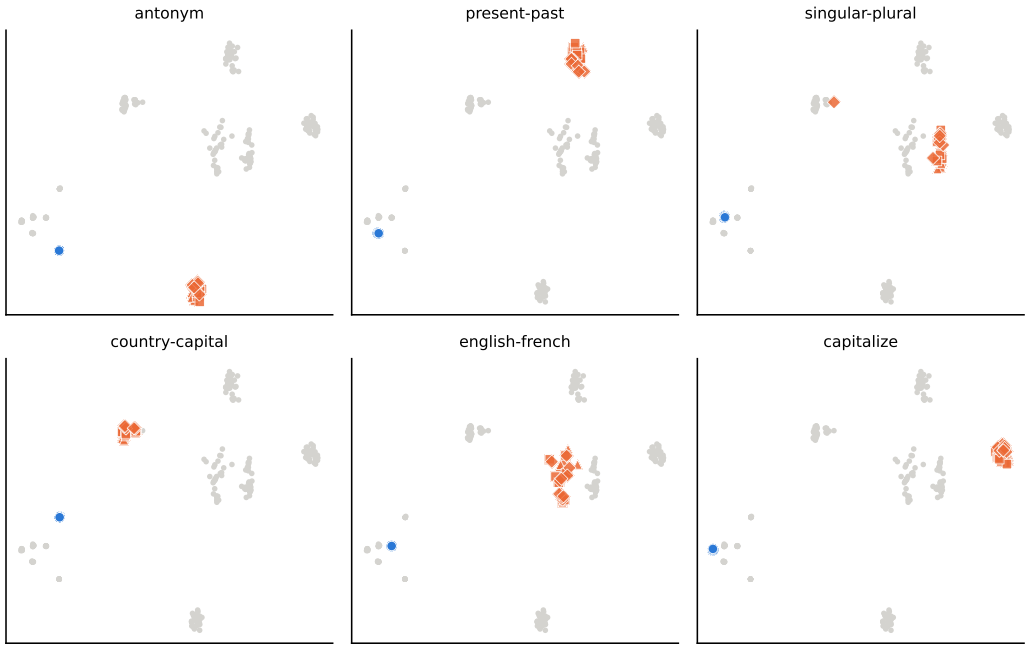


Figure 8: Per-task views of the Llama-3.2-1B embedding. Each panel highlights one task against the complete embedding in gray. Multiple seed vectors can overlap visually; the plot contains ten FVs per task, not a single FV observation.

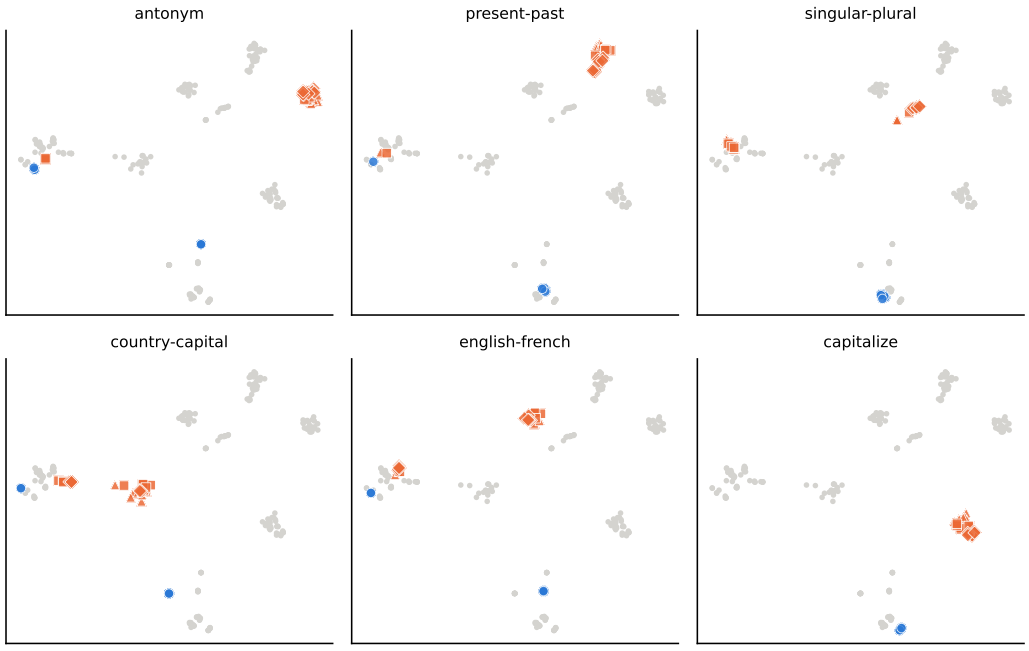


Figure 9: Per-task views for Llama-3.1-8B, with the same extraction and visualization protocol as Figure 7.

H.5 CONDITIONING-COUNT SWEEPS AT LARGER SCALES

Figure 10 repeats the conditioning-count sweep on the two larger models.

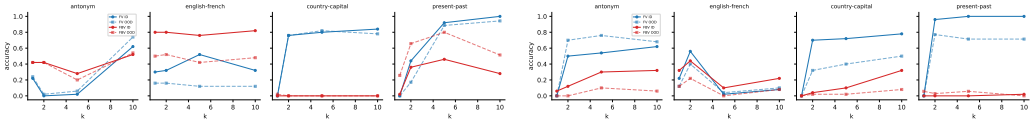


Figure 10: Conditioning-count sweeps on Llama-3.1-8B (left) and Gemma-2-9B (right), using seed 0 and four tasks. FVs use demonstrations; FBVs use Method C, rule feedback, and cross-task corruption. Both vary $k \in \{1, 2, 5, 10\}$. Layer and scale are selected on the ID training pool at $k = 10$ and frozen across the sweep. These are single-seed diagnostics, not the three-seed Method-B OOD comparisons.

I EXPLORATORY COMPOSITION PROBES

We add independently extracted Method-B vectors for two task pairs: antonym plus capitalize, and plural plus capitalize. At 1B, joint exact-match accuracy is .00–.02, despite partial expression of the component transformations. In the reported diagnostic, 76% of outputs are capitalized and base-rule accuracy ignoring case is .30.

At 8B, antonym plus capitalize reaches .38 joint accuracy, compared with .00 for either vector alone and .02 for both rules as tokens. Plural plus capitalize reaches only .06, versus .44 for tokens; capitalization is applied at .90 while pluralization is often lost. The two pairs therefore do not support a general compositionality claim. They provide an example of successful composition at 8B and a failure under the same broad construction. Both use capitalization as the modifier, so the tested combinations are narrow. This differs from combining an instruction-derived and a demonstration-derived vector for the same task (Davidson et al., 2025).