CORE-FV: FROM MARGINAL TO CONDITIONAL HEAD SELECTION FOR FUNCTION VECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Function vectors provide a compact mechanism for extracting and transferring task-specific behaviors from in-context demonstrations by aggregating the contributions of a small subset of attention heads. Existing approaches typically identify these heads using marginal importance scores, but evaluating heads independently overlooks interactions among them, causing redundant or complementary heads to be selected suboptimally. We introduce **CORE-FV** (Conditional Ordering through Removal Evaluation for Function Vectors), a conditional head selection framework that evaluates each attention head in the context of the remaining intervention set. CORE-FV performs an adaptive backward refinement procedure that iteratively removes heads whose absence minimally affects the induced task behavior, producing an interaction-aware ordering of heads. Across eight in-context learning transformations and multiple open-weight language models, CORE-FV generally outperforms a matched Average Indirect Effect baseline, achieving average accuracy gains of up to 11.84 percentage points. Moreover, on composite tasks requiring sequential or parallel transformations, CORE-FV substantially outperforms marginal head selection, improving average accuracy from 41.81% to 87.34% on Llama-3.1-8B. These results highlight that head contributions can be inherently context-dependent, and that interaction-aware selection provides a more effective foundation for constructing compact and algebraically composable task representations.

1 INTRODUCTION

Large language models (LLMs) can infer and execute a new input–output mapping from a handful of demonstrations in their context, a capability known as in-context learning (ICL; Brown et al., 2020). Mechanistic studies suggest that the task specified by such demonstrations is represented internally in a compact and manipulable form. A single hidden state read out at the end of a demonstration prompt recovers much of the few-shot behavior when patched into a zero-shot forward pass (Hendel et al., 2023), and in-context vectors distilled from demonstrations can steer a model's latent states toward the demonstrated task (Liu et al., 2024). Function vectors (FVs; Todd et al., 2024) localize this signal to model components: an FV is the sum of the task-conditioned outputs of a small set of attention heads, and adding it to the residual stream of a zero-shot prompt elicits the demonstrated behavior. FVs thereby connect a mechanistic account of ICL to the broader practice of activation steering (Turner et al., 2023; Li et al., 2023; Zou et al., 2023), and later evidence supports their premise that few-shot ICL relies primarily on such heads (Yin & Steinhardt, 2025).

A complementary line of work improves the vector itself rather than its components. State vectors refine an extracted vector at test time through inner and momentum optimization and aggregate it across many demonstrations (Li et al., 2024), and learned task vectors (LTVs) are trained directly by gradient descent and surpass extracted vectors in accuracy (Yang et al., 2026). Notably, mechanistic analysis of LTVs indicates that they steer predictions mainly through attention-head output–value circuits, with a small subset of "key heads" being most decisive (Yang et al., 2026). This underscores the importance of deciding which heads take part in a task vector.

The quality of an FV hinges on which heads it contains. The standard answer is the average indirect effect (AIE; Todd et al., 2024), a causal-mediation score (Pearl, 2001; Vig et al., 2020; Meng et al., 2022). AIE measures how much patching a head's task-conditioned mean activation into a corrupted,

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.

label-shuffled prompt restores the probability of the correct answer, and the top-k heads are retained. AIE is causally grounded but *marginal*: every head is assessed with all other components held at a fixed reference, so the score cannot reveal whether a head remains necessary once the other selected heads are also intervened on. Recent methods depart from single-head patching in different ways. Hu et al. (2025) optimize per-head coefficients jointly under a sparsity objective, and Pham et al. (2026) attribute head relevance with layer-wise relevance propagation (Achtibat et al., 2024) at roughly the cost of a backward pass; both report improvements over AIE. In both cases, however, a single attribution is computed once and the retained heads are read off from it. Neither re-evaluates a head's contribution after other heads have been withdrawn from the intervention.

Attention heads, however, do not contribute independently. Many heads can be removed with little loss (Michel et al., 2019; Voita et al., 2019), heads compose through the residual stream (Elhage et al., 2021), and ablating a component can trigger compensation by others, from backup heads in the indirect-object-identification circuit (Wang et al., 2023) to self-repair across layers (McGrath et al., 2023). Interactions can also be constructive or destructive: Qu et al. (2025) use Harsanyi dividends to show that significant cooperative gains are concentrated in sparse head combinations while others act competitively, and Held & Yang (2023) show with Shapley values that removing interfering heads can improve multilingual transfer. Under such interactions, two individually strong heads may carry overlapping information and both score highly under a marginal criterion although one suffices, whereas a head that matters only alongside others may be ranked low.

We therefore propose **CORE-FV** (Conditional Ordering through Removal Evaluation for Function Vectors), a task-specific head selection procedure based on conditional backward elimination. CORE-FV keeps the standard construction of function vectors from mean activation displacements between few-shot and zero-shot prompts and changes only which heads enter them. It starts from the full set of candidate heads and repeatedly asks what happens to the joint intervention when a head is withdrawn. At each step, every remaining head is scored by how well the set without it steers the model in the zero-shot setting, measured as the fraction of training queries whose entire target sequence is predicted correctly (sequence-level exact match); the most removable heads are eliminated and the rest are re-evaluated in their new context. Reversing the elimination order yields a complete ordering whose nested prefixes define candidate function vectors, so a head's importance is conditional on the heads that remain and the ordering is sensitive to redundancy, complementarity, and interference. In contrast to LTVs and state vectors, the procedure involves no gradient-based training or test-time optimization of the vector: it requires only forward passes, and an adaptive coarse-to-fine schedule removes groups of heads whenever doing so preserves correctness, keeping the backward search practical. Here, "deletion" means withdrawing a head's task-conditioned displacement from the injected steering set; the model itself is left intact, which distinguishes our procedure from head pruning (Michel et al., 2019; Voita et al., 2019) and from circuit discovery by edge ablation (Conmy et al., 2023), whose deletion-based logic we borrow. The search is greedy and offers no guarantee of minimality, so a head's exclusion should not be read as evidence of its irrelevance.

Since AIE is the selection rule of the original FV pipeline, it can be exchanged for CORE-FV with every other component held fixed, and we use it as the controlled comparison. Consistent with the conditional view, an ablation that freezes the ordering after a first pass of single-head removals, instead of re-evaluating heads as they are removed, causes the largest degradation among the design choices we test (Section 5).

We evaluate CORE-FV on eight input–output tasks from Todd et al. (2024): Antonymy, Capitalize, Present–Past, Singular–Plural, Country–Capital, English–French, Next-item, and Prev-item. Against the matched AIE baseline, in which only the head ordering differs, CORE-FV raises the eight-task average accuracy from 88.94% to 91.95% (+3.01 points) on GPT-J 6B, from 81.79% to 93.63% (+11.84) on Llama-3.1-8B, and from 85.40% to 92.02% (+6.62) on Qwen3-4B-Base; on Mistral-7B-v0.3, AIE is slightly higher, with 92.74% versus 92.24%. On OLMoE-1B-7B, CORE-FV again improves the average, from 89.68% to 91.51% (+1.83). Across all five models, CORE-FV is better than AIE on 23 of 40 model–task pairs, ties on one, and is lower on 16. It also exceeds the stronger of the SV or LTV baselines in average accuracy on every model, by 4.85, 0.82, 12.61, 3.61, and 3.90 points on GPT-J, Llama-3.1-8B, Qwen3-4B-Base, Mistral-7B-v0.3, and OLMoE-1B-7B, respectively. Because AIE scores heads independently, a set of heads that is effective only jointly can score poorly head-by-head and be pushed out of AIE's selected prefix even though it would work well together; the largest gains of CORE-FV over AIE occur on several such difficult transformations.

We further compare the same selectors on composite tasks that chain two or three of the standard operations above or apply different operations to two operands in parallel (Table 2). The accuracy gap between CORE-FV and AIE widens substantially on these tasks relative to the standard ones. We attribute this to synergy among heads: composite tasks are harder, and the behavior they require depends more heavily on heads acting together rather than on any one head in isolation, as reflected by the fact that among the 60 top-ranked AIE heads, 81.0% of head pairs interact positively on held-out prompts (Appendix B.5). Because CORE-FV scores each head against the heads still retained rather than against a fixed reference, it captures this synergy more effectively than a marginal ranking does: on Llama-3.1-8B, averaged over the tasks in Table 2, CORE-FV raises accuracy from 41.81% to 87.34%, ahead of AIE on every composite task but one, where the two rankings tie. Function vectors built from CORE-FV heads also compose more directly: on a separate algebraic vector-composition benchmark on GPT-J, composition without target-level coefficient tuning reaches 54.44% versus 24.88% and 28.62% for task-specific and global AIE, and the three rankings become comparable once coefficients are refined on the composed task's validation split (Section 4.4, Appendix B.5).

Our contributions are as follows.

- **A conditional criterion for head selection.** We formulate head selection for function vectors as conditional deletion: each head is scored against the heads still retained in the intervention rather than against a fixed reference.
- **CORE-FV.** An adaptive backward-refinement algorithm produces a complete task-specific ordering of heads, requiring every accepted group deletion to preserve sequence-level correctness on the refinement set, without gradient-based training of the vector.
- **A controlled evaluation protocol.** It combines per-size tuning of the steering strength, identical vector construction and intervention for all selectors, and an ablation of the conditional re-evaluation itself.
- **Empirical evidence.** On eight standard ICL tasks, conditional ordering improves over a matched AIE baseline on four of five models and exceeds state-vector and learned-task-vector baselines on average on all five; on composite tasks it raises accuracy from 41.81% to 87.34%, and it yields function vectors that compose without target-level tuning.

2 Related Work

Task representations in in-context learning. The task inferred from demonstrations is materialized as an explicit, manipulable object in a model's activations. Hendel et al. (2023) show that a single hidden state read at the end of an in-context prompt recovers much of the few-shot behavior when patched into a zero-shot pass. Todd et al. (2024) localize the signal to a small set of attention heads identified by causal mediation analysis (Pearl, 2001; Vig et al., 2020) and construct a function vector (FV) from those heads' task-conditioned mean outputs. Liu et al. (2024) obtain a related object by principal component analysis over paired latent states. Later work improves the representation while leaving its localization intact: Li et al. (2024) refine an extracted vector at test time through momentum-style optimization and aggregation across many demonstrations, reading whole early layers rather than a selected set of heads, and Yang et al. (2026) instead learn the vector directly by gradient descent, reporting that the learned vector steers predictions mainly through the output–value circuits of a small subset of "key" attention heads. We compare against both as additional baselines in our main evaluation, denoting them SV (state vector) and LTV (learned task vector); unlike our method, both optimize the vector's entries directly rather than selecting which heads should contribute to it, and neither addresses which heads an extracted, non-learned FV should retain. We hold the FV construction fixed and ask the prior question of which heads should enter it.

Locating task-relevant attention heads. Todd et al. (2024) score each head by its average indirect effect (AIE): the recovery in the correct answer's probability when that head's mean-conditioned activation is patched into a label-shuffled prompt, retaining a fixed top-k. Yin & Steinhardt (2025) confirm this is the right family of components, finding across twelve models that few-shot ICL depends primarily on FV heads rather than induction heads (Olsson et al., 2022). Two recent studies revisit head-level attribution for in-context behavior with methods that no longer patch heads one at a time. Hu et al. (2025) use an optimization-based procedure to localize a narrow addition-learning behavior in Llama-3-8B to a small number of jointly identified attention heads, and further

characterize the low-dimensional activation subspace those heads carry; Pham et al. (2026) instead attribute head relevance with layer-wise relevance propagation (Achtibat et al., 2024), which requires roughly the cost of a single backward pass. Both differ from AIE in the same respect: rather than holding every other head at a fixed reference while scoring one head at a time, each computes an attribution over the candidate heads jointly, so the resulting score is not marginal in AIE's sense. This is a genuine improvement over per-head patching, since it lets the attribution reflect shared or competing contributions among heads at the point where it is computed. However, in both cases the attribution is computed once, from the full or fixed candidate pool, and a top-k prefix is then read off from it; neither re-evaluates a head's contribution after other heads have already been removed from the intervention, so a head's assessed importance does not adapt as the retained set shrinks. Pham et al. (2026) raise a related concern, attributing part of their gains over AIE to better capturing interactions among heads. CORE-FV targets exactly the interactions that a one-shot joint attribution can still miss once heads start being withdrawn, by re-scoring the remaining heads at every step of the elimination.

Redundancy and interaction among attention heads. Attention heads do not contribute independently. Redundancy is well documented: large fractions of heads can be removed with little degradation (Michel et al., 2019; Voita et al., 2019). Interactions can also be constructive or destructive: Qu et al. (2025) analyze coalitions of heads through cooperative game theory and identify both positive cooperative and negative competitive effects, and Held & Yang (2023) show, in a multilingual transfer setting, that removing interfering heads can improve downstream behavior. The concern is sharpest where a behavior is implemented by several interacting components, as mechanistic analyses increasingly find: heads compose through the residual stream to implement higher-order computations (Elhage et al., 2021), ablating a component can trigger compensation by others across layers (McGrath et al., 2023), and indirect object identification has been reverse-engineered into a circuit of many heads spanning several functional classes (Wang et al., 2023). Under such interactions, two individually strong heads may be largely redundant, while another head may matter only alongside the rest of the computation. A score computed once, whether marginal or jointly attributed, cannot track this as the retained set changes.

3 METHOD

We propose a task-specific, deletion-based method for identifying a compact set of attention heads that jointly encode a task-relevant steering signal. The central principle is that the contribution of a head should be evaluated *in the context of the other heads with which it acts*. We therefore initialize the search with all attention heads and progressively remove those whose deletion least affects the behavior induced by the remaining set. The resulting importance of a head is conditional on the current set of retained heads, making the ranking sensitive to redundancy, complementarity, and interference among heads.

3.1 TASK-CONDITIONED HEAD VECTORS

We first construct a task-conditioned steering vector for every attention head. The goal is to isolate the change in a head's representation that is induced by observing demonstrations of the task, while keeping the query fixed.

For each training example i, we construct two matched prompts. The in-context prompt p_i^{ICL} contains demonstrations of the target task followed by the query x_i, whereas the zero-shot prompt p_i^0 contains the same query without demonstrations. Let $a_{\ell j}(p) \in \mathbb{R}^{d_h}$ denote the output of attention head j at layer ℓ, evaluated at the final prompt token before the attention output projection. We define the ICL-induced displacement of the head as

$$\Delta_{\ell j}^{(i)} = a_{\ell j}(p_i^{\mathrm{ICL}}) - a_{\ell j}(p_i^0), \qquad \bar{\Delta}_{\ell j}^{t} = \frac{1}{|\mathcal{P}_t|} \sum_{i \in \mathcal{P}_t} \Delta_{\ell j}^{(i)}, \tag{1}$$

where $\mathcal{P}_t$ is the set of successful training prompts used for task t. Since the ICL and zero-shot prompts contain the same query, $\Delta_{\ell j}^{(i)}$ captures the change in the head representation associated with introducing demonstrations of the task. Averaging these changes yields a task-level displacement for each head.

The displacement $\bar{\Delta}^t_{\ell j}$ is expressed in the head-output space. We map it to the residual stream using the corresponding slice of the attention output projection:

$$v^t_{\ell j} = W^O_{\ell,j} \bar{\Delta}^t_{\ell j}. \tag{2}$$

We use only the linear projection and omit the bias, since the steering signal is defined as a difference between matched head activations. For a selected set of heads $\mathcal{H}$, we preserve the layer at which each head contributes. The task-conditioned vectors belonging to the same layer are summed,

$$v^t_\ell(\mathcal{H}) = \sum_{j:(\ell,j)\in\mathcal{H}} v^t_{\ell j}, \qquad r_\ell \leftarrow r_\ell + \alpha\, v^t_\ell(\mathcal{H}), \tag{3}$$

where r_ℓ is the residual representation and α controls the strength of the intervention. Thus, heads from different layers are not collapsed into a single global vector; each selected head contributes to the intervention at its native layer.

3.2 Deletion-Based Joint Refinement

We identify important attention heads through a backward deletion procedure that evaluates each head in the context of the other heads currently retained. The search starts from the complete set of attention heads, $\mathcal{H}_0$. At each refinement step, let $\mathcal{H}$ denote the current set of heads. For every $h \in \mathcal{H}$, we remove h and evaluate the intervention produced by the remaining set $\mathcal{H} \setminus \{h\}$.

Candidate deletions are evaluated on zero-shot training queries using gold-prefix scoring. For a target sequence $y_{1:T}$, each target token y_t is evaluated conditioned on the ground-truth prefix $y_{<t}$, rather than on previously generated tokens. An example is counted as correct only if every target token is the top-1 prediction at its corresponding position. When multiple deletions achieve the same sequence-level correctness, the mean log-probability assigned to the target tokens is used as a tie-breaker. Therefore, the head whose removal leaves the best-performing remaining set is considered the most removable at that refinement step.

Importantly, head importance is not treated as a fixed individual property. After each accepted deletion, the retained set is updated and the remaining heads are evaluated again. Thus, the importance of a head depends on the other heads that are still present in the intervention. This allows the refinement process to account for interactions such as redundancy, complementarity, and interference between heads.

Without grouped deletions, evaluating every possible single-head removal at each step requires $\Theta(H^2)$ subset evaluations, where H is the number of model heads. To make backward refinement computationally tractable when many heads remain, we use an adaptive coarse-to-fine deletion strategy. After ranking all single-head deletions, a group of the most removable heads is proposed for joint deletion. The proposal is accepted only if removing the group preserves the current sequence-level correctness,

$$C(\mathcal{H} \setminus \mathcal{R}) \geq C(\mathcal{H}), \tag{4}$$

where $\mathcal{R}$ is the proposed removal set and $C(\cdot)$ denotes the sequence-level correctness fraction under the gold-prefix scoring rule above. If the proposal decreases correctness, smaller deletion groups are considered. If no multi-head deletion satisfies this condition, the best single-head deletion is performed so that the refinement can continue. When group deletions repeatedly remove a constant fraction of the remaining heads, this falls to $\Theta(H)$ subset evaluations, matching the head-count scaling of original Function Vectors and making refinement more feasible for larger models.

The procedure terminates when a single head remains. The final importance ranking is obtained by reversing the deletion order: heads removed later are ranked as more important because their contribution remains necessary after more alternative heads have already been eliminated. The resulting task-specific ordering $\pi_t = (h_{(1)}, h_{(2)}, \ldots)$ is then used to construct Top-K head sets for the final intervention.

3.3 Final Intervention

The ranking $\pi_t = (h_{(1)}, h_{(2)}, \ldots)$ defines a nested family of interventions $\mathcal{H}^t_K = \{h_{(1)}, \ldots, h_{(K)}\}$. The number of retained heads K and steering strength α are selected using validation performance.

Once selected, the configuration is fixed and used for zero-shot generation. During autoregressive inference, the corresponding task-conditioned vector is added at each selected head's native layer at every generation step. Further implementation and evaluation details are provided in Appendix A.

4 EXPERIMENTS

We evaluate whether conditional backward elimination produces a more effective ordering of task-relevant attention heads than a marginal causal score. Our main controlled comparison is against a matched AIE baseline, for which the experimental setup is held fixed and only the head-selection mechanism is changed. We also compare against existing task-vector methods to place the results in a broader context.

4.1 EXPERIMENTAL SETUP

Tasks and evaluation protocol. We evaluate eight input–output transformations drawn from the Function Vectors benchmark (Todd et al., 2024): Antonymy, Capitalize, Present–Past, Singular–Plural, Country–Capital, English–French, Next-item, and Prev-item. For each task, we use a deterministic 60/20/20 train/validation/test split and 10-shot ICL filtering, following the protocol in Appendix A. Task-conditioned head vectors and rankings are constructed using training data only; K and α are selected on the validation split and then fixed for test evaluation. Main experiments are repeated across three seeds.

Models. We evaluate five decoder-only language models: GPT-J 6B, Llama-3.1-8B, Qwen3-4B-Base, Mistral-7B-v0.3, and OLMoE-1B-7B. The grouped-query models have fewer key–value heads, but each query-head output occupies a distinct slice of the attention output projection and is treated as a separate candidate. Exact checkpoints, model dimensions, and candidate counts are reported in Appendix Table 5.

Compared methods. We compare CORE-FV against four baselines. For GPT-J, we include the published Function Vector (FV) results from Todd et al. (2024), adapted from the original paper rather than reproduced in our pipeline. We also compare with State Vectors (SV) (Li et al., 2024), which refine extracted task representations, and Learned Task Vectors (LTV) (Yang et al., 2026), which learn task vectors directly by gradient descent.

Our primary controlled baseline is Average Indirect Effect (AIE), the head-selection criterion used by Function Vectors. AIE scores each head independently by measuring the recovery in the probability of the correct first answer token when its clean task-conditioned mean activation is restored in a label-shuffled prompt. After the ranking is formed, the baseline follows the same intervention and evaluation pipeline as CORE-FV. Thus, the comparison isolates the effect of head selection: AIE uses a marginal ranking, whereas CORE-FV uses the conditional backward ordering introduced in Section 3.2.

4.2 MAIN RESULTS

Table 1 reports accuracy for each model and task. Relative to the matched AIE baseline, CORE-FV achieves higher average accuracy on GPT-J (91.95% vs. 88.94%), Llama-3.1-8B (93.63% vs. 81.79%), Qwen3-4B-Base (92.02% vs. 85.40%), and OLMoE-1B-7B (91.51% vs. 89.68%). AIE is slightly higher on Mistral-7B-v0.3 (92.74% vs. 92.24%).

The strongest improvements over AIE appear on Llama-3.1-8B and Qwen3-4B-Base, most visibly on English–French, where accuracy rises from 12.18% to 88.65% and from 49.02% to 85.00%, respectively. Beyond the controlled AIE comparison, CORE-FV also achieves higher average accuracy than both reproduced baselines, SV and LTV, on every model, often by several percentage points. At the same time, these baselines remain competitive on individual model–task pairs and attain the best result in several cases.

Model	Method	Antonymy	Capitalize	Present–Past	Singular–Plural	Country–Capital	English–French	Next-item	Prev-item	Avg.
GPT-J 6B	FV*	48.2±2.0	70.5±2.4	19.7±5.9	47.0±3.4	83.2±2.7	44.7±1.2	–	–	52.22±1.34
	SV	45.65±4.99	56.85±3.26	36.95±4.67	44.63±10.78	66.94±5.59	24.30±4.77	20.46±11.47	4.05±0.56	37.48±2.37
	LTV	81.13 ± 1.56	97.31 ± 0.59	**98.86 ± 0.81**	90.85 ± 4.74	87.80 ± 5.63	69.96 ± 0.92	**91.36 ± 1.34**	79.50 ± 10.16	87.10 ± 1.60
	AIE	81.81±2.72	**98.76±0.50**	97.69±0.85	**95.83±4.25**	92.49±3.45	82.75±0.73	77.60±2.82	84.55±5.96	88.94±1.13
	CORE-FV	**84.29±0.92**	98.55±0.29	**98.86±0.81**	93.37±1.24	**94.37±3.99**	**84.54±0.37**	88.76±7.52	**92.83±2.75**	**91.95±1.14**
Llama-3.1-8B	SV	82.82±3.34	98.75±0.02	89.63±2.77	89.12±10.07	**96.88±3.12**	74.93±9.68	94.37±2.77	72.34±7.48	87.35±2.51
	LTV	84.69±0.63	93.01±2.77	**97.15±0.80**	96.73±3.04	94.38±0.26	85.54±2.74	**96.03±2.24**	95.00±3.53	92.82±0.82
	AIE	**87.44±1.00**	**99.59±0.29**	87.41±3.29	**98.37±1.15**	88.73±2.56	12.18±0.67	90.55±4.54	90.07±2.01	81.79±0.84
	CORE-FV	86.59±1.26	98.77±0.87	**97.15±0.80**	98.33±2.36	92.50±4.89	**88.65±1.49**	90.45±2.47	**96.62±1.63**	**93.63±0.82**
Qwen3-4B-Base	SV	52.28±2.76	55.23±3.66	17.33±1.59	44.91±6.41	85.38±6.56	20.81±4.75	32.56±6.15	19.80±15.52	41.04±3.45
	LTV	81.27±2.29	12.58±1.33	95.94±2.97	82.56±8.24	**93.64±2.48**	**85.98±0.72**	93.67±4.06	89.61±2.48	79.41±1.33
	AIE	**81.68±1.72**	98.55±0.78	97.68±1.64	97.50±3.54	85.12±7.74	49.02±2.70	87.38±5.56	86.29±6.77	85.40±1.59
	CORE-FV	80.95±1.93	**98.97±1.06**	**98.27±1.41**	**98.35±1.16**	89.83±3.65	85.00±0.77	**94.44±4.96**	**90.37±2.96**	**92.02±0.93**
Mistral-7B-v0.3	SV	78.41±2.99	95.74±0.72	87.63±4.99	91.74±0.13	93.26±0.13	66.41±4.77	87.93±7.21	41.34±11.45	80.31±2.46
	LTV	**85.13±2.68**	54.91±21.08	**98.86±0.81**	**100.00±0.00**	96.18±1.46	86.68±1.74	92.02±2.19	**95.14±2.07**	88.62±2.70
	AIE	84.92±1.68	**100.00±0.00**	95.44±2.64	99.17±1.44	92.50±6.00	85.95±0.24	**93.61±1.31**	90.31±0.35	**92.74±0.88**
	CORE-FV	83.05±2.45	98.98±1.27	96.59±1.70	95.89±3.74	**97.06±2.94**	**87.96±1.38**	90.36±4.94	87.99±3.70	92.24±1.07
OLMoE-1B-7B	SV	29.52±3.10	58.48±1.01	27.63±9.71	52.28±11.24	86.26±3.97	33.16±3.55	24.45±5.70	11.84±10.74	40.45±2.61
	LTV	82.61±2.26	71.42±34.02	96.00±2.09	79.65±27.02	**95.57±2.45**	85.23±1.24	**96.79±1.18**	**93.63±2.86**	87.61±5.47
	AIE	80.56±0.96	**99.59±0.71**	**96.54±1.74**	**97.48±2.57**	91.92±5.44	79.50±0.08	87.26±6.82	84.60±7.95	89.68±1.53
	CORE-FV	**85.22±4.90**	99.18±0.35	96.52±3.01	97.46±2.56	91.92±4.72	**87.49±2.49**	90.43±2.16	83.84±2.95	**91.51±1.13**

Table 1: Exact-match test accuracy (%). Best and second-best results within each model are bolded and underlined, respectively. FV* is adapted from Todd et al. (2024).

	GPT-J-6B			Llama-3.1-8B		
Composite task	AIE	LTV	CORE-FV	AIE	LTV	CORE-FV
Antonym → French	62.94±6.38	70.86±2.21	**76.60±12.39**	36.12±31.42	**84.41±4.94**	78.79±6.52
Antonym → French → Capitalize	29.37±19.67	44.29±5.15	**46.03±22.78**	17.90±4.67	62.51±8.08	**82.28±8.09**
Next-item → Capitalize	83.29±11.09	27.53±6.39	**86.18±2.99**	55.04±8.33	45.87±8.99	**83.91±2.73**
Singularize → Capitalize	96.00±0.00	**97.33±1.15**	96.67±0.94	**100.00±0.00**	98.67±1.15	**100.00±0.00**
Singularize ⊕ Antonym	0.00±0.00	2.56±1.28	**76.50±2.18**	0.00±0.00	1.21±1.19	**91.73±0.61**
Average	54.32±2.81	61.33±0.95	**76.39±7.76**	41.81±5.86	58.07±3.53	**87.34±2.82**

Table 2: Exact-match test accuracy (%) on composite tasks. Values are mean ± population standard deviation across seeds.

4.3 Generalization to Composite Tasks

The transformations above can often be solved by a single lexical or morphological operation. We therefore construct five additional tasks that require two or three operations: Antonym → French, Antonym → French → Capitalize, Next-item → Capitalize, Singularize → Capitalize, and a parallel task that singularizes one operand while producing the antonym of another. We use fixed, audited train, validation, and test partitions for every composite. For the controlled AIE comparison, only the ranking changes: AIE and CORE-FV use the same training-only displacement vectors, native-layer injection, eligible validation/test examples, and validation selection of (K, α) for each model and seed. Because the composites can require larger coalitions, we extend the candidate grid to $K \in \{1, \ldots, 10, 15, 20, 30, 40, 60, 70, 80, 90, 100\}$.

We evaluate both GPT-J-6B and Llama-3.1-8B. Table 2 reports the results. CORE-FV achieves the highest average accuracy on both models, reaching 76.39% on GPT-J and 87.34% on Llama, compared with 54.32% and 41.81% for AIE and 61.33% and 58.07% for LTV, respectively. Against the matched AIE baseline, CORE-FV improves all five GPT-J composites and four of five Llama composites, tying on Singularize → Capitalize. The largest gains appear on the parallel task, where AIE remains at 0% while CORE-FV reaches 76.50% on GPT-J and 91.73% on Llama. The head-count curves in Appendix B.3 show that this advantage remains consistent across a broad range of K values.

We attribute this widening gap to complementarity among heads playing a larger role as tasks grow harder: composite tasks require several transformations to be satisfied jointly on the same input, a setting in which single-head evaluations may be insufficient to identify the heads required jointly. AIE patches one head at a time into a corrupted, label-shuffled prompt while holding every other head at reference (Section 4.1); when success depends on several heads acting together, this marginal criterion cannot separate the heads a task truly depends on, so most candidates end up with similarly

Last-T task	10-shot ICL*	Direct FV*	Sum FV*	CORE-FV, component α	CORE-FV, refined α
Antonym	25 ± 2	2 ± 1	7 ± 2	12.00 ± 7.76	$\mathbf{22.33 \pm 2.25}$
Capitalize	91 ± 2	64 ± 3	76 ± 4	78.50 ± 10.26	$\mathbf{88.33 \pm 2.90}$
Country–Capital	32 ± 2	15 ± 3	$\mathbf{60 \pm 2}$	26.50 ± 4.42	50.50 ± 8.84
English–French	45 ± 4	16 ± 2	6 ± 2	4.83 ± 1.43	$\mathbf{33.17 \pm 8.66}$
Present–Past	89 ± 2	18 ± 1	29 ± 3	85.17 ± 2.05	$\mathbf{86.33 \pm 2.49}$
Singular–Plural	90 ± 1	28 ± 1	29 ± 3	69.83 ± 26.07	$\mathbf{94.50 \pm 1.08}$
Capitalize first letter	75 ± 1	76 ± 2	$\mathbf{95 \pm 0}$	53.33 ± 31.17	91.67 ± 0.94
Product–Company	35 ± 3	30 ± 2	41 ± 3	15.83 ± 21.33	$\mathbf{49.17 \pm 0.94}$
Average	60.25	31.13	42.88	43.25 ± 9.16	$\mathbf{64.50 \pm 0.39}$

Table 3: Top-1 test accuracy (%) for algebraic vector composition. CORE-FV uses GPT-J and $K = 10$ heads per component. CORE-FV entries report the mean and population standard deviation across three seeds. Starred columns are adapted from Table 7 of Todd et al. (2024) and were not rerun. Bold marks the best intervention column, excluding the ICL reference.

small, indistinguishable scores and the resulting top-k selection is only weakly tied to actual head importance. This is sharpest on the parallel task (Singularize $\oplus$ Antonym), where the AIE-selected prefix yields 0% accuracy on both models (Table 2), a clean instance of the univariate ranking failure already noted in Section 1, in which heads that are individually uninformative but jointly indispensable are exactly where marginal scores stop tracking true importance. CORE-FV avoids this failure mode because it scores a head by whether withdrawing it degrades the correctness of the heads still retained (Section 3.2), crediting a contribution a head only makes jointly rather than penalizing it for lacking one alone, consistent with the positive pairwise interactions reported among top-ranked AIE heads below. As a descriptive check, we measure pairwise patching interactions within AIE Top-60 pools on held-out prompts. Across the 12 task–seed pools with Top-60 diagnostics, 81.0% of head pairs have positive mean interaction. This analysis does not enter either ranking or model selection; full statistics and the number of seeds per task are in Appendix Table 8.

4.4 Composing Selected Function Vectors

We next ask whether the selected vectors remain useful when combined algebraically. Following the list transformations of Todd et al. (2024), for each operation T we form

$$V^*_{\text{Last-}T} = \alpha_T V_{\text{First-}T} + \alpha_L V_{\text{LastCopy}} - \alpha_F V_{\text{FirstCopy}}. \tag{5}$$

The eight targets cover lexical, morphological, factual, and positional transformations. For compatibility with the original composition protocol, we use GPT-J, inject the summed residual vector once at layer 9, and score the first target token. We therefore treat this as a separate stress test rather than an additional estimate of the native-layer exact-generation results above.

We compare CORE-FV with task-specific AIE and the fixed global AIE ranking used in the original setup, for $K \in \{3, 5, 8, 10\}$ heads per component. The *component-α* protocol directly combines the coefficient selected on each component's validation set. The *refined-α* protocol starts from those values and makes one coordinate pass over $(\alpha_T, \alpha_F, \alpha_L)$ on the composed task's validation split; test examples are never used for tuning.

Table 3 gives the predetermined $K = 10$ comparison to the published Function Vectors composition results. With the proposed ranking, target-level coefficient refinement improves the eight-task average from 43.25% to 64.50% and is especially important for Last-Capitalize first letter ($+38.34$ points) and Last-Product–Company ($+33.34$ points). The refined vector numerically exceeds the adapted ten-shot ICL average and the published summed-vector result by 4.25 and 21.62 points, respectively, although the adapted columns are contextual rather than matched reruns.

Across the three seeds, target-level refinement improves 23 of the 24 task–seed evaluations and reduces cross-seed variation in the macro-average. Appendix B.5 reports the seed-specific results and selected coefficient tuples, together with the complete seed-42 comparison across rankings, head counts, and coefficient protocols.

5 ABLATIONS

We isolate five choices in the proposed construction using GPT-J across three seeds. Unless stated otherwise, each variant retains the same data splits, training-only estimation, validation search, and exact-generation test protocol as the full method. Results are reported as mean accuracy and sample standard deviation across seeds. Appendix Table 14 records every validation-selected configuration.

Variants. *Raw mean* replaces the matched displacement $\mathbb{E}[a^{\mathrm{ICL}} - a^{0}]$ with $\mathbb{E}[a^{\mathrm{ICL}}]$; *single layer* sums the selected contributions and injects once at GPT-J layer 9; *fixed strength* sets $\alpha = 1$ and tunes only K; *global ranking* shares one training-only ordering across all tasks while retaining task-specific displacements; and *non-accumulative ranking* fixes the task-level ordering after first-pass single-head removals rather than updating it as heads are successively removed. Appendix C gives the complete protocols.

Method	Antonymy	Capitalize	Present–Past	Singular–Plural	Country–Capital	English–French	Avg.	Δ Avg.
CORE-FV	$\mathbf{84.29 \pm 1.13}$	$\underline{98.55 \pm 0.35}$	$\mathbf{98.86 \pm 0.99}$	93.37 ± 1.51	$\mathbf{94.37 \pm 4.88}$	$\mathbf{84.54 \pm 0.46}$	$\mathbf{92.33 \pm 0.61}$	$\mathbf{0.00 \pm 0.00}$
Raw mean activations	81.15 ± 3.68	97.31 ± 0.35	97.11 ± 2.05	95.02 ± 2.53	89.71 ± 8.43	80.57 ± 2.51	90.15 ± 1.00	-2.19 ± 1.16
Single-layer injection (L9)	77.96 ± 4.47	97.93 ± 0.94	94.86 ± 5.17	94.21 ± 3.83	76.61 ± 18.73	78.81 ± 0.26	86.73 ± 2.69	-5.60 ± 2.69
Fixed $\alpha = 1$	82.97 ± 0.67	98.35 ± 1.55	97.70 ± 1.00	$\mathbf{96.69 \pm 1.46}$	$\underline{91.56 \pm 5.52}$	$\underline{83.29 \pm 0.93}$	$\underline{91.76 \pm 1.01}$	$\underline{-0.57 \pm 0.71}$
Global shared ranking	$\underline{83.87 \pm 0.51}$	$\mathbf{98.96 \pm 0.73}$	$\underline{97.71 \pm 0.95}$	95.02 ± 2.53	90.61 ± 4.15	83.15 ± 1.58	91.56 ± 0.79	-0.78 ± 0.84
Non-accumulative ranking	77.70 ± 5.82	97.72 ± 1.92	$\mathbf{98.86 \pm 0.99}$	$\underline{95.83 \pm 5.20}$	33.31 ± 55.30	72.60 ± 8.11	79.34 ± 9.25	-13.00 ± 8.88

Table 4: GPT-J ablations across three seeds. Entries are mean accuracy (%) ± sample standard deviation. Avg. is the macro-average across tasks, and Δ Avg. is the paired per-seed change from the full method in percentage points. Bold and underline indicate the highest and second-highest means in each column, respectively.

Non-accumulative ranking is the most consequential of the tested choices. Fixing the task-level ordering after first-pass single-head removals reduces the average by 13.00 ± 8.88 points and yields particularly high variation on Country–Capital. Collapsing the selected contributions into a single layer reduces the average by 5.60 ± 2.69 points and lowers the mean accuracy on all six tasks, including a 6.33-point loss on Antonymy. Replacing matched displacements with raw means costs 2.19 ± 1.16 points overall and lowers the mean accuracy on five of the six tasks, consistent with the zero-shot subtraction removing query-specific content that should not be steered. Fixing $\alpha = 1$ produces a smaller 0.57 ± 0.71-point drop, indicating that magnitude tuning is useful but not responsible for the main result. The global ranking trails by 0.78 ± 0.84 points; a task-specific ordering remains best on average in this ablation.

6 CONCLUSION

We study function-vector head selection as a conditional rather than marginal attribution problem. CORE-FV uses backward elimination to repeatedly reassess each attention head in the context of the heads that remain, allowing the selected set to reflect redundancy, complementarity, and interference. Across five models and eight standard ICL transformations, this conditional ranking generally improves over the matched AIE baseline and achieves higher average accuracy than the reproduced SV and LTV baselines. The advantage becomes especially clear on composite tasks, while the composition experiments show that the resulting vectors can also retain useful algebraic structure. Ablations further support the role of conditional re-evaluation, with fixing the ranking after the initial removal pass causing the largest degradation among the tested variants.

Overall, the results suggest that how heads are selected matters as much as how their task representations are constructed: treating their contributions as context-dependent provides a simple and effective refinement of existing function-vector pipelines. The current method requires white-box access and repeated intervention-based evaluation, which limits its direct application to larger or closed models. Improving the efficiency of conditional selection and extending it to broader model scales are natural directions for future work.

AI USE STATEMENT

We used generative AI tools, including ChatGPT, to polish the writing, assist with implementing and debugging experiments, and suggest minor experimental refinements. No other required-disclosure uses apply to this work. All AI-assisted content was reviewed and checked by the authors for correctness and consistency with the intended methods and results. The authors take full responsibility for the final content of this work, including all text, claims, code, results, and other artifacts.

REFERENCES

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 135–168. PMLR, 2024. URL https://proceedings.mlr.press/v235/achtibat24a.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352, 2023. doi: 10.52202/075280-0719.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

William Held and Diyi Yang. Shapley head pruning: Identifying and removing interference in multilingual transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2416–2427, Dubrovnik, Croatia, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.177. URL https://aclanthology.org/2023.eacl-main.177/.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, 2023. doi: 10.18653/v1/2023.findings-emnlp.624.

Xinyan Hu, Kayo Yin, Michael I. Jordan, Jacob Steinhardt, and Lijie Chen. Understanding in-context learning of addition via activation subspaces. *arXiv preprint arXiv:2505.05145*, 2025. URL https://arxiv.org/abs/2505.05145.

Dongfang Li, Zhenyu Liu, Xinshuo Hu, Zetian Sun, Baotian Hu, and Min Zhang. In-context learning state vector with inner and momentum optimization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 7797–7820, 2024. doi: 10.52202/079017-0250.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32287–32307, 2024.

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

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, pp. 14014–14024, 2019.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL https://arxiv.org/abs/2209.11895.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 411–420, 2001.

Minh An Pham, Anton Segeler, Thomas Wiegand, Wojciech Samek, Sebastian Lapuschkin, Patrick Kahardipraja, and Reduan Achtibat. Fast & faithful function vectors. *arXiv preprint arXiv:2606.05079*, 2026. doi: 10.48550/arXiv.2606.05079. URL https://arxiv.org/abs/2606.05079.

Xiaoye Qu, Zengqi Yu, Dongrui Liu, Wei Wei, Daizong Liu, Jianfeng Dong, and Yu Cheng. Cooperative or competitive? understanding the interaction between attention heads from a game theory perspective. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14079–14099, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.688. URL https://aclanthology.org/2025.acl-long.688/.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

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.

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*, volume 33, pp. 12388–12401, 2020.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580/.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Haolin Yang, Hakaze Cho, Kaize Ding, and Naoya Inoue. Task vectors, learned not extracted: Performance gains and mechanistic insights. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RGEbVZgf4E.

Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72428–72461, 2025.

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. *arXiv preprint arXiv:2310.01405*, 2023.

A Implementation and Evaluation Details

Data splits. For each task, we deterministically divide the dataset into 60% training, 20% validation, and 20% test examples using a fixed random seed. Training data are used to construct task-conditioned head vectors and to learn the deletion-based head ordering. Validation data are used to select the number of heads K and intervention strength α. Test performance does not affect either selection.

In-context prompts. We use 10-shot ICL prompts throughout the experiments. A training prompt is considered successful when greedy generation from the unmodified model exactly matches the target output. Head displacements are estimated from 100 successful training prompts. The deletion-based refinement procedure uses up to 25 successful training queries, evaluated in their corresponding zero-shot form.

For every successful ICL prompt used to estimate the displacement vectors, we construct a matched zero-shot prompt containing the same query but no demonstrations. We record the final-token head activations for both prompts and average the difference $a_{\ell j}(p^{\mathrm{ICL}}) - a_{\ell j}(p^0)$ across successful examples.

Subset evaluation during refinement. Candidate head subsets are evaluated using gold target prefixes. For each target sequence, every token is scored conditioned on the ground-truth preceding target tokens, and the sequence is counted as correct only when every target token is the top-1 prediction at its corresponding position. The primary subset score is the fraction of completely correct examples. Mean target-token log-probability is used as a secondary score when two candidate subsets have the same correctness.

The deletion search itself uses the unscaled head vectors ($\alpha = 1$). The intervention strength is optimized only after the complete head ordering has been obtained.

Adaptive deletion schedule. At an iteration with k retained heads, all k possible single-head deletions are first evaluated. The resulting scores determine an ordering from most to least removable. To accelerate pruning, the initial proposed multi-head deletion size is the power of two nearest to $k/8$, capped below k. For example, the initial proposals for $k = 1024$, 512, 60, 32, and 16 are respectively 128, 64, 8, 4, and 2 heads.

The proposed group consists of the heads currently ranked as most removable. A group deletion is accepted when the resulting subset has sequence-level correctness no lower than that of the current subset. Otherwise the proposal size is repeatedly halved until a valid deletion is found or the group size reaches two. If no multi-head proposal satisfies the guard, the best single-head deletion is performed. The single-head fallback is not required to preserve correctness; this guarantees that the procedure eventually produces a complete ordering of all heads.

Head-count selection. We evaluate prefixes of the learned ranking with

$$K \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 40, 60\}, \tag{6}$$

excluding values that are not smaller than the total number of available heads. The complete all-head set is not treated as a model-selection candidate. Each value of K is paired with its own validation-tuned intervention strength.

Intervention-strength search. For each K, the search for α begins with the coarse grid

$$\alpha \in \{0.5, 1, 2, 4, 8, 16, 32\}.$$

The best coarse region is subsequently searched using two bounded refinement stages. The first uses up to 12 intervals and the second up to 4 intervals, with a minimum spacing of 0.125. Validation ties between intervention strengths are resolved in favor of the smaller α.

After optimizing α separately for each K, we select the configuration with the highest validation accuracy. Candidate values of K are processed in ascending order; consequently, equal validation accuracy is resolved in favor of the smaller head set.

Validation and test subsets. For validation and test evaluation, we restrict attention to queries that the unmodified model can answer correctly with a 10-shot ICL prompt. The demonstrations used for this filtering are drawn from the training split. The intervention is then evaluated on the corresponding zero-shot query. This filtering measures how much of the behavior available to the model under ICL can be recovered through the learned intervention.

Generation and scoring. Final performance is measured using greedy autoregressive generation. At every decoding step, the layer-specific steering vector is added to the active final-token representation at each layer containing selected heads. Generated text is truncated to the first line and stripped of surrounding whitespace before exact-match comparison with the target.

For the primary reported result, (K, α) is selected using validation data and subsequently frozen for test evaluation. We additionally evaluate the validation-tuned configuration associated with each candidate K on the test set to obtain accuracy-versus-head-count curves; these measurements are analysis-only and are not used for model selection.

B Extended Results

This appendix provides the supporting results needed to interpret and reproduce the experiments in Sections 4 and 5. We report the configurations selected in the main benchmark, disaggregate the head-count analysis by task, record the ablation configurations, and then give the extended composite-task and vector-composition results. We omit redundant validation traces and raw per-example outputs.

B.1 Main Benchmark Details

Model details. Table 5 gives the exact checkpoints and candidate spaces. For the grouped-query models, each query-head output is a separate candidate because it occupies a distinct slice of the attention output projection.

Model	Checkpoint	Params.	Layers	Q/KV heads	Candidates	Load dtype
GPT-J 6B	`EleutherAI/gpt-j-6b`	6.1B	28	16/16	448	FP16
Llama-3.1-8B	`meta-llama/Llama-3.1-8B`	8B	32	32/8	1024	FP16
Qwen3-4B-Base	`Qwen/Qwen3-4B-Base`	4B	36	32/8	1152	FP16

Table 5: Models used in the experiments. "Q/KV heads" gives the query and key–value heads per layer; "Candidates" is the number of query/output head slots ranked per task. Models are loaded in FP16 without quantisation.

Validation-selected configurations. Table 6 records the number of heads and intervention strength chosen before test evaluation for every run underlying Table 1. Each entry has the form K/α, and seed columns follow the order used throughout the paper. Selection is heterogeneous across tasks and seeds for both rankings; in particular, the advantage of CORE-FV cannot be attributed to systematically selecting larger prefixes.

		AIE			CORE-FV		
Model	Task	Seed 42	Seed 43	Seed 44	Seed 42	Seed 43	Seed 44
GPT-J 6B	Antonymy	20/1.5	30/1.5	15/2	60/1.25	60/1.625	40/1.625
	Capitalize	30/1.25	40/1.125	30/1	20/1.125	20/1.25	30/1.25
	Present–Past	40/1.375	40/1.375	40/1.375	15/2	30/2.125	60/1.625
	Singular–Plural	1/8.5	60/0.875	60/1.125	1/5	4/4.75	7/3.5
	Country–Capital	7/2	6/1.25	15/1.125	15/2.375	10/2	60/1.25
	English–French	60/1	60/1.125	60/1	40/1.375	30/1.25	30/1.5
Llama-3.1-8B	Antonymy	60/1	40/1	60/1.125	40/1.25	60/1.375	40/1.625
	Capitalize	9/1.5	10/1.375	9/1.25	30/1.125	60/1.125	30/1.125
	Present–Past	4/3.25	60/1	4/3	1/11	15/2.125	40/1.75
	Singular–Plural	30/1.75	8/3.25	7/2.75	3/4.25	10/2.125	10/2.875
	Country–Capital	20/1.875	30/1.625	15/3.25	20/3.25	20/2.375	40/1.25
	English–French	20/2.5	6/3.5	60/2.125	20/1.75	60/1.375	40/1.625
Qwen3-4B-Base	Antonymy	40/2.375	60/2.25	60/2	60/1.625	8/4.5	40/2.5
	Capitalize	10/4	15/2	30/1.625	7/2.75	20/2.125	30/1.75
	Present–Past	20/2.75	60/1.5	30/2.25	8/4.25	30/1.75	15/3.5
	Singular–Plural	20/1.5	15/3.25	60/1.5	20/4.25	30/2.75	20/3.5
	Country–Capital	60/1	60/1.375	20/2.5	5/2.75	10/2.75	40/1.375
	English–French	30/3.25	30/3.5	30/3.5	40/1.75	60/1.5	60/1.375

Table 6: Validation-selected configurations for the main benchmark. Each cell reports K/α; no test statistic is used in either choice. The corresponding exact-match test accuracies are in Table 1.

B.2 Accuracy as the Head Set Grows

Figure 1 evaluates every candidate K in Eq. 6. Each point uses the coefficient $\alpha_K^\star$ selected independently on validation data, after which the corresponding configuration is evaluated on the test split. The curves are therefore diagnostic: they do not alter the validation-selected result in Table 1, but reveal how task information accumulates as the head coalition expands.

On Llama-3.1-8B and Qwen3-4B-Base, the conditional ordering separates from AIE with only a few retained heads and preserves that advantage across the entire range. GPT-J behaves differently: the two curves become close once the prefix contains several heads, consistent with the smaller gap between their final selected configurations. Across models, gains diminish as K grows, showing that a substantial fraction of the recoverable task signal is concentrated near the front of the ordering. Furthermore, the curves are not strictly monotone: adding lower-ranked heads can introduce interfering displacements, and each value of K is paired with its own validation-selected intervention strength.

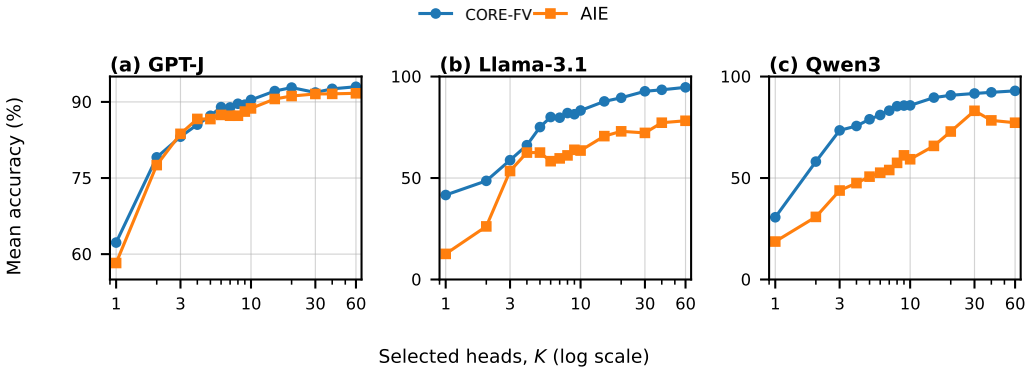

Figure 1: Mean exact-match test accuracy as a function of the number of selected heads. Each curve first macro-averages the six tasks within a seed and then averages across seeds 42, 43, and 44. For every K, the intervention strength is selected independently on the validation split. CORE-FV denotes the proposed method.

Task-level head-count curves. To explain the origins of these aggregate differences, Figures 2–4 disaggregate Figure 1 across individual tasks. The aggregate GPT-J curves converge because the conditional and marginal rankings largely select overlapping effective heads after the first few steps on most tasks. In contrast, on Llama-3.1-8B and Qwen3-4B-Base, the advantage of CORE-FV is heavily concentrated in Present–Past, Country–Capital, and especially English–French. The two rankings achieve comparable performance on Antonymy and eventually saturate on Capitalize and Singular–Plural. Thus, the macro-average improvement of CORE-FV reflects the avoidance of severe, task-specific failure modes on difficult transformations rather than an incremental, uniform gain across all tasks.

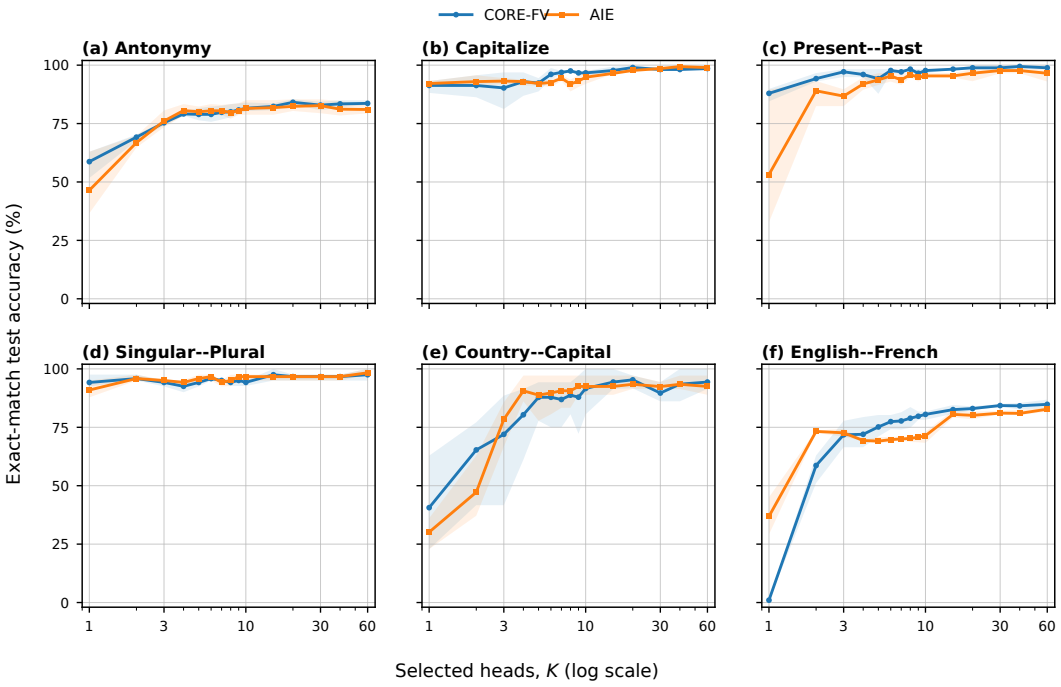

Figure 2: Task-level counterpart of Figure 1 for GPT-J 6B. Lines are means over seeds 42, 43, and 44; shaded regions span the observed range. For every ranking, seed, task, and K, α is selected independently on validation data before test evaluation.

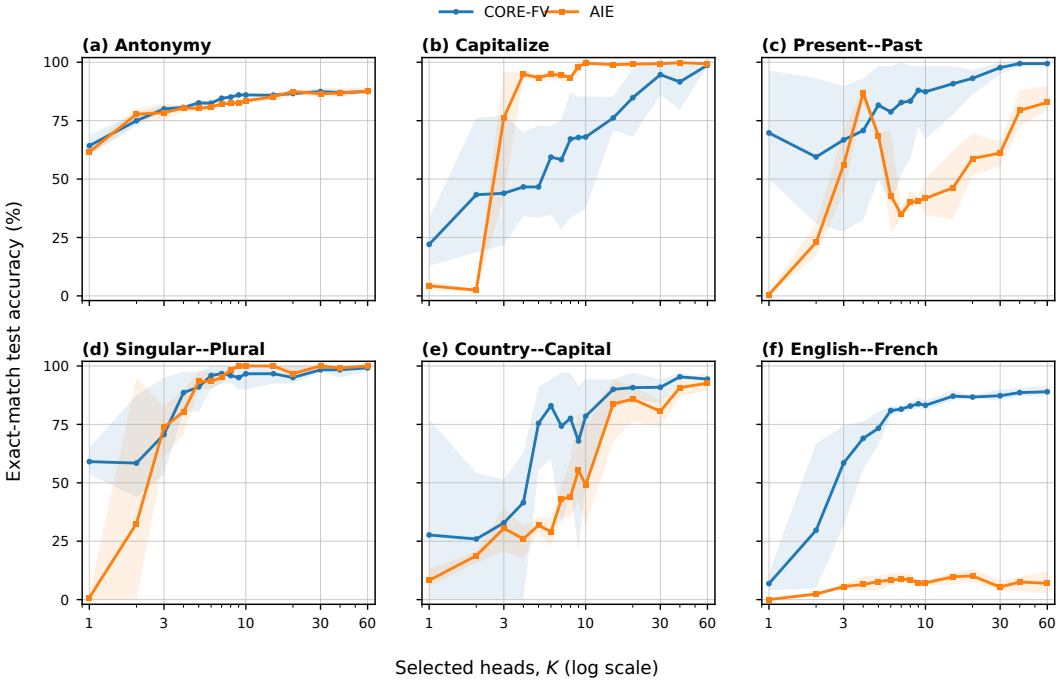

Figure 3: Task-level counterpart of Figure 1 for Llama-3.1-8B. Lines are means over seeds 42, 43, and 44; shaded regions span the observed range. All intervention strengths are selected on the validation split.

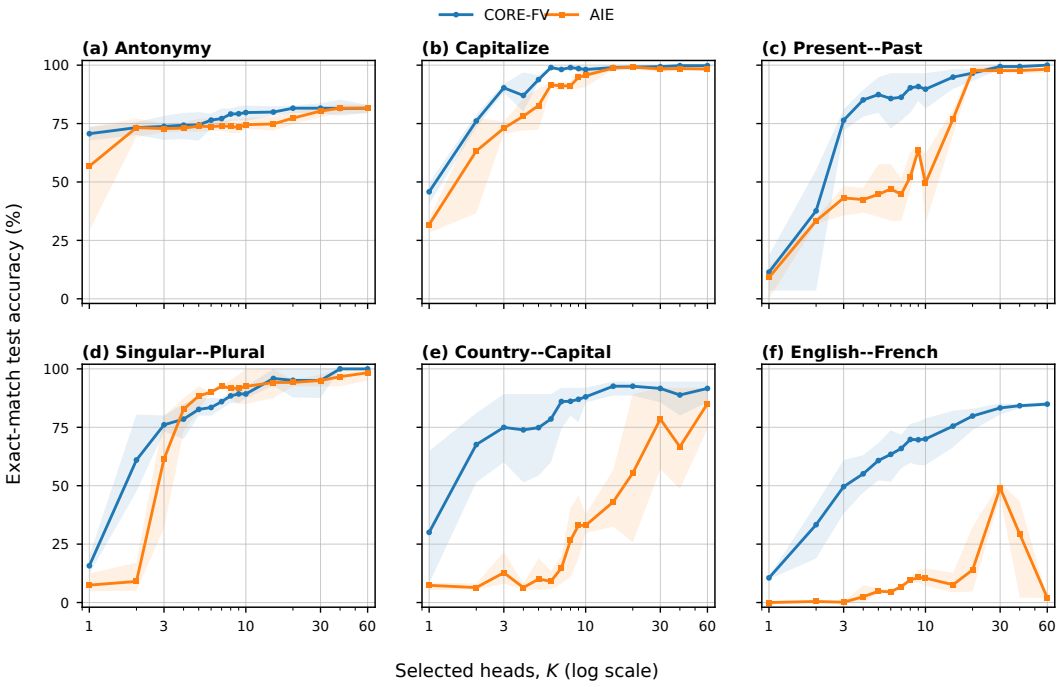

Figure 4: Task-level counterpart of Figure 1 for Qwen3-4B-Base. Lines are means over seeds 42, 43, and 44; shaded regions span the observed range. All intervention strengths are selected on the validation split.

B.3 Composite-Task Results

Selected configurations. Table 7 gives the validation-selected configuration for all three Llama-3.1-8B seeds. The two methods use the same eligible examples within each row. On Singularize → Capitalize, CORE-FV reaches 100% test accuracy with 15 heads in every seed, whereas AIE selects 80–100 heads.

		AIE			CORE-FV		
Task	Seed	K	α	Test	K	α	Test
Antonym → French	42	60	2.000	14.29	100	1.500	80.00
	43	100	2.000	13.51	100	1.000	70.27
	44	100	1.375	80.56	60	1.500	86.11
Antonym → French → Cap.	42	100	1.625	16.67	60	1.625	93.33
	43	70	1.750	12.90	90	1.375	74.19
	44	90	1.375	24.14	100	1.250	79.31
Next-item → Capitalize	42	70	1.500	65.31	60	1.750	87.76
	43	40	1.750	54.90	70	1.500	82.35
	44	40	1.750	44.90	40	2.000	81.63
Singularize → Capitalize	42	100	0.875	100.00	15	1.750	100.00
	43	80	0.875	100.00	15	2.375	100.00
	44	100	0.875	100.00	15	2.250	100.00
Singularize ⊕ Antonym	42	1	1.875	0.00	80	1.875	91.25
	43	8	1.000	0.00	70	1.875	91.36
	44	9	0.750	0.00	60	1.750	92.59

Table 7: Validation-selected configurations and exact-match test accuracy (%) for all three Llama-3.1-8B seeds.

Head-count curves. Figure 5 shows the 15 head counts evaluated for all five tasks in all three seeds ($K = 1, \ldots, 10, 15, 20, 30, 40, 60$). At $K = 20$–60, the macro-average ranges from 79.4% to 88.2% for CORE-FV and from 30.2% to 36.8% for AIE. Both rankings reach perfect accuracy on Singularize → Capitalize, but CORE-FV does so with fewer heads.

Pairwise interaction diagnostic. For an AIE-ranked Top-60 pool, we compute $I_{ij} = F(\{i, j\}) - F(\{i\}) - F(\{j\}) + F(\emptyset)$ on every unordered pair. Table 8 includes every available Top-60 diagnostic; each contributes $\binom{60}{2} = 1770$ pairs. Across the 12 task–seed pools, 81.0% of pairs have positive interactions. The distribution contains both signs.

Composite task	Seeds	$I_{ij} > 0$ (%)	$10^3\,\mathbb{E}[I_{ij}]$	$10^3\,\mathbb{E}[\lvert I_{ij}\rvert]$
Antonym → French	3	81.41 ± 5.40	0.671 ± 0.130	0.780 ± 0.092
Antonym → French → Cap.	3	84.43 ± 7.93	0.640 ± 0.108	0.689 ± 0.084
Next-item → Capitalize	2	71.50 ± 0.65	0.256 ± 0.033	0.299 ± 0.038
Singularize → Capitalize	2	81.61 ± 0.54	1.747 ± 0.011	2.127 ± 0.112
Singularize ⊕ Antonym	2	84.27 ± 4.15	2.451 ± 0.115	2.565 ± 0.101

Table 8: Descriptive AIE Top-60 pair-interaction statistics, reported as mean ± population standard deviation over the indicated seeds. Interactions are measured on held-out prompts and never used for selection.

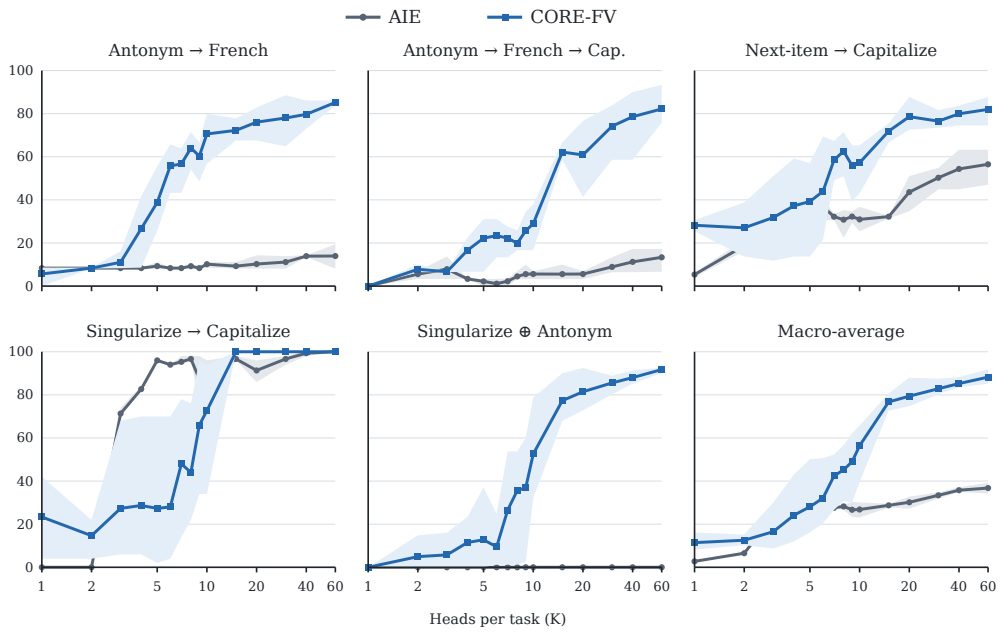

Figure 5: Composite-task exact-match test accuracy over the head-count grid shared by all three Llama-3.1-8B seeds. Lines show seed means and shaded regions show the observed range. For each ranking and K, α is selected independently on validation data.

B.4 Arithmetic Add-k Benchmark

Motivation and task definition. While standard in-context learning benchmarks predominantly assess factual retrieval and lexical associations, arithmetic transformations provide a controlled testbed for evaluating whether extracted function vectors capture systematic algorithmic operations. Inspired by Hu et al. (2025), we evaluate an arithmetic Add-k benchmark consisting of five distinct tasks: `number-add2`, `number-add4`, `number-add5`, `number-add10`, and `number-add20`. For each task, the objective is to execute the deterministic mapping $f_k(x) = x + k$ for a query integer x. Each task dataset contains 500 consecutive integer pairs with inputs $x \in \{1, 2, \ldots, 500\}$ and ground-truth targets $y = x + k$, deterministically partitioned into 60% train, 20% validation, and 20% test sets.

Experimental setup. Prompts follow the standard question–answer template: `Q: {x}\nA: {x + k}\n\n` with 5-shot ICL demonstrations. Following our standard protocol, test queries are filtered to instances solved correctly by the base model under 5-shot ICL. All methods (CORE-FV, AIE, and LTV) are evaluated on GPT-J-6B and Llama-3.1-8B using the held-out test split across three random seeds (42, 43, 44), scored by exact match on the first generated line.

Results and discussion. Table 9 reports test accuracy across all tasks and models. Across both architectures, CORE-FV achieves higher average accuracy than marginal AIE selection, improving the 5-task average from 55.68% to 84.41% on GPT-J-6B and from 87.07% to 97.38% on Llama-3.1-8B. The difference is most notable on `number-add20`, where AIE drops to 45.81% on Llama-3.1-8B while CORE-FV reaches 95.98%. This indicates that larger arithmetic offsets rely more heavily on heads acting in combination, where conditional selection can capture joint contributions that independent marginal scoring tends to overlook.

When compared against continuous learned task vectors (LTV), CORE-FV achieves a higher average on GPT-J-6B (84.41% vs. 83.29%). On Llama-3.1-8B, where the base model already performs near ceiling under clean ICL (97.60%), gradient-based optimization allows LTV to reach 100.00%, while CORE-FV closely follows at 97.38% without updating any vector parameters.

Task	Clean 5-shot ICL	AIE	LTV	CORE-FV (Ours)
GPT-J-6B				
number-add2	93.67±3.21	89.35±4.69	<u>91.43±2.03</u>	**97.88±1.81**
number-add4	39.00±6.00	<u>83.13±6.92</u>	63.76±11.31	**94.20±2.45**
number-add5	35.33±5.13	35.50±6.34	<u>88.53±1.59</u>	**91.58±5.82**
number-add10	48.33±4.04	47.62±41.44	**91.88±3.52**	<u>85.07±10.35</u>
number-add20	19.33±2.52	22.82±15.79	**80.87±12.47**	<u>53.34±9.73</u>
Average	47.13	55.68	<u>83.29</u>	**84.41**
Llama-3.1-8B				
number-add2	98.67±0.58	99.33±1.17	**100.00±0.00**	**100.00±0.00**
number-add4	99.33±0.58	96.27±3.87	**100.00±0.00**	<u>98.32±2.10</u>
number-add5	99.67±0.58	<u>97.31±3.83</u>	**100.00±0.00**	95.32±4.51
number-add10	98.67±1.15	96.63±2.11	**100.00±0.00**	<u>97.28±4.71</u>
number-add20	91.67±0.58	45.81±7.15	**100.00±0.00**	<u>95.98±5.20</u>
Average	97.60	87.07	**100.00**	<u>97.38</u>

Table 9: Exact-match test accuracy (%) on the arithmetic Add-k benchmark across random seeds 42, 43, and 44. Bold and underline indicate the best and second-best intervention methods within each task and model, respectively. Clean 5-shot ICL is provided as an unsteered reference.

	CORE-FV		Task AIE		Global AIE	
K	Component α	Refined α	Component α	Refined α	Component α	Refined α
3	**31.19**	53.00	24.00	50.88	24.38	**57.62**
5	**34.56**	**56.94**	22.50	53.56	22.56	55.12
8	**38.25**	62.19	25.94	62.62	29.12	**64.94**
10	**54.44**	64.56	24.88	63.94	28.62	**66.38**

Table 10: Eight-task macro-average composition accuracy (%) across the complete seed-42 K grid. Bold marks the best ranking separately for each coefficient protocol and K.

B.5 Vector-Composition Results

The composition study combines a complete GPT-J seed-42 ranking-by-K matrix with $K = 10$ CORE-FV replications for seeds 43 and 44. Each of the 192 rows in the matrix corresponds to one of eight targets, three rankings, two coefficient protocols, and four values of K. The replications add both coefficient protocols for all eight targets, giving three complete seeds for the CORE-FV comparison reported in the main paper. Every accuracy is measured on the same target-specific held-out set for the configurations compared within a seed.

Full task-level curves. Figure 6 visualizes the complete seed-42 matrix. Refining the three coefficients consistently closes the gap between rankings, but its effect depends strongly on the target. The conditional ranking is much more effective without target-level tuning on Present–Past, Singular–Plural, and Product–Company, while the AIE rankings can recover after refinement.

Macro-averages over K. Table 10 macro-averages the eight seed-42 tasks for each ranking, coefficient protocol, and K. Without target-level coefficient refinement, CORE-FV is best at every K and its advantage widens with larger prefixes. With refinement, the spread across rankings is at most 6.74 points and falls to 2.44 points at $K = 10$.

Three-seed CORE-FV replication at $K = 10$. Figure 7 and Table 11 report the complete seed-specific CORE-FV results. Refinement improves 23 of the 24 task–seed evaluations; the only decrease is one point on Last-Capitalize for seed 43. At the macro level, the component-α averages are 54.44%, 32.00%, and 43.31% for seeds 42, 43, and 44, while the refined-α averages are 64.56%, 64.00%, and 64.94%. The resulting three-seed estimates are $43.25 \pm 9.16\%$ and $64.50 \pm 0.39\%$,

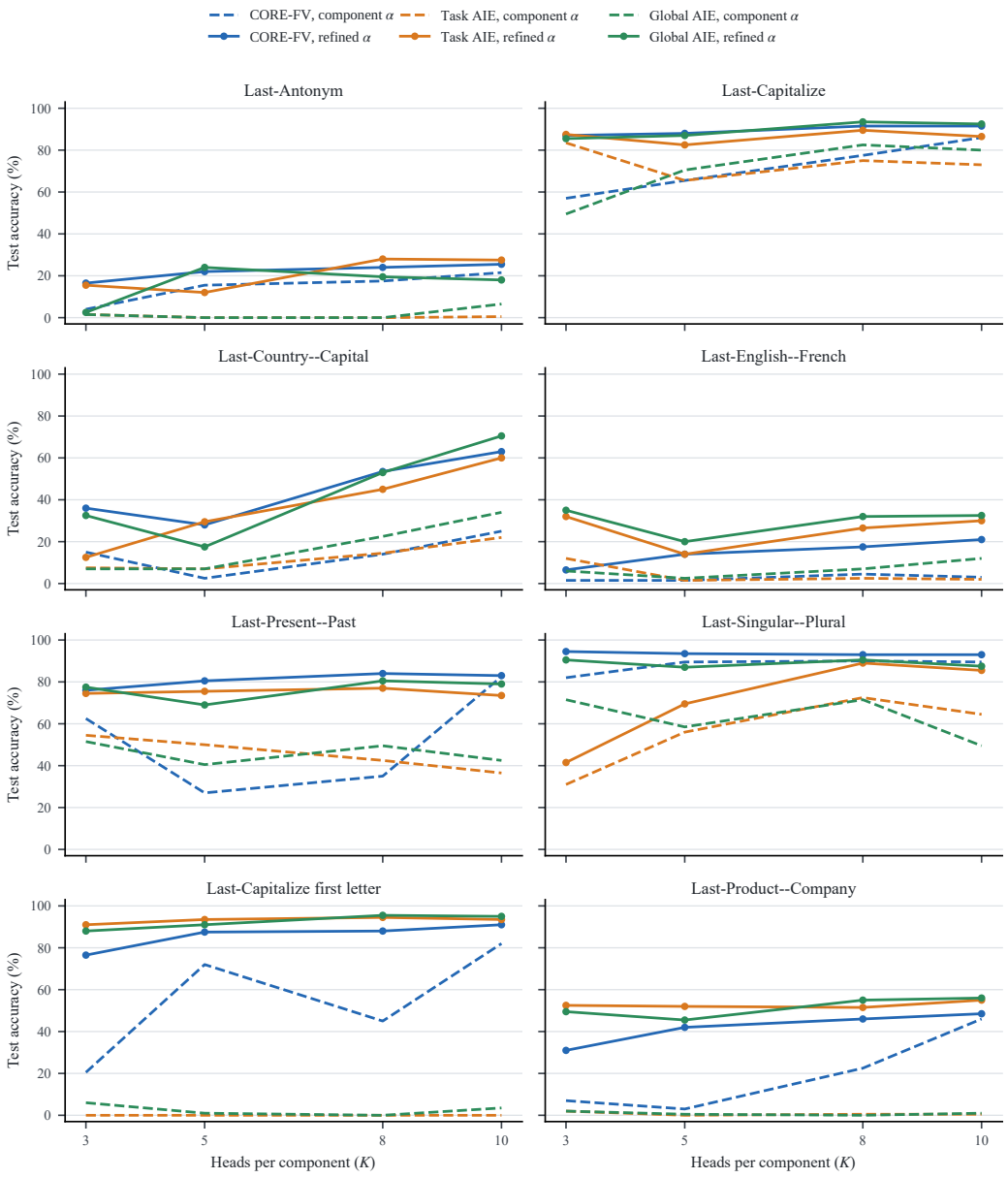

Figure 6: All 192 vector-composition test accuracies in the complete seed-42 matrix. Dashed lines combine coefficients selected independently on the three component tasks; solid lines make one validation-only coordinate-refinement pass on the composed target. Each panel uses GPT-J.

respectively, showing that target-level refinement substantially reduces cross-seed variation in the eight-task average.

Task-level ranking comparison at $K = 10$. The complete ranking comparison is available for seed 42, for which all three rankings and all four values of K were evaluated. Table 12 expands the corresponding macro-average. CORE-FV is best on two targets after refinement; global AIE is best on five and task AIE on one. In contrast, with component-only coefficients, CORE-FV is best on six of eight targets and has the highest macro-average by 25.81 points. The comparison therefore separates two claims: conditional selection helps produce directly composable vectors, but validation tuning on the final composition can partly substitute for that property.

Coefficients selected for the proposed ranking. Table 13 records the $K = 10$ coefficients used for the two CORE-FV columns in Table 3. The initial tuple combines three independently selected component coefficients; the second tuple is obtained by the single validation-only pass described in Section 4.4. The variation across both targets and seeds explains why a single global rescaling would not reproduce the refinement gains.

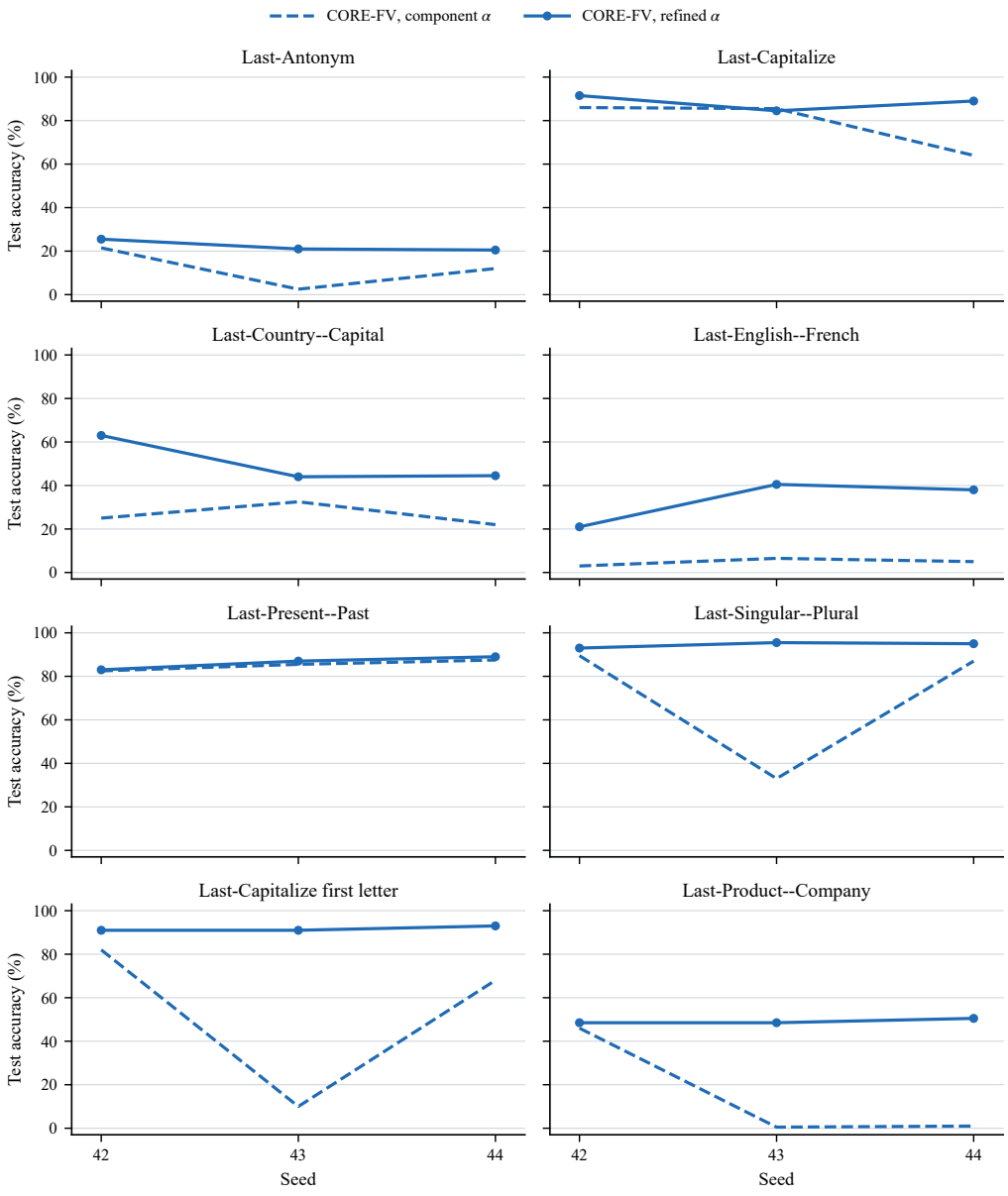

Figure 7: Seed-specific GPT-J CORE-FV composition accuracy at $K = 10$ heads per component. Dashed lines combine the validation-selected component coefficients directly; solid lines make one validation-only coordinate-refinement pass on the composed target. All eight targets are complete for seeds 42, 43, and 44.

	Seed 42		Seed 43		Seed 44		Mean ± population s.d.	
Last-T task	Component	Refined	Component	Refined	Component	Refined	Component	Refined
Antonym	21.5	25.5	2.5	21.0	12.0	20.5	12.00±7.76	22.33±2.25
Capitalize	86.0	91.5	85.5	84.5	64.0	89.0	78.50±10.26	88.33±2.90
Country–Capital	25.0	63.0	32.5	44.0	22.0	44.5	26.50±4.42	50.50±8.84
English–French	3.0	21.0	6.5	40.5	5.0	38.0	4.83±1.43	33.17±8.66
Present–Past	82.5	83.0	85.5	87.0	87.5	89.0	85.17±2.05	86.33±2.49
Singular–Plural	89.5	93.0	33.0	95.5	87.0	95.0	69.83±26.07	94.50±1.08
Capitalize first letter	82.0	91.0	10.0	91.0	68.0	93.0	53.33±31.17	91.67±0.94
Product–Company	46.0	48.5	0.5	48.5	1.0	50.5	15.83±21.33	49.17±0.94
Average	54.44	64.56	32.00	64.00	43.31	64.94	43.25±9.16	64.50±0.39

Table 11: Task-level GPT-J CORE-FV composition accuracy (%) at $K = 10$ heads per component for all three seeds. The final two columns report the mean and population standard deviation across seeds.

	CORE-FV		Task AIE		Global AIE	
Last-T task	Component	Refined	Component	Refined	Component	Refined
Antonym	21.5	25.5	0.5	**27.5**	6.5	18.0
Capitalize	86.0	91.5	73.0	86.5	80.0	**92.5**
Country–Capital	25.0	63.0	22.0	60.0	34.0	**70.5**
English–French	3.0	21.0	2.0	30.0	12.0	**32.5**
Present–Past	82.5	**83.0**	36.5	73.5	42.5	79.0
Singular–Plural	89.5	**93.0**	64.5	85.5	49.5	87.5
Capitalize first letter	82.0	91.0	0.0	93.5	3.5	**95.0**
Product–Company	46.0	48.5	0.5	55.0	1.0	**56.0**
Average	54.44	64.56	24.88	63.94	28.62	**66.38**

Table 12: Task-level GPT-J composition accuracy (%) at $K = 10$ heads per component for the complete seed-42 ranking comparison. Bold marks the best of the six configurations in each row.

Last-T task	Seed	Component-optimal $(\alpha_T, \alpha_F, \alpha_L)$	Refined $(\alpha_T, \alpha_F, \alpha_L)$
Antonym	42	$(2.00, 1.13, 3.00)$	$(2.38, 1.27, 2.94)$
Antonym	43	$(2.25, 1.00, 4.00)$	$(4.36, 2.00, 2.38)$
Antonym	44	$(2.88, 1.13, 3.50)$	$(3.95, 1.97, 2.84)$
Capitalize	42	$(1.75, 1.13, 3.00)$	$(0.88, 1.55, 2.94)$
Capitalize	43	$(1.38, 1.00, 4.00)$	$(1.44, 0.88, 4.00)$
Capitalize	44	$(3.00, 1.13, 3.50)$	$(1.50, 2.11, 2.99)$
Country–Capital	42	$(3.50, 1.13, 3.00)$	$(2.33, 2.25, 2.94)$
Country–Capital	43	$(2.00, 1.00, 4.00)$	$(2.63, 2.00, 4.25)$
Country–Capital	44	$(3.00, 1.13, 3.50)$	$(5.16, 2.25, 2.33)$
English–French	42	$(2.38, 1.13, 3.00)$	$(3.27, 1.55, 2.13)$
English–French	43	$(1.13, 1.00, 4.00)$	$(2.25, 1.50, 2.38)$
English–French	44	$(1.63, 1.13, 3.50)$	$(3.25, 2.25, 2.33)$
Present–Past	42	$(2.50, 1.13, 3.00)$	$(2.50, 1.41, 3.00)$
Present–Past	43	$(1.75, 1.00, 4.00)$	$(2.19, 1.00, 4.00)$
Present–Past	44	$(3.00, 1.13, 3.50)$	$(2.25, 1.97, 3.65)$
Singular–Plural	42	$(1.75, 1.13, 3.00)$	$(1.81, 1.83, 1.88)$
Singular–Plural	43	$(1.38, 1.00, 4.00)$	$(2.75, 1.75, 4.00)$
Singular–Plural	44	$(1.50, 1.13, 3.50)$	$(2.02, 2.11, 1.97)$
Capitalize first letter	42	$(2.13, 1.13, 3.00)$	$(2.13, 1.97, 2.19)$
Capitalize first letter	43	$(1.13, 1.00, 4.00)$	$(2.25, 1.63, 2.50)$
Capitalize first letter	44	$(1.00, 1.13, 3.50)$	$(1.38, 2.25, 3.28)$
Product–Company	42	$(3.50, 1.13, 3.00)$	$(2.77, 1.97, 3.00)$
Product–Company	43	$(2.00, 1.00, 4.00)$	$(4.00, 1.88, 2.00)$
Product–Company	44	$(1.38, 1.13, 3.50)$	$(2.75, 2.25, 2.00)$

Table 13: Validation-selected coefficient tuples for CORE-FV at $K = 10$ for all three seeds. Here α_T scales First-T, α_F scales the subtracted First-Copy vector, and α_L scales Last-Copy. Values are rounded to two decimals for readability.

C Ablation Details

All ablation variants use the same data splits, training-only estimation, validation selection, and held-out test protocol as the full method.

Raw mean. This variant uses the success-filtered few-shot mean $\mathbb{E}[a^{\mathrm{ICL}}]$ in place of the matched displacement $\mathbb{E}[a^{\mathrm{ICL}} - a^{0}]$. The remaining selection and intervention procedure is unchanged.

Single layer. The selected projected contributions are summed and injected once at GPT-J layer 9 rather than at their native layers. Layer 9 follows the original Function Vectors setup (Todd et al., 2024).

Fixed strength. This variant removes magnitude tuning by fixing $\alpha = 1$ and selecting only the number of heads K on the validation split.

Global ranking. A single training-only head ordering is constructed from the pooled tasks and shared across all six evaluations. The displacement vectors remain task-specific.

Non-accumulative ranking. Each head is scored once by its single-head removal effect from the initial model state, and this first-pass ordering is then kept fixed. Unlike the full method, removal effects are not recomputed after successive deletions.

Validation-selected configurations. Table 14 reports the selected K/α configuration for every task and seed. The fixed-strength variant has $\alpha = 1$ by construction; all other configurations result from the common validation search.

Task	CORE-FV	Raw mean	Single layer (L9)	Fixed $\alpha = 1$	Global ranking	Non-accum. ranking
			Seed 42			
Antonymy	60/1.25	15/1.875	15/2.375	60/1	30/1.75	60/1.375
Capitalize	20/1.125	15/1.75	10/1.25	15/1	7/2.375	15/1.125
Present–Past	15/2	60/1.375	10/1.75	60/1	15/3	60/1.25
Singular–Plural	1/5	1/5.5	15/1.875	30/1	1/5	1/8.5
Country–Capital	15/2.375	20/1.625	40/1.5	60/1	60/1.625	1/0.5
English–French	40/1.375	40/1.125	40/1	60/1	60/1.125	60/1.25
			Seed 43			
Antonymy	60/1.625	20/2.5	20/2.875	60/1	60/1.25	10/2.375
Capitalize	20/1.25	8/2.125	40/1.125	30/1	60/1.125	60/0.875
Present–Past	30/2.125	15/2.125	2/3.5	60/1	40/1.25	30/1.25
Singular–Plural	4/4.75	2/3.5	20/2	20/1	9/1.75	40/1.25
Country–Capital	10/2	8/3	3/3.75	20/1	30/1	30/1.125
English–French	30/1.25	7/1.75	20/1	60/1	30/1.25	6/1.75
			Seed 44			
Antonymy	40/1.625	20/1.875	20/2.25	60/1	30/1.75	30/1.75
Capitalize	30/1.25	60/1.75	60/1.25	60/1	30/1.25	30/2.25
Present–Past	60/1.625	15/3	20/2.25	60/1	40/1.75	40/1.375
Singular–Plural	7/3.5	2/5.5	3/3.25	40/1	30/1.375	15/2.625
Country–Capital	60/1.25	20/1.875	20/2.5	60/1	20/1.75	1/0.5
English–French	30/1.5	60/1.625	20/2	60/1	60/0.75	20/2.125

Table 14: Validation-selected configurations for the GPT-J ablations. Each cell reports K/α; the corresponding accuracies are reported in Table 4.