EXTRACTING AND COMPOSING FUNCTION VECTORS FOR ANALOGICAL REASONING OVER LATENT RELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Analogical reasoning requires models to abstract relational structure and transfer it across domains, yet how large language models internally represent and manipulate such relations remains poorly understood. Recent work on mechanistic interpretability suggests that in-context tasks can be compressed into compact activation-space representations, or *function vectors*, but their capacity to support compositional and out-of-distribution reasoning has received limited study. We introduce a framework for learning, composing, and refining relation-specific function vectors for analogical reasoning. First, we optimize function vectors using as few as 10 relational examples while keeping the language model frozen. These fine-tuned function vectors substantially improve zero-shot relation completion and exhibit stronger alignment with human judgments of relational similarity. Next, we show that previously unseen relational concepts can be represented as weighted linear combinations of learned function vectors, enabling transfer to novel analogies. Finally, we introduce Analogical Function Vectors (AFVs), which refine these composition weights at inference time using a relation-consistency objective between source and target domains. Injecting AFVs into model activations improves performance on four-term analogies involving out-of-distribution relations not used to derive the function vectors, SAT analogy problems, and narrative-context analogy tasks across models ranging from GPT-2 to LLaMA-3.1-8B. Our results suggest that relational knowledge in language models can be represented and manipulated through compact activation-space structures, providing a parameter-efficient and mechanistically interpretable approach to improving analogical reasoning.

1 INTRODUCTION

A hallmark of human intelligence is the ability to abstract relations between individual entities to draw analogies across disparate domains (Holyoak, 2025). From childhood metaphors (Gentner, 1977) to complex conceptual mappings, this capacity allows for robust structural inference without relying on surface-level features or explicit relational phrases. Analogical reasoning, classically evaluated via four-term proportional analogies like *blindness : sight :: poverty : wealth*, remains a fundamental metric of cognitive ability (Gray and Thompson, 2004).

As outlined by Spearman (1923), solving such analogies requires inferring the relation of the source pair and projecting it onto the target domain, even though the relevant relation is not explicitly stated in the problem. Psychological studies suggest that humans solve analogies through relation extraction and refinement, in which representations of the source and target are updated to satisfy the constraints of the broader analogy (Hofstadter and Mitchell, 1995; Spellman and Holyoak, 1996; Yan et al., 2013). For example, completing "blindness : sight :: poverty : *wealth*" may initially evoke a broad relation of semantic opposition (*opposite of*) from the source pair, which is then refined to the more specific relation (*lack of*) in light of the target analogy. This interplay of relation extraction and refinement remains central to analogical reasoning in humans (Mitchell, 2021) and provides an important benchmark for representation learning in artificial systems.

Large Language Models (LLMs) represent the frontier of artificial systems attempting to meet this cognitive benchmark (Topsakal and Akinci, 2023; Yan et al., 2024). Yet, their capacity for robust

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.

analogical inference remains debated. Some studies indicate that LLMs demonstrate emerging analogical reasoning (Webb et al., 2023; 2025), while other studies indicate that their inferences remain brittle with errors unlike those from humans (Mitchell and Krakauer, 2023; Lewis and Mitchell, 2024). However, this literature relies on black-box behavioral testing; one exception (Lee et al., 2026) transfers relational information through activation patching, yet does not yield reusable representations. Despite the central role of relational representation in human analogical reasoning, it remains an open question how relational knowledge is structurally encoded and adaptively refined within the internal representations of LLMs.

To investigate these internal representations, we focus on LLMs' ability to perform in-context learning (Brown et al., 2020; Garg et al., 2022; Furuya et al., 2024; Dong et al., 2024), a mechanism that allows LLMs to solve analogy problems "on the fly" by generalizing from a few prompt demonstrations without updating their parametric weights. However, whereas humans can consolidate relational knowledge acquired through past experience and reuse it to support subsequent analogical inference (Gick and Holyoak, 1983; Holyoak et al., 2024), standard ICL is inherently transient. The relational knowledge elicited within a forward pass does not persist beyond the current context, limiting its direct reuse and transfer to novel situations. In contrast, human relational knowledge appears to be organized around reusable representations of semantic relations. Psychological evidence suggests that humans possess core knowledge of basic semantic relations, with graded structure that guides relational similarity judgments (Chaffin, 2012; Lu et al., 2019), while neurobiological evidence indicates that such relations are represented by distinct, distributed patterns of neural activity (Chiang et al., 2021).

To bridge this gap between transient in-context representations and persistent relational knowledge, we extend the *function vector* (FV) framework, which captures task-relevant knowledge as compact internal representations identified through causal mediation analysis. First, we show that optimizing these function vectors with minimal supervision (at least 10 related examples per relation) significantly improves zero-shot generalization on relational tasks. Second, we represent novel, untrained relations as weighted linear combinations of these base vectors, with each weight determined by the model's evidence that the corresponding base relation accounts for the source pair. Finally, we refine these composition weights at inference time, first to better capture the source relation and then to enforce relational consistency between the source and target domains. The resulting *analogical function vectors* (AFVs) provide a steerable, parameter-efficient mechanism for improving performance on both structured four-term analogies and narrative-level analogy tasks.

2 BACKGROUND

Activation Steering and Function Vectors. Activation steering adds behavior-specific vectors to a model's internal activations during inference (Turner et al., 2023; Subramani et al., 2022; Jorgensen et al., 2023; Yin et al., 2024). Todd et al. (2024) demonstrated that compact task representations, termed *function vectors*, can be localized to a small subset of attention heads identified through causal mediation analysis. By patching mean activations from an intact, task-consistent prompt into an uninformative (shuffled) context, we measure the subsequent increase in target prediction probability to compute each head's Causal Indirect Effect (CIE). The function vector is then constructed by summing the mean activations of the top attention heads (typically 10) with the highest CIEs. This extracted representation can then be injected directly into the LLM's hidden states during zero-shot inference to explicitly trigger the target relation for a novel query.

Linear Representation Hypothesis. The linear representation hypothesis states that high-level concepts are represented as linear directions in the model's representational space (Elhage et al., 2021). This hypothesis is supported by linear probes, causal mediation analysis, and concept edits through targeted projections (Mikolov et al., 2013b; Elhage et al., 2022; Meng et al., 2022). Such representations can be formulated under a causal inner product, where separable concepts in a unified space can be represented as orthogonal vectors (Park et al., 2024). These vectors can guide the model toward the target domain's concept (Zou et al., 2023) when added into the LLM's residual stream. In addition, if some basis concept representations can be combined linearly to form other concepts, this representation space can be generalized to inferences with out-of-distribution concepts.

3 METHODS

Building on psychological evidence that humans possess core knowledge of basic semantic relations (Chaffin, 2012; Lu et al., 2019), we hypothesize that a small set of base relations can be mechanistically isolated within Large Language Models (LLMs) as function vectors (Todd et al., 2024), and that these relation-specific vectors can be composed to represent previously unseen relations. Our framework proceeds in three stages. First, we optimize causally initialized function vectors for base relations using limited additional supervision, improving relation-completion performance and yielding representations that better align with human relational-similarity structure (Section 3.1). Second, we treat the resulting library of relation-specific vectors as a compositional basis, constructing activation-space representations for novel relations that are not explicitly represented in the basis itself (Section 3.2). Third, we refine these composite representations at test time in the context of a full analogy, allowing the relation representation to adapt to source-target constraints and, in some cases, further improve analogy performance (Section 3.3).

3.1 FINE-TUNED FUNCTION VECTOR

Following Todd et al. (2024), we initially derive a function vector for a specific relation z by contrasting the model's activations on informative versus uninformative in-context demonstrations. We then optimize this vector $\mathbf{v_z}$ using a supervision set of at least 10 examples that instantiate the target relation (Figure 1). Given a query x from this set, $\mathbf{v_z}$ is injected into an intermediate layer and trained to minimize the objective:

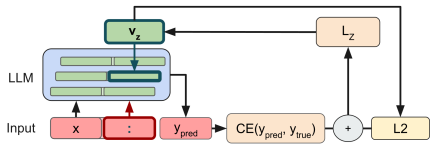


Figure 1: Illustration of the training procedure of the fine-tuned function vectors.

$$\mathcal{L}_z = \mathrm{CE}(y_{pred}, y_{true}) + \frac{\lambda}{2}||\mathbf{v_z}||_2^2 \tag{1}$$

where the cross-entropy compares the predicted and ground-truth distributions (y_{pred}, y_{true}) and the second term is an L2 penalty on the FV to prevent overfitting. The penalty is implemented as decoupled weight decay in AdamW (Loshchilov and Hutter, 2019) with $\lambda = 0.01$. Because backpropagation only updates the low-dimensional function vector while preserving the LLM's pre-trained weights, the optimization is highly parameter-efficient.

3.2 COMPOSITE VECTOR

To construct representations of novel, untrained relations encountered in analogy problems, we follow the linear representation hypothesis (Mikolov et al., 2013b; Elhage et al., 2022; Park et al., 2024) by modeling the source relation as a Composite Vector, computed as a weighted sum of the fine-tuned function vectors (FFVs) as illustrated in Figure 2. To construct this, we derive the weights for relations Z given a source analog (x_S, y_S) (e.g., *aspirin : pain*) using the LLM's predictive distribution under intervention. Specifically, for each base relation $z \in Z$, we inject its corresponding FFV $\mathbf{v_z}$ into the forward pass and record the log-probability $s_z = \log P(y_S|x_S, z)$ of generating the correct answer y_S given the query x_S. We then use a temperature-scaled softmax across the base relations to convert these log-probabilities into normalized relation weights $P(z|x_S, y_S)$, which quantify the relative evidence that each base relation accounts for the source pair. These normalized weights serve as the coefficients used to construct the composite vector $\mathbf{v}$:

$$P(z \mid x_S, y_S) = \frac{\exp(s_z/\tau)}{\sum_{z' \in Z} \exp(s_{z'}/\tau)}, \qquad \mathbf{v} = \sum_{z \in Z} P(z \mid x_S, y_S)\, \mathbf{v_z}. \tag{2}$$

When $\tau = 1$, $P(z \mid x_S, y_S)$ is the normalized likelihood over the base relations under a uniform prior. The resulting composite vector thus emphasizes the contributions of highly relevant base relations while attenuating irrelevant ones, yielding a tailored relational representation for solving the analogy.

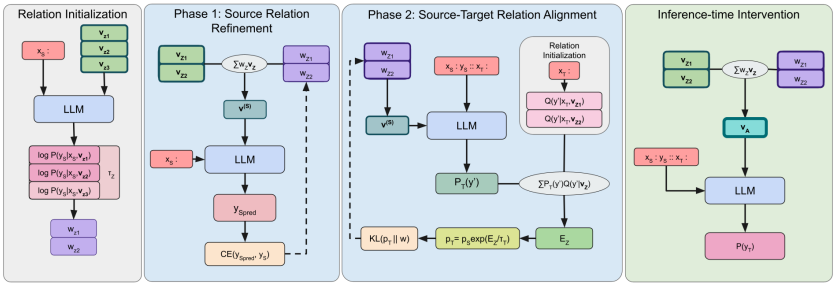


Figure 2: Illustration of the composite vector $\mathbf{v}$ and analogical function vector (AFV) for a four-term analogy problem (e.g., $x_S : y_S :: x_T : ?$). The composite vector's weights are set up through relation initialization (grey box) while the AFV's weights go through the two-phase refinement process (blue boxes) before intervention at inference time.

3.3 Analogical Function Vector

In analogical reasoning, relation consistency between the source and target is a fundamental constraint guiding human inference (Holyoak, 2012; Gentner and Toupin, 1986; Gentner and Markman, 1997). Furthermore, relational structures in both domains are updated to satisfy the analogy and its constraints (Hofstadter and Mitchell, 1995; Spellman and Holyoak, 1996; Yan et al., 2013). To emulate this, we introduce the *Analogical Function Vector* (AFV) that dynamically refines the composition weights initially derived from the source domain to facilitate robust cross-domain mapping. Developing the composition weights for the AFV involves a two-phase optimization process: source relation refinement, followed by source-target relation alignment.

Source Relation Refinement. We initialize the composition weights $w = \mathrm{softmax}(\theta)$, where $\theta_z = s_z/\tau$ is the scaled log-probability from injecting the fine-tuned function vector $\mathbf{v_z}$ and a softmax temperature τ that can optionally be used to emphasize the relevant relations. θ are the trainable logits; all refinement updates θ, and w is the resulting mixture. From there, we prune the irrelevant relations ($w_z < 0.001$) so that the refinement phases can only focus on the relations supported by the source analog. Finally, with the appropriate relations and weights we construct the composite vector $\mathbf{v}$. By injecting $\mathbf{v}$ into the LLM and prompting it with the source query x_S, we obtain a predicted distribution y_{SPred}. We then compute the cross-entropy loss $\mathrm{CE}(y_{SPred}, y_S)$ to update the logits θ and hence the composition weights w_z. As the initialization and objective share the same source-refinement likelihood, this phase aligns the intermediate vector towards the relations most capable of reconstructing the source analog with the prior $p_S = \mathrm{softmax}(\theta^{(S)})$.

Source-Target Relation Alignment. Because a single word pair can instantiate multiple valid relations, a second phase enforces relation consistency between the source and target domains. We treat the source-refined vector from Phase 1 $v^{(S)}$ as a relational prior and prompt the vector-injected LLM with the full analogy context (e.g., $(x_S, y_S), x_T$) to yield a predicted target distribution $P_T(y')$. In parallel, we perform relation initialization on the target query alone: injecting each base vector $\mathbf{v}_z$ with the prompt x_T gives a relation-specific prediction $Q(y' \mid x_T, \mathbf{v}_z)$. Each relation is then scored by how well it accounts for the analogy-conditioned prediction through evidence term $E_z = \sum_{y' \in V} P_T(y') \log Q(y' \mid x_T, \mathbf{v}_z)$, the expected log-likelihood of the target distribution under relation z. This updates the prior in closed form $p_T(z) = \frac{p_S(z) \exp(E_z/\tau_T)}{\sum_{z' \in Z} p_S(z') \exp(E_{z'}/\tau_T)}$ where the temperature τ_T controls how strongly target evidence shifts the source prior. Finally, the composition weights are refined toward the posterior by minimizing the objective L_A:

$$L_A = \mathrm{KL}\big(p_T \,\big\|\, w\big) \tag{3}$$

The optimization leads to the *Analogical Function Vector* (AFV): $\mathbf{v_A} = \sum_{z \in Z} w_z \mathbf{v_z}$, where $w = \mathrm{softmax}(\theta)$, visualized in Figure 2. The resulting weights represent dynamically refined relational knowledge that is consistent with both the source and target domains. This representation

is transferred via inference-time intervention, where $\mathbf{v_A}$ is injected into the LLM's activations during the forward pass of the target query x_T : ?. Because the AFV is custom-tailored to the analogy's structural constraints, it functions as a steerable, parameter-efficient mechanism for enforcing relational consistency during zero-shot inference.

4 EXPERIMENTS

Model details. To construct and evaluate our approach, we use GPT-2-Medium, GPT-J-6B, LLaMA-2-7B-Chat, and LLaMA-3.1-8B-Instruct. Architectural specifications for these models are summarized in Appendix Table 2. We train and evaluate the vectors for each model on a single NVIDIA RTX A5500 GPU using PyTorch and Hugging Face Transformers modules. Additional hyperparameters, experimental configurations, and per-analogy inference cost are detailed in Appendix Section A.2.

4.1 DATASETS AND SETUP

Datasets. We first construct a pool of word pairs instantiating 118 relations, including 15 simple relations from the original paper by Todd et al. (2024), 79 relations grouped into 10 relation types from SemEval-2012 Task-2 dataset (Jurgens et al., 2012), 8 relations from the Google dataset (Mikolov et al., 2013a), and 16 relations from the Microsoft Research (MSR) dataset (Mikolov et al., 2013b). The detailed list of these relations and representative examples is provided in Tables 4 and 7 of Appendix Section A.1. In addition, we use three question-answer tasks from the original paper (Todd et al., 2024) that focus on classification or problem-solving (AG news, CommonsenseQA, Sentiment Analysis). These tasks are used only to evaluate and compare fine-tuned function vectors.

For the analogy benchmarks, we use three complementary datasets: Green's analogy dataset (Green et al., 2010), containing 80 four-term problems (40 near analogies and 40 far analogies); the SAT dataset (Turney et al., 2003), containing 374 analogy problems; and Turney's dataset (Turney, 2008), containing 20 pairs of analogous domains that we convert into 374 four-term analogy problems. We evaluate analogy completion using generative four-term prompts (e.g., "blindness : sight :: poverty : ?"), and report top-5 accuracy which reduces sensitivity to lexical competition by allowing the target to appear among the model's five highest-ranked predictions. All reported improvements are differences in top-5 accuracy, measured in percentage points. Baseline performance varies substantially with model scale and analogy difficulty: for example, GPT-2 achieves 20.0% for Green's far-analogy problems, 19.8% on the SAT problems, whereas LLaMA-3.1 achieves 55.0% and 52.7%, respectively. The full results are provided in Table 12.

Fine-tuned Function Vector Testing. For the simple-task and question-answer datasets, each individual relation contains many examples of word pairs (more than 100). We first apply causal mediation analysis to derive an initial function vector for each relation, and then use this vector to initialize subsequent optimization. To facilitate comparison with prior work and other fine-tuning-based baselines, the examples for each relation are partitioned into three subsets: 1) 70% (138-7673 pairs) for extracting the initial function vector, 2) 10% (18-987 pairs) for optimizing the function vector, and 3) 20% (41-2302 pairs) for zero-shot testing of generalization to held-out examples.

For the SemEval dataset, each individual relation contains relatively few examples ($\sim$25 word pairs). We conduct causal mediation analysis at the broader relation-type level and use the resulting vector to initialize the FFV for each specific relation. For each relation, 10 word pairs are used for FFV optimization and approximately 3-5 held-out pairs are used for testing; the remaining examples are pooled with other relations of the same type to derive the initial function vector.

For zero-shot relation completion (e.g., given the *opposite* relation, "rich : ?"), we evaluate on held-out examples from the simple-task and question-answer datasets of Todd et al. (2024), enabling direct comparison with prior work, as well as on SemEval-2012 Task-2 (Jurgens et al., 2012), which additionally supports comparison with human relational-similarity judgments. There is no overlap in word pairs between the training and testing sets for any of these datasets.

Analogical Function Vector Testing. For one-shot analogies (e.g., given blindness : sight, "poverty : ?"), we evaluate on three benchmark datasets spanning different forms of analogical reasoning

mentioned above. Green's and SAT analogies are presented directly in four-term format, whereas the Turney problems additionally include narrative context describing each source and target domain.

These evaluation datasets are held out from FFV construction and optimization, and we found no exact word-pair overlap with the training data used to construct the base FFV library. Further details on the datasets and experiments are included in Appendix Sections A.1 and A.2 respectively.

5 RESULTS

5.1 FINE-TUNED FUNCTION VECTOR EVALUATION

For the initial function vector (FV), we replicated the results from Todd et al. (2024) in that adding the function vector improves the model's performance in the zero-shot tasks (e.g., "rich : ?"). The fine-tuned function vectors (FFVs) consistently outperform the initial function vectors, yielding increases of 30-61 pp for GPT-J (30 for simple-task dataset, 61 for question-answer dataset, 33 for SemEval) and 27-60 pp for GPT-2 (39 simple-task, 60 question-answer, 27 SemEval). For larger-scale models such as LLaMA-2 and LLaMA-3.1, the FFVs achieve roughly 20-30 pp improvements over the initial FVs on the simple-task and SemEval datasets, and 50-70 pp on the question-answer dataset. Figure 9 shows the average accuracy for SemEval across all four LLMs for the relation completion task. The substantial improvements observed across both small- and large-scale LLMs suggest that fine-tuned function vectors capture relational knowledge more effectively than those derived from causal mediation analysis. Detailed results for the LLM baseline, initial FVs, and FFVs on the three datasets are provided in Appendix Table 9, which reports the average accuracy across 5 seeds. Appendix Figure 7 reports layer-wise accuracy on the SemEval dataset, showing that FV intervention is most effective in the middle layers. Based on this result, we use layer 12 as the injection layer throughout the paper.

We compare the performance of fine-tuned function vectors with four prominent models based on activation steering and fine-tuning methods: ActAdd (Turner et al., 2023), Low-Rank Adaptation (LoRA) (Hu et al., 2022), Localized Fine-Tuning (LoFiT) (Yin et al., 2024), and Dynamic Vectors (DyVec) (Cai et al., 2025). This model comparison was conducted using six representative relations from the simple-task dataset (Todd et al., 2024), as each relation contains hundreds of instantiating word pairs that satisfy the larger training data requirements of LoRA, LoFiT and DyVec. For ActAdd, we extract steering vectors using the counterbalancing approach $\mathbf{h}_+ - \mathbf{h}_-$ for the six simple-task relations from Todd et al. (2024). For LoFiT, we use the same configurations provided in Yin et al. (2024); LLaMA-2's defaults were also used for LLaMA-3.1, and Gemma's defaults were used for the GPT models. For LoRA and DyVec, we follow the same configurations and training procedures provided by Cai et al. (2025). Further implementation details are provided in Appendix Section A.3.1.

We evaluated fine-tuned function vectors in GPT-J and LLaMA-2 using interventions at layer 12. To facilitate direct comparison with results reported in prior work, we use Top-1 accuracy. As detailed in Table 1 and Appendix Table 8, fine-tuned function vectors match or exceed every baseline at inference time. While ActAdd yields modest gains, increasing Top-1 accuracy by +10.05 pp for GPT-J and +8.06 for LLaMA-2, LoRA and DyVec provided more substantial improvements, with +64.06 and +69.84 pp for GPT-J and +62.77 and +67.11 for LLaMA-2 respectively. LoFiT led to the strongest improvements for LLaMA-2 compared to the other baselines at +75.22 pp. Ultimately, our FFV demonstrates the highest efficacy on average, delivering maximum accuracy increases of +79.51 pp for GPT-J and +75.63 for LLaMA-2. While it was competitive with LoFiT on relations with the most training pairs, the FFVs maintained high accuracies for those with small amounts of data.

Furthermore, we tested whether FFVs align the model's relational representations with human judgments, correlating relational dissimilarity matrices of the LLM's activations with human dissimilarity judgments from Experiment 2 of Ichien et al. (2022) (Figure 3). For GPT-J, the correlation increases to 0.734 with FFV intervention, compared with 0.228 without intervention and 0.542 with the initial FVs. These correspond to improvements of $+0.51$ (95% CI $[0.34, 0.67]$) over the baseline LLM, and $+0.19$ ($[0.12, 0.25]$) over the initial FV. The confidence intervals were computed by bootstrap resampling over word pairs. Results for all models are reported in Appendix Table 11. Together, these findings suggest that fine-tuned function vectors not only improve task performance but also yield relational representations that more closely correspond to human judgments of relational similarity.

Method	GPT-J	LLaMA-2
LLM	7.50%	13.28%
+ ActAdd	17.55%	21.34%
+ LoRA	71.56%	76.05%
+ DyVec	77.34%	80.39%
+ LoFiT	48.71%	88.50%
+ $\mathbf{v_t}$ Initial FV	56.75%	68.45%
+ $\mathbf{v_z}$ FFV	**87.01%**	**88.91%**

Table 1: Average Top-1 accuracies on the zero-shot relation-completion task across six relations in the simple-relation dataset. FV = function vector, FFV = fine-tuned function vector.

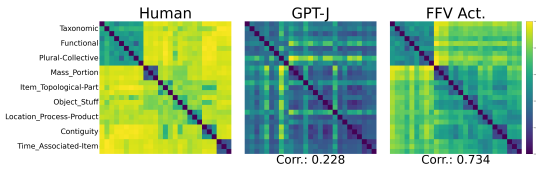


Figure 3: Relational dissimilarity matrices of human judgments and model activation-derived representations. Each cell represents the dissimilarity between the row's word pair and the column's pair. Warmer colors indicate more dissimilarity.

5.2 Analogical Function Vector Evaluation

We evaluate the analogical function vector across tasks with differing levels of linguistic context: the standard four-term analogy format ("A : B :: C : ?") and the narrative-based analogy (e.g., Appendix Table 6). This evaluation differs fundamentally from the relation-completion setting used for FFVs. Unlike FFV evaluation where the relevant task is explicitly given to the model, AFV evaluation uses a single source analog as context without explicitly stating its underlying relation. The model must therefore infer the latent relation from the source analog and transfer it to the target analog. Our AFV framework is designed to support this process by composing learned relation vectors to represent the source relation and then refining that representation to transfer to the target analog.

5.2.1 Out-Of-Distribution Evaluation with Four-Term Analogy: SemEval

Since AFVs are constructed by composing learned relation vectors to represent an unseen relation, a central question is whether this compositional mechanism can generalize beyond the relations explicitly represented in the base vector library. To test this directly, we construct an analogy dataset from SemEval word pairs, which allows us to control whether the target relation is in- or out-of-distribution with respect to the available FFV library. In total, we generate 40,770 analogies from the 79 SemEval relations. Because many analogies share the same underlying word-pair pools, we treat each relation, rather than each analogy instance, as the unit of analysis.

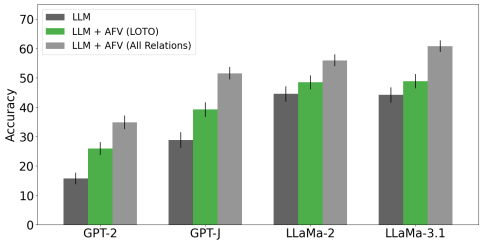


Figure 4: Top-5 accuracy on SemEval analogies for the Leave-One-Type-Out (LOTO) task.

We evaluate out-of-distribution generalization using a Leave-One-Type-Out (LOTO) design. For each fold, all FFVs belonging to one SemEval relation type are withheld from the base library, and analogies drawn from that held-out type are evaluated only in that fold. For example, when evaluating the analogy "black : white :: large : ?" the model may compose FFVs corresponding to relation types such as similarity, class inclusion, and causality, but it has no access to FFVs from the held-out *contrast* relation type. The model must therefore construct a composite representation of the unseen source relation from the remaining FFVs and then refine it into an AFV for analogical transfer. We assess performance differences with a Wilcoxon signed-rank test across the 79 relations and compute 95% confidence intervals by bootstrapping over relations.

The LOTO results show that AFVs retain substantial gains even when the relation type required by the test analogy is completely absent from the base FFV library. Relative to the one-shot baseline, AFV intervention improves Top-5 accuracy by 10.7 pp for GPT-2, 10.4 for GPT-J, 4.0 for LLaMA-2, and 4.5 for LLaMA-3.1 ($p \leq 3.8 \times 10^{-5}$). Depending on the model, AFVs outperform the baseline on 58-74 of the 79 individual relations. As expected, performance improves further when the complete

library of 79 SemEval relations is available: compared to the baseline, the accuracy increases by 19.4 pp for GPT-2, 22.1 for GPT-J, 11.1 for LLaMA-2, and 15.8 for LLaMA-3.1. Together, these results show that although access to the matching relation type provides additional benefit, a substantial portion of such improvements remains when that type is withheld. Additional LOTO results, as well as a more stringent Leave-Two-Types-Out (LTTO) evaluation, are reported in Appendix Section A.4 and Table 15.

5.2.2 FOUR-TERM ANALOGY: GREEN/SAT

Green's Far-Analogy	SAT Problem
answer:riddle :: keys:(*lock*)	repeal:law :: withdraw:(*offer*)

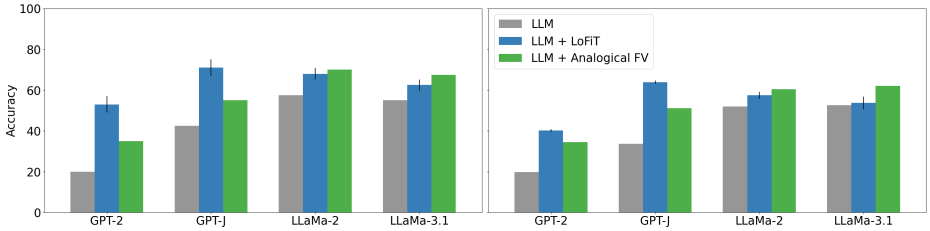


Figure 5: Top: Example analogies from Green's and SAT datasets. Italicized words represent the text that is needed to be generated by the model. Bottom: Top-5 prediction accuracy for the one-shot analogies in Green's far-analogy and SAT datasets. The grey bars represent the LLM baseline, the blue bars represent the LoFiT intervention model, and the green bars represent the AFV intervention. AFV = analogical function vector.

As illustrated in Figure 5, injecting analogical function vectors to LLMs yielded improvements over the LLM alone for four-term analogy completion. For the far-analogy problems in Green's dataset, in which the target analog is drawn from a distinctly different semantic domain than the source's (e.g., *blindness : sight :: poverty : ?*), AFV intervention improved accuracy by 12.50 pp over the GPT-J baseline (Figure 5, left). Comparable gains were observed across model scales (+15.00 pp GPT-2, +12.50 LLaMA-2, and +12.50 LLaMA-3.1). On the SAT dataset, AFV intervention likewise improved performance over the corresponding baseline LLM for all four models, with gains of +14.70 pp GPT-2, +17.38 GPT-J, +8.56 LLaMA-2, and +9.36 LLaMA-3.1. Together, these results show that the performance gains from AFV intervention generalize across both analogy datasets and across models of substantially different scales. Detailed results for both datasets and all four models are provided in Appendix Section A.4 and Tables 12 and 16 of the Appendix.

To compare AFV intervention with a fine-tuning-based approach, we trained LoFiT on 2,898 analogies constructed from the 118 relations and used it as a baseline. This comparison is deliberately stringent because LoFiT receives direct supervision on four-term analogy problems presented in the same format as the evaluation tasks, whereas AFV is not trained on four-term analogies and instead adapts its composition weights at inference time. AFV achieved higher accuracy than LoFiT for the larger models in both Green's dataset (+2.0 pp LLaMA-2, +5.0 LLaMA-3.1) and the SAT dataset (+3.0 pp LLaMA-2, +8.3 LLaMA-3.1). In contrast, LoFiT performed better than AFV for the smaller GPT-2 and GPT-J models; the task-specific supervision available to LoFiT may therefore be particularly beneficial for smaller models. However, LoFiT's advantage depends in part on the close match between its fine-tuning and evaluation formats. When the evaluation prompt deviates from the standard `Q :/A :` format used during LoFiT training, its performance decreases by 5.19-36.79 pp across four alternative templates. In contrast, AFV maintains its original performance on three of the four templates and declines by only 7.22 pp on the remaining template. These results suggest a trade-off between task-specific fine-tuning and inference-time intervention: LoFiT can provide strong performance when training and test formats are closely matched, particularly for smaller models, whereas AFV appears more robust to changes in prompt format without requiring analogy-specific training. Further details are reported in Appendix Section A.4.

In addition, we conduct ablation studies to validate the necessity of constructing the AFV from the full fine-tuned function vector library (Appendix Section A.4.2). We found that replacing the FFVs with the initial FVs removes most of the gain for GPT-J on the narrative task (18.45% vs. 22.46%

with FFVs, 17.91% one-shot; Appendix Table 16), while reducing the number of base relations by approximately half did not significantly affect AFV performance, provided that the remaining FFVs preserved comparable coverage of the relational representation space.

5.2.3 Narrative Analogy

To evaluate narrative-based analogical reasoning, we use Turney's domain-mapping dataset augmented with LLM-generated narrative contexts spanning multiple source and target domains. As illustrated in Appendix Table 6, the input prompts for this task embed the analogical mapping within a narrative context, structured as a question-and-answer format. The performance outcomes, detailed in Figure 6, demonstrate that Analogical Function Vector (AFV) interventions consistently enhance accuracy across all tested model scales compared to standard one-shot baselines, even when the weights were updated using contextless prompts (i.e., the four-term format "A : B :: C : ?"). Specifically, AFVs yielded improvements of +20.86 pp for GPT-2, +4.55 for GPT-J, +5.35 for LLaMA-2, and +2.14 for LLaMA-3.1.

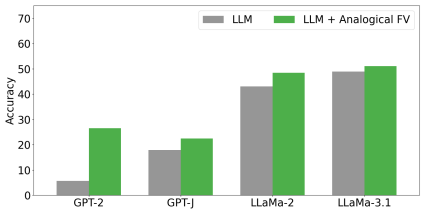


Figure 6: Top-5 prediction accuracy for the narrative-context analogy completion task in Turney's dataset.

The right column of Appendix Table 12 shows that the AFVs improve on the composite weights through refinement for the GPT models (+5.08 pp GPT-2, +4.28 GPT-J) while matching it for the larger LLaMA models (0.00 pp LLaMA-2, +0.27 LLaMA-3.1). These results demonstrate that injecting AFVs into LLMs improves their capacity to complete analogies with complex, narrative-level context. The AFV's performance gains can be attributed to composing relational knowledge from the source analogy, while its consistency comes from alignment to the target domain. By providing this structural guidance, the AFV enables the LLM to pinpoint the relevant relational knowledge within the source and target narratives that fulfill the analogy.

6 Future Work and Limitations

One limitation concerns the hierarchical nature of human relational representations. For example, whole–part relations encompass multiple subtypes, such as object–component (*hand : finger*), collection–member (*army : soldiers*), and mass–portion (*hour : seconds*). Further research is needed to determine whether LLMs naturally acquire these hierarchical structures in their relational representations, or whether additional alignment is required to achieve them. Another limitation is that our current evaluation of narrative analogies relies on pre-provided keywords for mapping using the question-answer format. When applied to analogies with paragraph-long narratives, the performance in LLMs degrades significantly, often presenting gaps between LLMs and humans especially for semantically far analogies (Sourati et al., 2024). Future research should explore the broader interplay between general narrative comprehension and analogical mapping. Specifically, developing mechanisms to autonomously extract relevant relational concepts from unstructured, long-context narratives will be crucial for scaling this framework to real-world reasoning tasks.

7 Conclusion

In this work, we demonstrate that structural relational knowledge can be robustly extracted, refined, and composed within Large Language Models through an expanded framework of function vectors. Our results suggest that LLMs do not need to rely solely on semantic associations to navigate complex reasoning tasks. By extracting, composing, and dynamically refining relational vectors, our framework enables analogical inference across distinct domains without analogy-specific training or access to the target's answer. This approach not only enhances model generalization on rigorous cognitive benchmarks but also provides a more interpretable, mechanistically grounded framework for understanding how LLMs can form relational knowledge more closely aligned with human judgments.

AI USE STATEMENT

In this work, we used large language models to generate the narrative contexts for Turney's dataset, expand on the methods' code, and run simulations for baseline comparisons. We did not use generative AI for formulating any mathematical or theoretical claims. We reviewed all AI-assisted work: the generated narratives were confirmed to have the appropriate keyword mappings, while the generated experiment code was verified against anchors reproducing previous results to ensure that all results can be traced to the proper files. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

All experiments use publicly available models (Appendix Table 2) and datasets (Section 4, Appendix A.1). Hardware specifications, data splits, fine-tuning and refinement hyperparameters, and stopping rules are provided in Appendix A.2. The base FFVs are averaged over five seeds, making the composite and AFV evaluations deterministic. We provide the results in the attached supplemental files and an anonymous repository.

ETHICS STATEMENT

This work analyzes and steers internal representations of publicly available language models on public analogy benchmarks. No experiments with human subjects were conducted; the data used for the human similarity evaluation were reused from the original publications. Note that the narrative stimuli were LLM-generated and author-reviewed.

REFERENCES

Isaac I Bejar, Roger Chaffin, and Susan Embretson. *Cognitive and psychometric analysis of analogical problem solving*. Springer Science & Business Media, 2012.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Wang Cai, Hsiu-Yuan Huang, Zhixiang Wang, and Yunfang Wu. Beyond demonstrations: Dynamic vector construction from latent representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5842–5857, Suzhou, China, 2025.

Roger Chaffin. The concept of a semantic relation. In *Frames, fields, and contrasts*, pages 253–288. Routledge, 2012.

Jeffrey N Chiang, Yujia Peng, Hongjing Lu, Keith J Holyoak, and Martin M Monti. Distributed code for semantic relations predicts neural similarity during analogical reasoning. *Journal of Cognitive Neuroscience*, 33(3):377–389, 2021.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

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.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Takashi Furuya, Maarten V de Hoop, and Gabriel Peyré. Transformers are universal in-context learners. *arXiv preprint arXiv:2408.01367*, 2024.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Dedre Gentner. Children's performance on a spatial analogies task. *Child development*, pages 1034–1039, 1977.

Dedre Gentner and Arthur B Markman. Structure mapping in analogy and similarity. *American psychologist*, 52(1):45, 1997.

Dedre Gentner and Cecile Toupin. Systematicity and surface similarity in the development of analogy. *Cognitive science*, 10(3):277–300, 1986.

Mary L Gick and Keith J Holyoak. Schema induction and analogical transfer. *Cognitive psychology*, 15(1):1–38, 1983.

Jeremy R Gray and Paul M Thompson. Neurobiology of intelligence: science and ethics. *Nature Reviews Neuroscience*, 5(6):471–482, 2004.

Adam E Green, David JM Kraemer, Jonathan A Fugelsang, Jeremy R Gray, and Kevin N Dunbar. Connecting long distance: semantic distance in analogical reasoning modulates frontopolar cortex activity. *Cerebral cortex*, 20(1):70–76, 2010.

Douglas R Hofstadter and Melanie Mitchell. The Copycat project: A model of mental fluidity and analogy-making. In *Fluid Concepts and Creative Analogies*, chapter 5. Basic Books, 1995.

Keith J Holyoak. 13 analogy and relational reasoning. *The Oxford handbook of thinking and reasoning*, page 234, 2012.

Keith J Holyoak. *The human edge: Analogy and the roots of creative intelligence*. MIT Press, 2025.

Keith J Holyoak, Nicholas Ichien, and Hongjing Lu. Analogy and the generation of ideas. *Creativity Research Journal*, 36(3):532–543, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Nicholas Ichien, Hongjing Lu, and Keith J Holyoak. Predicting patterns of similarity among abstract semantic relations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 48(1): 108, 2022.

Ole Jorgensen, Dylan Cope, Nandi Schoots, and Murray Shanahan. Improving activation steering in language models with mean-centring. *arXiv preprint arXiv:2312.03813*, 2023.

David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. SemEval-2012 task 2: Measuring degrees of relational similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 356–364, 2012.

Taewhoo Lee, Minju Song, Chanwoong Yoon, Jungwoo Park, and Jaewoo Kang. The curious case of analogies: Investigating analogical reasoning in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 31492–31500, 2026. doi: 10.1609/aaai.v40i37.40414.

Martha Lewis and Melanie Mitchell. Using counterfactual tasks to evaluate the generality of analogical reasoning in large language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Hongjing Lu, Ying Nian Wu, and Keith J Holyoak. Emergence of analogy from relation learning. *Proceedings of the National Academy of Sciences*, 116(10):4176–4181, 2019.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013b.

Melanie Mitchell. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101, 2021.

Melanie Mitchell and David C Krakauer. The debate over understanding in AI's large language models. *Proceedings of the National Academy of Sciences*, 120(13):e2215907120, 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39643–39666, 2024.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI technical report, 2019.

Zhivar Sourati, Filip Ilievski, Pia Sommerauer, and Yifan Jiang. ARN: Analogical reasoning on narratives. *Transactions of the Association for Computational Linguistics*, 12:1063–1086, 2024.

Charles Spearman. *The nature of "intelligence" and the principles of cognition*. Macmillan, 1923.

Barbara A Spellman and Keith J Holyoak. Pragmatics in analogical mapping. *Cognitive psychology*, 31(3):307–346, 1996.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland, 2022.

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.

Oguzhan Topsakal and Tahir Cetin Akinci. Creating large language model applications utilizing LangChain: A primer on developing LLM apps fast. In *International Conference on Applied Engineering and Natural Sciences*, volume 1, pages 1050–1056, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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.

Peter Turney. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd international conference on computational linguistics (COLING 2008)*, pages 905–912, 2008.

Peter D Turney, Michael L Littman, Jeffrey Bigham, and Victor Shnayder. Combining independent modules to solve multiple-choice synonym and analogy problems. *arXiv preprint cs/0309035*, 2003.

Ben Wang. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541, 2023.

Taylor W Webb, Keith J Holyoak, and Hongjing Lu. Evidence from counterfactual tasks supports emergent analogical reasoning in large language models. *PNAS nexus*, 4(5):pgaf135, 2025.

Jin Yan, Kenneth D Forbus, and Dedre Gentner. A theory of rerepresentation in analogical matching. In *Proceedings of the 25th Annual Cognitive Science Society*, pages 1265–1270. Psychology Press, 2013.

Yuzi Yan, Jialian Li, Yipin Zhang, and Dong Yan. Exploring the LLM journey from cognition to expression with linear representations. *arXiv preprint arXiv:2405.16964*, 2024.

Fangcong Yin, Xi Ye, and Greg Durrett. LoFiT: Localized fine-tuning on LLM representations. *Advances in Neural Information Processing Systems*, 37:9474–9506, 2024.

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 APPENDIX

Model details In this paper, we use GPT-2-Medium, GPT-J-6B, LLaMA-2-7B-Chat, and LLaMA-3.1-8B-Instruct. Each of the models are deployed through Hugging Face - details of their architectures are in Table 2.

Model	Citation	Parameters	$\|L\|$	$\|a_l\|$
GPT-2	Radford et al. (2019)	355M	24	16
GPT-J	Wang (2021)	6B	28	16
LLaMA-2-Chat	Touvron et al. (2023)	7B	32	32
LLaMA-3.1-Instruct	Dubey et al. (2024)	8B	32	32

Table 2: Summary of the models used for this paper. We provide the number of parameters, layers $|L|$, and attention heads per layer $|a_l|$ for each model.

A.1 DATASET DETAILS

Fine-tuned Function Vectors. The simple-task dataset contains relatively straightforward relations (e.g., country : capital), with varying amounts of word pairs (197-2699) per relation. In contrast, the question-answer dataset consists of relations that involve classifying a sentence to a category or answer (i.e. news headline : news category, question : answer) with a large number of pairs (1167-10962) per relation. The SemEval dataset covers a broad range of abstract relations (e.g., category membership, cause-effect) but offers only 17-29 word pairs per relation. The 79 specific relations are grouped into 10 relation types in SemEval to reflect the hierarchical structure of human relation representations. Examples are provided in Table 3.

The Google dataset is similar to the simple-task dataset in that it consists of straightforward semantic and syntactic relations (e.g. adjective : adverb) - however, each relation offers only 20-40 word pairs like SemEval. The MSR dataset only consists of syntactic relations, most of which are counterparts of each other (e.g. adjective : comparative vs. comparative : adjective) and offer 100 word pairs each. We list the remaining tasks in Table 7.

Composite and Analogical Function Vectors. Green's dataset includes 80 analogies without assigned relation labels; examples include within-domain/semantically near analogies (e.g., *blindness : sight :: deafness : hearing, furnace : coal :: woodstove : wood*) and cross-domain/semantically far analogies (e.g., *blindness : sight :: poverty : wealth, furnace : coal :: stomach : food*). The dataset has a total of 40 source analogs, each of which consists of two target analogs (one within-domain, one cross-domain). The far-analogy/cross-domain analogies are listed in Table 5.

The SAT dataset consists of 374 problems from SAT resources, with 90 questions from prep sites, 14 from ETS's site directly, 190 from previous SAT exams, and 80 from guidebooks. Each problem includes a source analog, the target analog that matches the source's relation, and four incorrect analogs akin to the SAT's multiple-choice format. For this paper, we only use the correct answer as the target analog.

Turney's dataset is comprised of 20 analogy groups, where each group consists of a source domain and a target domain. Each domain includes 5-9 words that are interconnected to each other through specific relations, all of which are represented in both the source and target. We evaluate the AFV on this dataset under the standard four-term template (i.e. *sun : planet :: nucleus : electron*) with a full narrative containing the domain's keywords as context. The narratives were generated using ChatGPT; examples of this are provided in Table 6.

A.2 EXPERIMENT DETAILS

We tested the LLM and our function vectors (FVs) using a broad range of semantic relations in the SemEval-2012 Task-2 dataset (Jurgens et al., 2012). This dataset, taken from a linguistic taxonomy (Bejar et al., 2012), contains a hierarchical structure of 10 general relation types (e.g., class inclusion, contrast, cause-purpose), each of which consists of 5-10 subtypes for a total of 79 primitive semantic

Type	Relation	Example
Class Inclusion	Functional	vehicle → car
	Collective	clothing → shirt
Part - Whole	Object:Component	face → nose
	Mass:Portion	water → drop
	Creature:Possession	millionaire → money
Similar	Synonym	buy → purchase
	Attribute Similarity	rake → fork
	Change	brighten → color
Contrast	Contradictory	remember → forget
	Defective	limp → walk
Attribute	Object:State	beggar → poverty
	Object:Typical Action	soldier → fight
Nonattribute	Object:Nonstate	war → tranquility
	Object:Atypical Action	recluse → socialize
Case Relation	Agent:Instrument	conductor → baton
	Action:Recipient	teach → student
	Recipient:Instrument	graduate → diploma
Cause-Purpose	Cause:Effect	joke → laughter
	Action:Goal	fertilize → grow
	Prevention	antidote → poison
Space-Time	Location:Item	bookshelf → books
	Location:Action	school → learning
	Sequence	prologue → narrative
Reference	Expression	smile → friendliness
	Plan	recipe → cake

Table 3: A subset of tasks and their paradigm examples from each relation type in the SemEval dataset.

	Example	
	Query	**Answer**
AG News	Dogs in Training to Sniff Out Cancer...	Science
CommonsenseQA	What home entertainment equipment requires cable? a: radio shack b: substation c: cabinet d: television e: desk	d
Sentiment Analysis	Very well-written and very well-acted.	Positive

Table 4: The relations in the question-answer dataset and their examples. These were used to test the efficacy of fine-tuned FVs in relations that involve more complicated formats.

relations. The full dataset consists of 3,215 word pairs (35-48 per relation) with prototypicality ratings; we only keep the 1,917 pairs with positive ratings (17-29 per relation), which are split for optimization vs. testing as described in Section 4. For each semantic relation, we can compute its corresponding function vector from the LLM's in-context learning.

To compute the fine-tuned function vector, we first compute or obtain the initial function vector $\mathbf{v_t} = \sum_{a_{lj} \in A} \bar{a}^z_{lj}$. For n epochs, we then train $\mathbf{v_t}$ into $\mathbf{v_z}$ by adding it to an intermediate layer of

Source Analog	Target Analog
answer : riddle	keys : lock
ash : fireplace	lint : pocket
aspirin : pain	muffler : noise
baker : cake	scientist : discovery
basket : picnic	holster : gun
basketball : hoop	traveler : destination
blindness : sight	poverty : money
blizzard : snowflake	army : soldier
bracelet : wrist	moat : castle
burger : bun	book : cover
cleanser : face	absolution : sinner
eraser : pencil	amnesia : memory
father : son	inventor : invention
flock : goose	constellation : star
foresight : future	xray : bone
foundation : house	premise : argument
furnace : coal	stomach : food
hoof : hoofprint	introduction : impression
immunization : disease	forewarning : surprise
jacket : zipper	wound : suture
ketchup : tomato	fuel : petroleum
kitten : cat	spark : fire
knee : kneepad	snail : shell
lamb steak : lamb	chapter : book
landscaper : lawn	stylist : hair
launchpad : helicopter	diving board : diver
school : lawyer	vineyard : wine
movie : screen	lightning : sky
multiplication : product	brewing : beer
nose : scent	antenna : signal
orchard : apple	neighborhood : apartment
painting : canvas	birthmark : skin
pen : pig	reservoir : water
rectangle : perimeter	nation : border
revising : manuscript	evolving : species
saxophone : jazz	typewriter : poetry
sugar : coffee	incentive : deal
thermometer : temperature	polygraph : honesty
train : track	signal : wire
watermelon : rind	cigarette : butt

Table 5: The word pairs used for Green's far-analogy dataset.

the LLM and calculating the loss from there. For training the FFVs, we use an AdamW optimizer with a learning rate of 0.01 on a NVIDIA RTX A5500 GPU. The FFVs are trained for 25 (SemEval) and 10 (other relations) epochs in GPT-2, 10 and 5 epochs in GPT-J, and 5 epochs in LLaMA-2 and 3.1. For all models, we use layer 12 to inject and train the FFV on as it consistently led to the highest accuracies for the initial FV. By running one word pair per relation in each batch, we can backpropagate on all of the dataset's FVs at once. This was done for 5 seeds, each sampling a new set of word pairs from the relation's pool. Evaluation is done with top-k accuracy, which measures the proportion of word pairs in the test set for which the model correctly generates the answer as one of the top k predicted responses. While we use Top-1 accuracy for the simple and question-answer relations, we use Top-5 accuracy for SemEval to consider the semantic flexibility of the relations; many of the word pairs in each relation can allow for multiple correct terms in comparison to the straightforwardness of the simple tasks with a single ground-truth answer.

Source Analog	Target Analog
Context: The **sun**, a massive star at the center of the **solar system**, contains about 99.86% of the total **mass**... Each **planet** revolves around the **sun** in an elliptical path... **Gravity**, the fundamental force responsible for the attraction between **masses**, is the key mechanism that maintains the structure of the **solar system**. Q: sun A: planet	Context: An **atom** is the fundamental unit of matter, consisting of a dense central **nucleus** surrounded by **electrons**. The **nucleus** exerts an attractive force on the negatively charged **electrons** due to the fundamental force of **electromagnetism**... The balance of these **charges** and forces is crucial for the stability of the **atom**. Q: nucleus A: (electron)

Table 6: An example of the narrative analogy prompts from Turney's dataset. Bolded words in the narrative represent keywords in the analog's domain.

To evaluate model performance, the FFV is injected to layer 12 of the LLM to perform the zero-shot and one-shot analogy tasks in the test set. This set consists of word pairs that the LLM has never encountered in previous phases of the vector's development (both initialization and optimization) until now. For the one-shot analogy task, we use the paradigm exemplars in SemEval - these are defined as the word pairs that best resemble the current relation. In conjunction with the pre-trained LLM, the model with the FFV can estimate the probability distribution of the response word. Given a query word q and the FFV $\mathbf{v_z}$ identified for a relation z, the prediction distribution $P(r|q,z)$ can be computed by adding the corresponding $\mathbf{v_z}$ to the attention heads' activations a_l in layer l as follows:

$$\mathrm{P}(r|q,z) \sim \mathrm{LLM}(r|q, a_l := a_l + \mathbf{v_z}) \tag{4}$$

For the composite vector and analogical function vector (AFV), we used all of the SemEval dataset's (Jurgens et al., 2012) relations as well as tasks from the Google dataset (Mikolov et al., 2013a), MSR dataset (Mikolov et al., 2013b), and the original paper (Todd et al., 2024). We also use AdamW optimizers to update the AFV's unnormalized weights per analogy: these would be the log-probabilities extracted during the relation initialization phase. We use higher learning rates (0.1 for GPT-2, 0.05 GPT-J, 0.03 LLaMA-2, 0.03 LLaMA-3.1) due to the larger range of the log-probabilities, which provide more subtle shifts in the final normalized weights. For the source relation refinement phase, we update the weights until the source analog's answer is ranked the highest or the loss stabilizes by 150 iterations at most, while for source-target alignment we update the weights for 3 iterations.

Our function vectors are evaluated through four tasks:

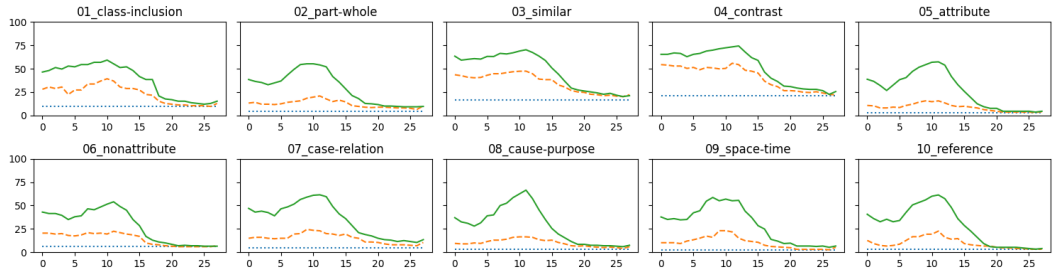
1. Shuffled-label Evaluation: For each relation, its FFV is evaluated on 10-shot prompts where the context sequences are shuffled. Accuracy is measured with respect to the proportion of word pairs in which the correct answer is the highest-ranking score. This evaluation task is used to compare the LLM baseline, the FV intervention, and the FFV intervention on the simple-task dataset.
2. Zero-Shot Evaluation: The models are evaluated on prompts with a zero-shot task given a relation (e.g., "long : ?" + the *antonym* function vector). This evaluation task is used to compare the LLM baseline, the FV intervention, and the FFV intervention on both the simple-task and SemEval datasets. Accuracy for the simple-task dataset is measured in the same way as for the shuffled-label evaluation, while for SemEval it is measured as the proportion of word pairs with correct answers that are within the top 5 scores.
3. Relational Similarity: Using a subset of the word pairs from SemEval, we compare the similarity structure of the relational representations captured by our function vectors to that of human judgments. The human results were provided by Ichien et al. (2022) - the authors utilized the multi-arrangement method to measure human similarity judgments in two experiments (Exp. 1 and Exp. 2 in the original study). Human similarity judgments are used to assess the extent to which word pairs instantiate similar relations. For example, *white : black* and *happy : sad* are judged as similar due to them representing the same type of relation, whereas *white : black* and *happy : joyful* are dissimilar because they reflect different relations.

Dataset	Relation Name	Example
Simple-Task	Antonym	flawed → perfect
	Capitalize first letter	whose → W
	Capitalize	without → Without
	Country-capital	Chile → Santiago
	Country-currency	Italy → Euro
	English-French	those → ceux
	Lowercase first letter	WITTY → w
	Next-item	zero → one
	Previous-item	one → zero
	Person-instrument	Skylar Grey → piano
	Person-occupation	Billy Roche → actor
	Person-sport	Colin Kaepernick → football
	Present-past	adapt → adapted
	Singular-plural	wallet → wallets
	Synonym	curse → swear
Google	Adjective-adverb	amazing → amazingly
	Comparatives	bad → worse
	Male-female	brother → sister
	National currency	Argentina → peso
	Nationality adjective	Denmark → Danish
	Past tense verbs	dancing → danced
	Plural verbs	eat → eats
	Present participles	code → coding
MSR	Adjective-comparative	good → better
	Comparative-adjective	better → good
	Adjective-superlative	good → best
	Superlative-adjective	best → good
	Comparative-superlative	better → best
	Superlative-comparative	best → better
	Noun-plural noun	person → people
	Plural noun-noun	people → person
	Noun-possessive noun	state → state's
	Possessive noun-noun	state's → state
	Present verb-Past verb	is → was
	Past verb-present verb	was → is
	Present verb-Infinitive verb	is → be
	Infinitive verb-present verb	be → is
	Past verb-infinitive verb	was → be
	Infinitive verb-past verb	be → was

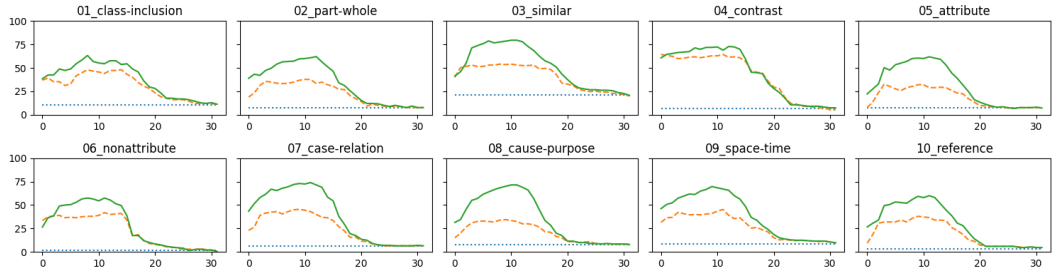
Table 7: The non-SemEval relations used for the composite and analogical function vectors and their examples, grouped by dataset.

4. One-shot Analogy Evaluation: Using the source word pair as one-shot context, models predict the answer to complete the analogy (e.g., *furnace : coal :: woodstove : ?*, in which the correct answer is "wood"). Accuracy is measured as the proportion of problems for which the answer is among the top-5 predictions. This evaluation task is used to compare the LLM baseline and AFV intervention on Green's dataset, the SAT problems, and Turney's dataset.

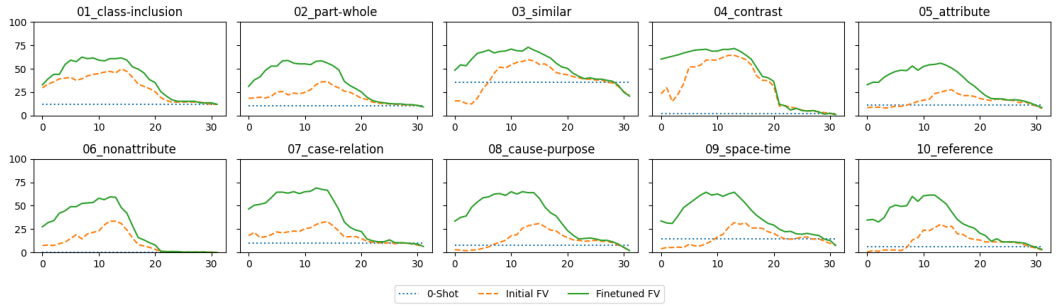
Test-Time Compute. For a Turney analogy with all 118 FFVs on the NVIDIA GPU specified previously, relation initialization runs the zero-shot source prompt once per base vector, batched across the relations (2.8 s GPT-2, 7.3 s GPT-J, 8.0 s LLaMA-2, 10.2 s LLaMA-3.1); source refinement runs 29 (LLaMA-2), 30 (LLaMA-3.1), 51 (GPT-J) and 109 (GPT-2) forward-backward passes on average (6.0, 7.0, 8.6 and 4.7 s), stopping well before its 150-step cap for all models except GPT-2. In contrast, source-target alignment only needs 16 forward passes through the model (2.7-8.5 s). The



(a) GPT-J



(b) LLaMA-2



(c) LLaMA-3.1

Figure 7: Layer-wise Top-5 prediction accuracy for the SemEval dataset by relation type for GPT-J, LLaMA-2, and LLaMA-3.1. The fine-tuned FV (solid green) exhibited similar trends to the initial FV (dashed orange) in that injecting the vector in the early-to-middle layers leads to the largest improvements, encouraging the LLM to respond to the prompt with the desired relation. Compared to those of the initial FV, the fine-tuned FV's accuracies across the earlier layers were both higher and more stable, with some relations exhibiting a gradual spike up to layer 12 before dropping. As the FV was fine-tuned based on layer 12 interventions, this may play a small factor in such a trend. FV = function vector.

cost is dominated by forward passes through the frozen model: one per base vector for initialization and the target-side probe, followed by one forward-backward pass per source-refinement step. This is incurred per analogy at inference rather than at training, with no analogy data or target answers involved.

A.3 Fine-tuned Function Vector Evaluation

The sample results for the LLM baseline, initial FVs, and fine-tuned FVs are provided in Table 9 which reports the average accuracy across 5 seeds. For the simple-task dataset we conducted simulations for the shuffled-label and zero-shot task, while for SemEval we only used the zero-shot task for evaluation due to its limited data per relation.

Alongside comparing the fine-tuned function vector with zero-shot learning and the initial function vector, we also compared it to 10-shot learning as well as a randomly initialized vector trained with

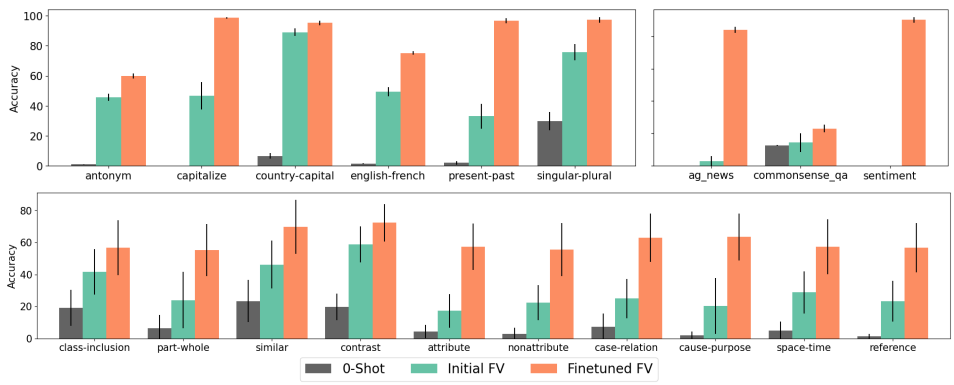


Figure 8: Zero-shot evaluation results for GPT-J: Top-1 prediction accuracy for the 6 representative relations (top left) and problem-solving relations (top right) in the simple-task dataset, and Top-5 prediction accuracy for the SemEval dataset by relation type (bottom). Baseline performance (grey) reflects running GPT-J directly on the zero-shot task. Compared to the initial FV (green), injecting the fine-tuned FV (orange) into GPT-J yielded the highest average accuracies. The error bars indicate the standard error across five runs. FV = function vector.

Method	Antonym	Capitalize	Country-Capital	English-French	Present-Past	Singular-Plural
GPT-J	3.13%	0.00%	6.67%	2.07%	1.97%	31.16%
+ ActAdd	1.47%	0.00%	9.05%	12.46%	34.43%	47.91%
+ LoRA	45.32%	68.82%	89.05%	68.09%	67.87%	90.23%
+ DyVec	36.87%	92.59%	88.10%	73.92%	77.70%	94.88%
+ LoFiT	**61.87%**	86.59%	29.05%	**78.38%**	1.97%	34.42%
+ $\mathbf{v_t}$ FV	45.08%	39.29%	87.14%	42.84%	36.39%	89.77%
+ $\mathbf{v_z}$ FFV	59.48%	**98.12%**	**94.29%**	75.89%	**97.05%**	**97.21%**
LLaMA-2	1.67%	34.00%	11.90%	3.65%	7.54%	20.93%
+ ActAdd	0.99%	45.41%	7.62%	10.80%	39.02%	24.19%
+ LoRA	37.94%	95.65%	84.76%	75.68%	82.30%	80.00%
+ DyVec	46.67%	98.24%	93.33%	50.11%	97.70%	96.28%
+ LoFiT	**69.40%**	**99.41%**	90.95%	**79.45%**	97.38%	94.42%
+ $\mathbf{v_t}$ FV	56.39%	98.35%	95.24%	21.84%	56.07%	82.79%
+ $\mathbf{v_z}$ FFV	66.27%	99.18%	**96.19%**	75.85%	**97.38%**	**98.60%**

Table 8: Average zero-shot Top-1 accuracies on the six simple-task relations under various interventions and baselines with GPT-J and LLaMA-2.

the same method. Since the initial FV is computed using 10-shot prompts, we tested if the FFV's performance in the zero-shot tasks match those of the 10-shot tasks where the relation can be easily determined. The randomly initialized vector was incorporated as another baseline to determine the necessity of using the initial FV for our finetuning method. From the results in Figure 10, we see that our FFV outperforms the randomly initialized FV while having competitive performances with 10-shot learning.

A.3.1 Baseline Details

ActAdd. We use the methods and code provided by Turner et al. (2023) to extract and evaluate the steering vectors for the six simple-task relations from Todd et al. (2024). This includes the counterbalancing approach, where the vector $\mathbf{h}_+ - \mathbf{h}_-$ is the difference in activations for the basis relation $\mathbf{h}_+$ and an opposing term $\mathbf{h}_-$ as prompts; for example, the resulting vector for the "singular-plural" relation would be $\mathbf{h_{Plural}} - \mathbf{h_{Singular}}$. We use a coefficient of $\lambda = 5$ to extract the steering vectors.

	Simple-Task		Question-Answer	SemEval
	Shuffled-Label	Zero-Shot	Zero-Shot	Zero-Shot
$[(x_{i1}, \tilde{y}_{i1}), ..., (x_{iN}, \tilde{y}_{iN}), x_{iq}]$		$[x_{iq}]$	$[x_{iq}]$	$[x_{iq}]$
GPT-2	26.46 ± 3.35%	5.08 ± 1.18%	0.02 ± 0.02%	3.21 ± 3.26%
+ $\mathbf{v_t}$ Initial FV	52.64 ± 2.83%	22.13 ± 3.06%	1.39 ± 1.65%	17.67 ± 10.42%
+ $\mathbf{v_z}$ **Fine-tuned FV**	**57.74** ± 3.09%	**61.32** ± 4.19%	**60.90** ± 1.80%	**44.52** ± 16.71%
GPT-J	33.76 ± 4.00%	7.50 ± 1.51%	3.44 ± 0.14%	7.28 ± 5.57%
+ $\mathbf{v_t}$ Initial FV	85.55 ± 2.05%	56.75 ± 4.93%	5.21 ± 3.15%	27.82 ± 12.53%
+ $\mathbf{v_z}$ **Fine-tuned FV**	**88.04** ± 1.61%	**87.01** ± 1.60%	**66.42** ± 1.99%	**61.02** ± 15.39%
LLaMA-2	29.16 ± 4.03%	13.28 ± 3.27%	16.86 ± 0.24%	8.15 ± 6.46%
+ $\mathbf{v_t}$ Initial FV	82.99 ± 3.28%	68.45 ± 9.11%	26.03 ± 16.98%	42.57 ± 15.78%
+ $\mathbf{v_z}$ **Fine-tuned FV**	**86.66** ± 3.27%	**88.91** ± 1.76%	**76.43** ± 3.51%	**65.96** ± 14.06%
LLaMA-3.1	17.99 ± 2.23%	2.74 ± 1.59%	23.67 ± 0.19%	11.12 ± 7.85%
+ $\mathbf{v_t}$ Initial FV	70.96 ± 5.13%	65.74 ± 7.55%	9.84 ± 6.00%	32.06 ± 14.93%
+ $\mathbf{v_z}$ **Fine-tuned FV**	**81.38** ± 4.14%	**89.95** ± 1.85%	**79.82** ± 3.69%	**61.11** ± 15.18%

Table 9: Average accuracies and standard errors across the simple-task, question-answer, and SemEval datasets. Compared to the baseline and initial FV intervention, we found that the fine-tuned FV leads to the best accuracies for all prompt types. FV = function vector.

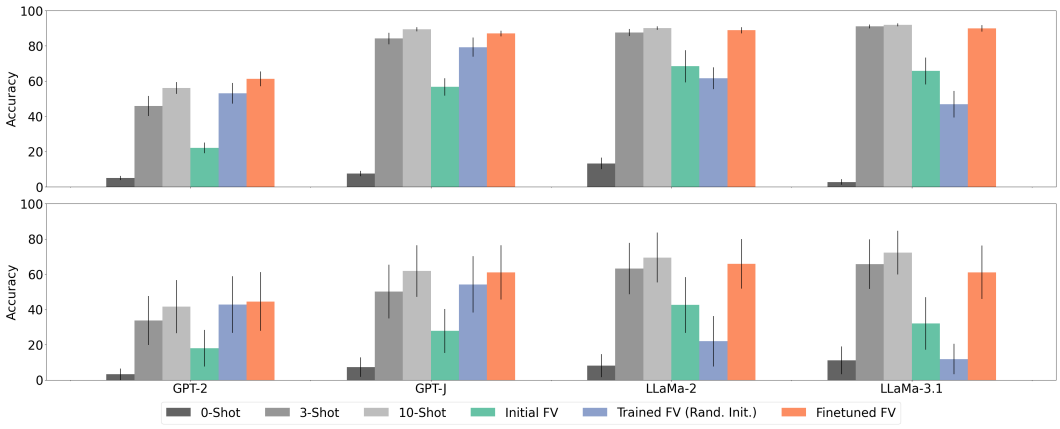


Figure 10: Average results for the Zero-shot evaluation of the simple-task dataset (top) and SemEval dataset (bottom). Compared to both the initial FVs (green) and trained FVs with random initialization (blue), applying the fine-tuned FV (orange) into the zero-shot task led to the best accuracies on average, similar to those for 10-shot learning (light grey). Additionally, the randomly initialized FVs for larger models require more training to maintain accuracies close to the fine-tuned FVs, emphasizing the need for initial FVs to fine-tune from. FV = function vector.

LoRA. We use the same causal language modeling configuration and procedure provided by Cai et al. (2025) across the six simple-task relations from Todd et al. (2024). We only apply LoRA modules to the attention projections (Q, K, V, O), setting the rank to $r = 16$, the scaling factor to $\alpha = 32$, and dropout to 0.2. We train these modules using an AdamW optimizer with $lr = 0.0002$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, weight decay $= 0.002$, and a cosine scheduler with 20% warmup for 5-10 epochs.

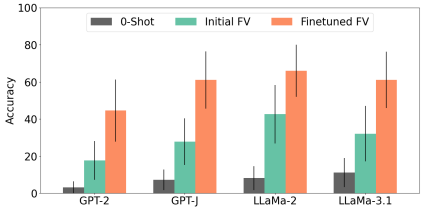


Figure 9: Top-5 prediction accuracy on SemEval for the relation completion task.

LoFiT. Across all four models, we use the configurations provided by Yin et al. (2024). As there are no GPT configurations in the paper, we used the Gemma-7B 3%-

head MQuAKE configuration for GPT-2 and GPT-J (λ=5e-4, learning rate 5e-3, bias tuning learning rate 8e-3). We use the LLaMA-2-7B 3%-head MQuAKE configuration (λ=1e-3, learning rate 5e-3, bias tuning learning rate 1e-2) for both LLaMA-2-7B and LLaMA-3.1-8B. For the FFV comparison we trained it on the same word pairs used for refining the FFVs, while for the AFV comparison we use 2,898 analogies formed with said word pairs.

DyVec. We used the default settings provided by Cai et al. (2025) for DyVec: this means partitioning the LLM to $S = H$ segments, which equates to the total number of attention heads. Rather than selecting only the top-k segments, we took the locations that meet a particular probability threshold after running the REINFORCE algorithm. We run this algorithm with learning rate 2e-4 for 15-25 epochs.

A.3.2 Decoding Function Vectors

To examine whether relation words/phrases can be extracted from the FV, we construct a decoder that runs the input through layer 13 (the layer following FV intervention) and the last layer before decoding its output to the token distribution. While Todd et al. (2024) decoded the FV by itself, we added the attention head values from layer 12 for a blank prompt (i.e. "Q: A: "). As the FV is added to the hidden state of layer 12, there is additional information from the attention values that may lead to a more interpretable distribution when combined with the FV. As shown in Table 10, the resulting tokens either are within the relation's output space or tend to be representative of the relation itself. With the FFV, tokens that relate to the relation's abstract concept rather than its output space tend to be promoted. For example, under the Singular–Plural relation, FFV decoding retrieves the token "plural" whereas the top tokens from decoding the initial FV do not include any relevant relation terms.

Task Name	Model	Decoded Tokens
Antonym	FV	‘ unc’, ‘ empty’, ‘ uncond’, ‘empty’, ‘ attacking’
	FFV	‘ *opposite*’, ‘ unc’, ‘ *oppos*’, ‘ empty’, ‘ uncond’
capitalize_first_letter	FV	‘ *acronym*’, ‘ *initials*’, ‘ *abbre*’, ‘ adjective’, ‘ acron’
	FFV	‘ *acronym*’, ‘ *initials*’, ‘ *abbre*’, ‘ acron’, ‘PN’
Singular-Plural	FV	‘Bs’, ‘PN’, ‘VE’, ‘storage’, ‘phones’
	FFV	‘Bs’, ‘ *plural*’, ‘phones’, ‘ bicycles’, ‘chairs’

Table 10: Top-5 tokens decoded from GPT-J's FV and fine-tuned FV, in the order of decreasing probability. Relation-relevant words are listed in italic. The fine-tuned FV appears to decode to more relation-relevant tokens compared to the FV. For example, the capitalize_first_letter FFV decodes to words associated with capitalizing the first letter, while 'plural' was one of the top-scoring tokens for the Singular-Plural FFV. FV = function vector, FFV = fine-tuned function vector.

A.3.3 Relational Similarity

For the FFVs, we ask whether their relation representations align with human representations of relations better than the initial FVs. Ichien et al. (2022) empirically measured human judgments on relation similarity between word pairs: for example, humans consider the word pair *black:white* to be more relationally similar to *happy:sad* than to *happy:joyful*. To compare the LLM's activity and analyze their relation similarity patterns, we run a dummy relation task where the target query is blank (e.g., *tall : short :: : ?*). We use the same word pairs as the ones used for the two human experiments; for each word pair, we extract the activations of the attention heads in the layer following FV intervention (Layer 13 for our experiments) for the final colon token. By doing so, we can analyze the FV's effects in the model's successive activities.

Figure 3 shows the relational dissimilarity matrices derived from human judgments in the two experiments from Ichien et al. (2022) and an intermediate layer's activations of the baseline (GPT-J), initial function vectors, and fine-tuned function vectors. We measure alignment as the Pearson correlation between the human and model dissimilarity matrices over the off-diagonal cells of each experiment's matrices, with 95% confidence intervals from a bootstrap over the 27 word pairs (5,000 resamples; cells that compare a resampled pair with its own copy are excluded). GPT-J on its

own correlates only weakly to moderately with human judgments ($r = 0.156$ in Experiment 1 and 0.228 in Experiment 2; the two experiments use different relations and word pairs). Both the initial FV and the FFV raise the correlation substantially in Experiment 1 (0.700 and 0.661, statistically indistinguishable), whereas in Experiment 2 the FFV clearly exceeds the initial FV (0.734 vs. 0.542; difference $+0.19$, 95% CI $[0.12, 0.25]$). Across the four models (Table 11), the FFV improves alignment over the unintervened model in every case (all eight bootstrap intervals for the difference exclude zero) and over the initial FV in Experiment 2 for GPT-J, LLaMA-2 and LLaMA-3.1 (for GPT-2 the difference is $+0.10$, $[-0.01, 0.20]$); in Experiment 1 the two are comparable for every model. Every correlation exceeds a permutation null that shuffles the word-pair labels of the model matrix (Mantel test, $p \leq 0.002$). Because both matrices share a block structure by relation type, we also partial out same-relation membership from both: the FFV-injected activations still correlate with the human judgments ($r = 0.48$-0.75 across models and experiments, versus 0.10-0.40 without intervention), so the alignment extends to graded similarity within and across relation types. In addition, we visualized FFVs of all relations from four datasets. As shown in Appendix Figure 12, there is clear clustering that separates syntactic relations from semantic relations, suggesting that FFVs capture fundamental structures of relational knowledge.

When examining the relational similarity across other models, we observe similar patterns for the activations under FV and FFV intervention. With FV intervention on larger-scale models like LLaMA-2 and 3.1, the activations for word pairs within the same relation type tend to exhibit higher cosine similarity, but this is attenuated with FFV intervention. Results for the other three models are in Figure 11.

	Exp. 1		Exp. 2	
	r	95% CI	r	95% CI
GPT-2	.404	[.23, .58]	.278	[.12, .43]
+ $\mathbf{v_t}$ Initial FV	.713	[.53, .85]	.669	[.54, .81]
+ $\mathbf{v_z}$ Fine-tuned FV	**.763**	[.59, .89]	**.770**	[.69, .85]
GPT-J	.156	[−.01, .38]	.228	[.08, .40]
+ $\mathbf{v_t}$ Initial FV	**.700**	[.51, .85]	.542	[.46, .67]
+ $\mathbf{v_z}$ Fine-tuned FV	.661	[.52, .81]	**.734**	[.67, .82]
LLaMA-2	.486	[.33, .68]	.282	[.17, .47]
+ $\mathbf{v_t}$ Initial FV	**.834**	[.66, .94]	.658	[.51, .82]
+ $\mathbf{v_z}$ Fine-tuned FV	.792	[.62, .90]	**.729**	[.62, .84]
LLaMA-3.1	.480	[.29, .67]	.417	[.27, .60]
+ $\mathbf{v_t}$ Initial FV	.812	[.63, .93]	.622	[.43, .79]
+ $\mathbf{v_z}$ Fine-tuned FV	**.818**	[.66, .92]	**.697**	[.55, .82]

Table 11: Pearson correlations between the model's relational dissimilarity matrix and human judgments (Ichien et al., 2022), over the off-diagonal cells of each matrix. 95% confidence intervals are from a bootstrap over word pairs (5,000 resamples, excluding cells comparing pairs with themselves). FV = function vector, FFV = fine-tuned function vector.

A.3.4 t-SNE

To examine the similarity across the FFVs themselves, we generate a t-SNE plot to visualize their clusters. As shown in Figure 12, there is clear clustering that separates syntactic relations from semantic relations. Specifically, we found that the FFVs computed from syntactic relations (from Google and MSR datasets) tend to cluster into one group, while those derived from semantic relations cluster according to their general relation type. This result provides additional evidence that FFVs capture fundamental structures of relational knowledge.

A.4 Composite + Analogical Function Vector Evaluation

While AFVs led to substantially better accuracies on the far-analogy problems in Green's dataset, their effects on the near-analogy problems were not of the same magnitude (first column in Table 12). This is likely due to such analogies being within-domain (e.g., *furnace : coal :: woodstove : wood*),

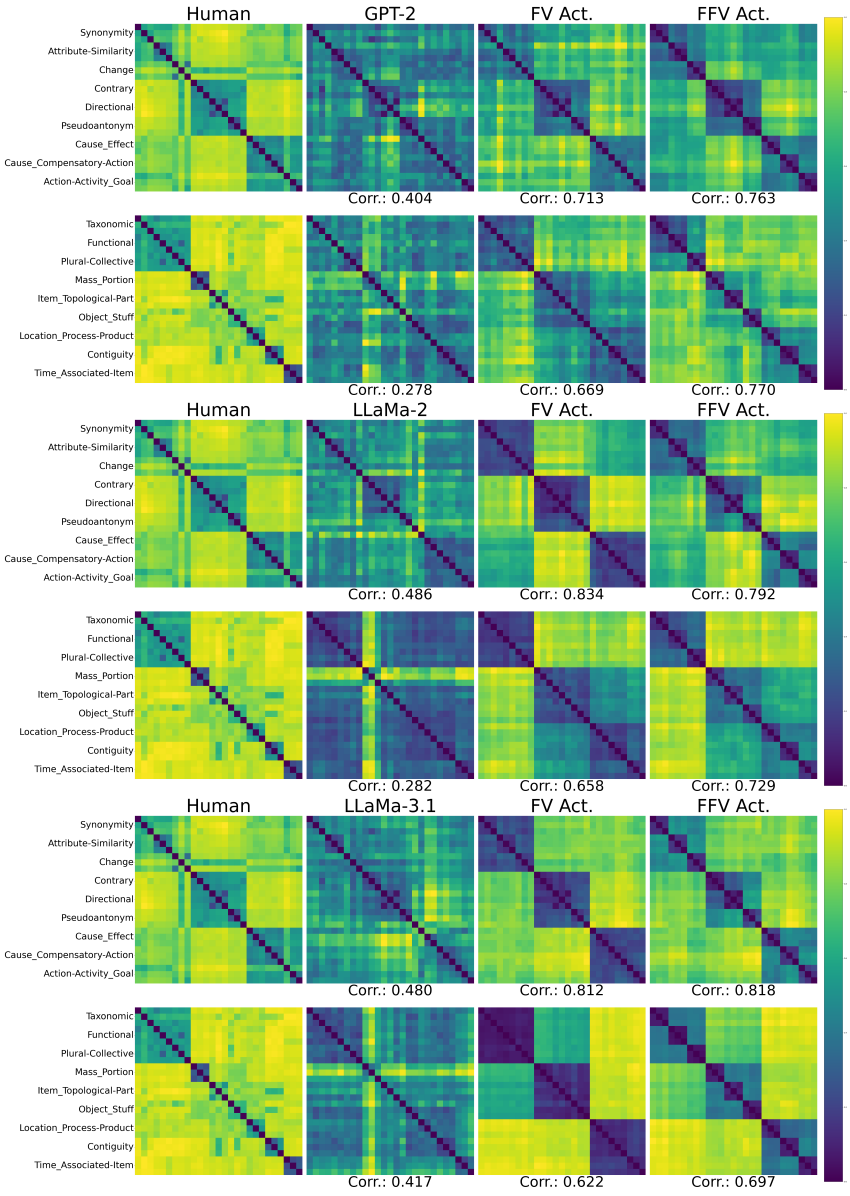


Figure 11: Relational dissimilarity matrices of human judgments and predictions generated by GPT-2 (top), LLaMA-2 (middle) and LLaMA-3.1 (bottom), both alone and with the FVs. In each panel, the top row represents Exp. 1 in Ichien et al. (2022), while the bottom row represents Exp. 2 with word pairs. The two experiments used different sets of relations for measuring human relation similarity judgments. While the activations of the initial FV-injected LLMs are tightly grouped by relation type, the FFV-injected LLMs account for dissimilarities across relations as shown in the similarity matrices. In general, the LLMs' activations tend to align closer to the human judgments with FV intervention, as evidenced by the structure of their similarity matrices and Pearson correlation. FV = function vector, FFV = fine-tuned function vector.

which can be easily solved by humans and LLMs through leveraging rich knowledge of semantic associations. When AFVs are used to steer the LLMs to focus on relational knowledge, alternative strategies based on these associations may be de-emphasized. However, such semantics-based strategies break down for more challenging reasoning scenarios such as the far-analogy problems involving word pairs from different domains (e.g., *furnace : coal :: stomach : food*). The AFV

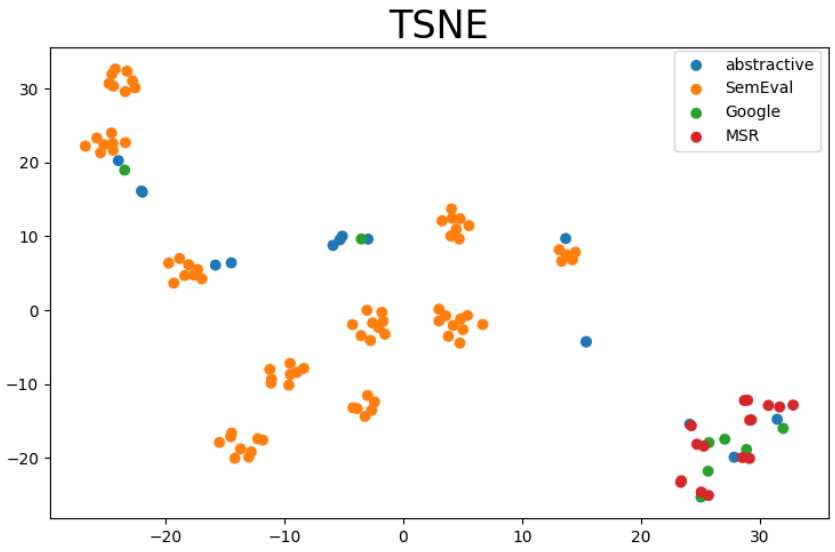


Figure 12: t-SNE of all FFVs computed with GPT-J. We found that the FFVs of syntactic relations (Google and MSR datasets) tend to be clustered together at the bottom right corner, while those of semantic relations (abstractive dataset from Todd's study and SemEval datasets) are clustered in the top left region. FFV = fine-tuned function vector.

	Green's Analogy			
	Near-Analogy	Far-Analogy	SAT Problems	Turney's Narratives
GPT-2	45.00%	20.00%	19.79%	5.61%
+ Comp v	55.00%	32.50%	32.09%	21.39%
+ AFV $\mathbf{v_A}$	**62.50%**	**35.00%**	**34.49%**	**26.47%**
GPT-J	75.00%	42.50%	33.69%	17.91%
+ Comp v	75.00%	**55.00%**	47.86%	18.18%
+ AFV $\mathbf{v_A}$	75.00%	**55.00%**	**51.07%**	**22.46%**
LLaMA-2	70.00%	57.50%	51.87%	43.05%
+ Comp v	67.50%	**70.00%**	59.89%	**48.40%**
+ AFV $\mathbf{v_A}$	67.50%	**70.00%**	**60.43%**	**48.40%**
LLaMA-3.1	85.00%	55.00%	52.67%	48.93%
+ Comp v	**87.50%**	**70.00%**	61.76%	50.80%
+ AFV $\mathbf{v_A}$	**87.50%**	67.50%	**62.03%**	**51.07%**

Table 12: Top-5 accuracy on one-shot four-term analogies from Green's dataset (40 near- and 40 far-analogy problems), the SAT dataset (374 problems) and Turney's narratives (374 problems). Each model row is the baseline with the source pair in context and no intervention; $\mathbf{v}$ = composite vector from relation initialization; $\mathbf{v_A}$ = analogical function vector with weights $< \epsilon = 10^{-3}$ pruned.

compensates the LLM for this limitation with its relational knowledge, enabling the model to solve such analogies regardless of their complexity.

Comparison of the Composite and Analogical Function Vectors. We also see that the AFV is competitive with the composite vector in Turney's dataset by varying degrees, with large improvements for the GPT models (+5.08 pp GPT-2, +4.28 GPT-J) in comparison to the LLaMA models (0.00 pp LLaMA-2, +0.27 LLaMA-3.1). The composition and source domain refinement carry the measured improvements in accuracy, but the source-target alignment forces consistency with analogy solving with no-to-minimal cost in accuracy. The results are provided in the rightmost column of Appendix Table 12.

Robustness to Prompt Templates. To ensure that our observed improvements are not merely artifacts of prompt engineering, we evaluate the AFVs across five prompt templates (detailed in Table 13). Across all templates the AFV's accuracy is $50.59 \pm 3.45\%$, revealing improvements of at least 9.36 pp over one-shot prompting. While the fifth template reduces the one-shot accuracy to 5.61%, the AFV still leads to accuracy gains of +38.24 pp. The composition is the main driver of such improvements as seen in Figure 13 and Table 13. This underscores the robustness of the composite and AFV frameworks, confirming that their capacity to guide analogical reasoning relies on deep structural relational knowledge rather than superficial prompt dependencies.

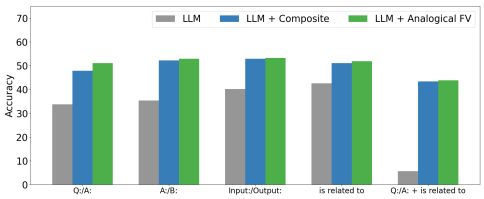


Figure 13: Top-5 accuracy of GPT-J on the SAT analogies under five prompt templates of Table 13.

Templates	GPT-J	LoFiT	Comp	AFV
`Q : {x_S}\nA : {y_S}\n\n` `Q : {x_T}\nA :`	33.69%	63.80%	47.86%	51.07%
`A : {x_S}\nB : {y_S}\n\n` `A : {x_T}\nB :`	35.29% (+1.60)	57.75% (−6.05)	52.14% (+4.28)	52.94% (+1.87)
`Input : {x_S}\nOutput : {y_S}\n\n` `Input : {x_T}\nOutput :`	40.11% (+6.42)	58.61% (−5.19)	52.94% (+5.08)	53.21% (+2.14)
`{x_S} is related to {y_S}` `as {x_T} is related to`	42.51% (+8.82)	53.05% (−10.75)	51.07% (+3.21)	51.87% (+0.80)
`Q : {x_S} is related to {y_S} as` `{x_T} is related to\nA :`	5.61% (−28.08)	27.01% (−36.79)	43.32% (−4.54)	43.85% (−7.22)

Table 13: Analogy prompt templates used to evaluate the AFV's robustness to input forms on the SAT dataset with GPT-J using the same evaluation as Table 12. LoFiT is trained in the `Q :/A :` format and the base FFVs used for the AFV are optimized with `Q :/A :` completion prompts where as the AFV receives no four-term analogy training. Parentheses represent the delta from each column's `Q :/A :` accuracy. Comp = composite vector, AFV = analogical function vector.

A.4.1 Out-of-Distribution Evaluation

For the Leave-One-Type-Out (LOTO) evaluation, we see in Table 14 that accuracy improves for nine of the ten types. Both vector interventions led to improvements over the baseline, indicating that they are not solely driven by relations with easy word pairs. Class-inclusion is the only relation type in which vector intervention degrades the LLM's performance; this may be due to it having the easiest analogy problems, making the LLM solve them easily without vector intervention.

Held-out type	FFVs	n	One-shot	Composite $\mathbf{v}$	AFV $\mathbf{v_A}$	Δ (pp)
Class-inclusion	74	3067	54.55%	44.11%	44.80%	−9.75
Part-whole	69	5452	30.69%	38.63%	38.00%	+7.32
Similar	71	3502	38.78%	43.83%	43.00%	+4.23
Contrast	71	2819	27.81%	45.58%	45.23%	+17.42
Attribute	70	5177	26.23%	38.75%	41.36%	+15.12
Non-attribute	71	4430	10.77%	25.19%	27.56%	+16.79
Case-relation	71	4376	28.75%	37.71%	38.99%	+10.24
Cause-purpose	71	4282	27.67%	40.80%	41.85%	+14.18
Space-time	70	4510	28.07%	37.89%	38.34%	+10.27
Reference	73	3155	23.14%	37.27%	37.46%	+14.33

Table 14: Per-type results for GPT-J with LOTO evaluation. Each row represents the results of the withheld relation type.

For testing the robustness of the out-of-distribution capabilities, we also evaluate the composite and analogical function vectors with a Leave-Two-Types-Out (LTTO) approach. For this evaluation, the

types are split in terms of the SemEval type order as shown in Table 15. While Fold 1 led to the lowest improvements, it was also the fold with the easiest analogy problems as mentioned previously with the Class-Inclusion relation type. The minimal headroom for this would lead to the LLM's ease of solving such analogies like before. The results from the LTTO evaluation showcase how removing an additional relation type does not affect the results significantly, making it robust to grouping.

Held-out types	FFVs	n	One-shot	Composite $\mathbf{v}$	AFV $\mathbf{v_A}$	Δ (pp)
Class-inclusion, Part-whole	64	8519	39.28%	40.83%	40.83%	+1.55
Similar, Contrast	63	6321	33.89%	44.61%	44.31%	+10.43
Attribute, Non-attribute	62	9607	19.10%	31.96%	33.41%	+14.31
Case-relation, Cause-purpose	63	8658	28.22%	39.43%	40.77%	+12.55
Space-time, Reference	64	7665	26.04%	37.53%	37.85%	+11.81

Table 15: Per-fold breakdown of the Leave-Two-Types-Out (LTTO) evaluation on GPT-J. Each fold withholds one pair of relation types from the SemEval vector library and is evaluated only on analogies drawn from those two types.

Oracle-FFV Reconstruction. For every SemEval source pair in this task we compare the composite built while its relation type is withheld against that relation's own FFV. The cosine similarity between the pair's composite vector and the relation's FFV is 0.82-0.85 across the four models on average, implying that the former is capable of approximating the withheld vector effectively. Because all FFVs share a common direction (mean pairwise cosine 0.70-0.74), the composite is not reliably closer than the nearest single base vector; after subtracting the library mean it still exceeds the most-weighted vector and shuffled weights for at least 81% of relations.

A.4.2 Ablation Studies

Ablation on the Vectors used for the Analogical Function Vector. For this study, we note that the composite vector's accuracy is competitive with the AFV for Green's dataset in the larger-scale models. This can be attributed to the model's capability for advanced reasoning with the composite vector such that refining the weights on the source analog may lead to overfitting. Despite this, the AFV-injected LLM demonstrates notable analogy-solving capabilities compared to the LLM. While such improvements from the AFV may be expected due to refining its weights using the source analog, they still showcase its capability in completing the analogy without any prior knowledge of the target's answer.

	Green's Analogy			
	Near-Analogy	Far-Analogy	SAT Problems	Turney's Narratives
GPT-J	75.00%	42.50%	33.69%	17.91%
$\mathbf{v_t}$ Initial FV	70.00%	42.50%	37.97%	18.45%
+ $\mathbf{v_z}$ Fine-tuned FV	**75.00%**	**55.00%**	**51.07%**	**22.46%**

Table 16: Ablation on the function vectors composed into GPT-J's AFV, using the same evaluation as Table 12. The initial-FV library has 49 vectors, as SemEval is represented by its 10 relation types rather than its 79 relations; the fine-tuned library has 118. FV = function vector.

Ablation of the Analogical Function Vector's Relations. To determine if the current relation pool is necessary for the AFV, we run an ablation study that reduces the number of FFVs to $K \in \{30, 49, 60, 90\}$ representatives. The FFVs were grouped through agglomerative clustering before the ones closest to the clusters' centroids were selected for the AFV pipeline. From Table 17, the AFV's performance on the SAT and Turney datasets starts to rise with library size before plateauing at around 90 relations. The smallest library ($K = 30$ relations) was the only one that led to significantly lower accuracies in comparison to the full one (SAT -4.55 pp, $p = 0.005$; Turney -3.21 pp, $p = 0.05$). While the composite vector has minimal changes in accuracy at $K > 30$, refining it to the AFV led to the largest improvements with the full library (+4.28 pp at $K = 118$ vs. +1.87 at $K = 30$). While pruning at the 0.001 threshold reduces the relation space in half, it did not significantly change the conclusions made with the ablation due to removing only 1% of the

initial weight. These results indicate that the AFV's ability to represent unseen relations depends on how much of the relation space its library covers.

	Green's Analogy			
Relations (K)	Near-Analogy	Far-Analogy	SAT Problems	Turney's Narratives
GPT-J (one-shot)	75.00%	42.50%	33.69%	17.91%
Composite $\mathbf{v}$				
30	72.50%	52.50%	44.65%	17.38%
49	70.00%	52.50%	47.33%	**18.72**%
60	70.00%	50.00%	47.86%	**18.72**%
90	70.00%	**55.00**%	**48.66**%	18.45%
118 (full)	**75.00**%	**55.00**%	47.86%	18.18%
AFV $\mathbf{v_A}$				
30	72.50%	50.00%	46.52%	19.25%
49	72.50%	50.00%	48.66%	21.39%
60	70.00%	47.50%	49.20%	21.12%
90	72.50%	52.50%	**51.34**%	22.19%
118 (full)	**75.00**%	**55.00**%	51.07%	**22.46**%

Table 17: Ablation of the AFV's FFV library size K on GPT-J. $K \in \{30, 60, 90\}$ represents around 25%, 50%, and 75% of the AFV's library space respectively, while $K = 49$ represents the initial FV's library size. FFV = fine-tuned function vector, AFV = analogical function vector.