CAN CIRCUIT ANALYSIS PREDICT CAPABILITY TRANSFER ACROSS TASKS AND MODALITIES?

Anonymous authors
Paper under double-blind review

ABSTRACT

Circuit analysis is mainly used post-hoc to explain model behavior, but could it also be used to guide training decisions? We show that a model's internal computation, measured before fine-tuning, can predict how training on one task will affect another. Across four vision-language models from two families and 18 tasks with textual and visual variants, we establish three findings. First, task circuits present in the base model largely persist after fine-tuning, meaning that circuits discovered before training remain representative afterward. Second, relationships between these base-model circuits (measured by cross-task circuit accuracy) can predict capability transfer: our circuit-based measure correlates with cross-task fine-tuning gains up to $r = 0.77$, and is competitive with an established black-box transferability estimation method while capturing non-trivial transfer across task types. Third, we characterize when these predictions succeed and fail, and find they are most reliable when task circuits are well localized, but break down when similar-looking circuits conceal conflicting computations. Together, these findings show how circuits measured once can inform capability transfer, turning circuit analysis from an explanatory tool into a predictive one.[1]

1 INTRODUCTION

A circuit is a subgraph of a model's full computation graph, describing how it solves a specific task (Olah et al., 2020; Elhage et al., 2021). Therefore, circuits are natural objects for reasoning about what a model does and does not share across tasks, and how these tasks interact. Today, circuit analysis is used almost entirely post-hoc, to explain behavior a model has already learned (Wang et al., 2022; Nikankin et al., 2024; Mueller et al., 2025). In this paper we ask: Could circuit analysis be used to predict fine-tuning gains across tasks and modalities?

We study this question through capability transfer—whether training on one task would improve a model's performance on another—a long-standing problem in machine learning (Pan & Yang, 2009; Weiss et al., 2016). Established work on transferability often studies distribution shifts under specific assumptions (Weiss et al., 2016) that do not scale to more complex models and tasks, or only relies on the final layer of the model (Mehra et al., 2024), without taking internal computations into account. In contrast, we take an interpretability-grounded approach and ask whether a model's internal computation, measured *before* fine-tuning, predicts transfer. Our results show that circuits found on the base model predict transfer across tasks and modalities after fine-tuning.

Our approach relies on two observations. First, circuits identified in the base model largely persist after fine-tuning. This holds both in terms of circuit components and their attribution scores, and consistent with previous work (Prakash et al., 2024). In the context of capability transfer, this indicates that circuits found on the base model remain representative of the test task even after training.

Next, we hypothesize that tasks relying on compatible computations will improve one another during fine-tuning, as training on one reinforces computations that are also useful for the other. In contrast, tasks relying on conflicting computations will degrade each other's performance, as training pushes those computations in different directions. Importantly, we do not measure this using circuit overlap alone, since similar functionality can be implemented by different components (Nikankin et al., 2025). Instead, we measure how well functionality transfers between circuits: whether the circuit

[1] Our code and data are available in the supplementary materials and will be made public.

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

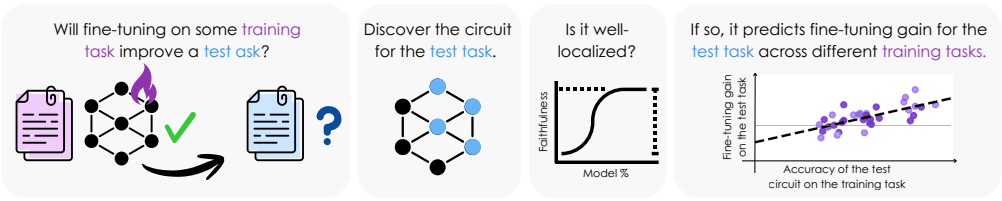


Figure 1: **Overview.** We ask whether task circuits can predict, *before* training, if fine-tuning on some training task will improve its performance on another test task. We discover the test task's circuit on the base model and assess how well-localized it is. Intuitively, a behavior is well-localized if a small subset of the model's components reliably describes it. If the test circuit is well-localized, its accuracy on the train task, measured using our cross-task circuit accuracy (XCA) metric, predicts the gain that fine-tuning on the training yields on the test.

identified for the test task can also perform the training task. We quantify this using *cross-task circuit accuracy*: the accuracy of the test-task circuit when evaluated on the train task.

We find that cross-task circuit accuracy measured on the base model can predict post fine-tuning capability transfer, with correlations of up to $r = 0.77$ (Fig. 1). Our analysis spans four vision-language models—Gemma-3 (4B and 12B) (Team et al., 2025) and Qwen-2.5-VL (3B and 7B) (Team, 2024)—and 18 tasks, each with language and vision-language variants (36 tasks in total). Importantly, this predictive relationship emerges only when the test-task circuit is *well-localized*, as measured by its faithfulness curve (Mueller et al., 2025).

We compare our circuit-based predictions against an established transferability estimation method that treats the model as a black box, reading only its final-layer representations (Mehra et al., 2024). Our method is on par at predicting fine-tuning gains in most cases, and in some cases surpasses it, while additionally surfacing non-trivial, interpretable transfer predictions, like transfer across different task families, which output-based scoring does not.

Our analysis further reveals factors that shape the predictive power of cross-task circuit accuracy. First, vision-language circuits are generally less well-localized than their language-only counterparts. Consequently, although cross-task circuit accuracy remains predictive for well-localized vision-language tasks, its overall predictive power is lower for these variants. We also identify a distinct failure mode when tasks share most of their functionality but differ in a small, critical computation. Cross-task circuit accuracy may then remain high (capturing the shared functionality), missing the difference that determines whether fine-tuning transfers positively or negatively.

In the spirit of actionable interpretability (Orgad et al., 2026), our results show that analyzing a model's internal computation can guide decisions, not only explain behavior, broadening the potential impact of interpretability methods.

2 Related Work

2.1 Localizing model computations

Recent studies aim to improve our understanding of internal computations performed by large neural networks (Ferrando et al., 2024; Mueller et al., 2024). In language modeling, early approaches uncovered linguistic properties in latent representations using probes (Belinkov, 2022). Driven by the need for causality (Antverg & Belinkov, 2021), later approaches aim to identify model components with a causal effect on model performance (Meng et al., 2022; Geiger et al., 2025). A recent line of work moved from individual components to entire computation paths, i.e., circuits (Elhage et al., 2021; Wang et al., 2022; Mueller et al., 2025). In language models, functionally similar circuits were shown to have both notable node overlap and high cross-task faithfulness (Olsson et al., 2022; Merullo et al., 2023; Mondorf et al., 2025). In contrast, VLMs were shown to use different components when performing equivalent tasks across modalities, yet in some cases performing equivalent and interchangeable computations (Nikankin et al., 2025). In this work, we extend this measure of functional equivalence and show it can be used to predict cross-task capability transfer.

2.2 Fine-tuning and model internals

A closely related question is how circuits persist under training. Across pre-training and scale (Tigges et al., 2024) and across fine-tuning (Prakash et al., 2024), prior work finds that the algorithm a circuit implements is largely preserved even as its components shift, and Sun (2025) links this stability to generalization. In contrast, Chen et al. (2026) argue that circuits evolve freely under fine-tuning, so mechanisms localized on the static base model become stale. Our results support the former view, and we use it to predict the effect that fine-tuning will have on the task. In language models, Leng & Xiong (2025) identify task-specific neurons that facilitate generalization and catastrophic forgetting, and Zhang et al. (2024) localize a compact "core linguistic region" facilitating cross-lingual transfer, and show that freezing it during further pre-training mitigates catastrophic forgetting. At a finer grain, Song et al. (2025) link out-of-distribution generalization to composition and induction-heads, arguing that a shared latent subspace bridges the components that must compose for a rule to transfer. Concurrent work with this paper shows that shared circuits are predictive of a model's ability to generalize to different task formats (de Varda et al., 2026). Together, these results suggest that where a capability lives in the network, and how much that location is shared across tasks, carries signal about transfer.

2.3 Transferability Estimation

A vast body of work is dedicated to transfer learning (Pan & Yang, 2009; Weiss et al., 2016). Early approaches analyze transferability under specific distribution-shift assumptions, such as covariate shift (Ben-David et al., 2006; Kumar et al., 2022) and label shift (Le et al., 2021), yielding guarantees that are difficult to extend to modern models. A separate line computes empirical transferability scores (Xue et al., 2024), reading a model's output representations or logits, either statically from pre-trained features (Nguyen et al., 2020; You et al., 2021; Gholami et al., 2023) or dynamically, by approximating fine-tuned features without fine-tuning (Li et al., 2023; Khoba et al., 2025; Zheng et al., 2026). These methods read only a model's final-layer outputs and treat its internal computation as a black box, which becomes critical to take into account as tasks and models grow in scale and complexity. Similar to our setting, Mehra et al. (2024) analyze transferability via task-relatedness, yielding an upper bound on fine-tuning gains, connecting transfer to task similarity. Like static methods, we utilize signal computed before fine-tuning to predict transferability. However, we measure the model's internal computation rather than solely relying on its output representations, providing an interpretability-grounded perspective on transfer.

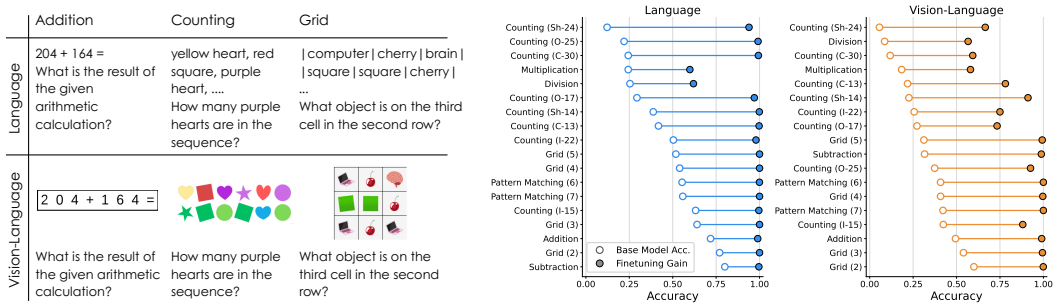
3 Preliminaries

We focus our analysis on two model families, analyzing two sizes of models from each: Gemma-3 4B and 12B (Team et al., 2025) and Qwen-2.5-VL 3B and 7B (Team, 2024).

3.1 Tasks and Datasets

We construct a dataset of 18 tasks, from four task families (examples in Fig. 2a):

- **Arithmetic Tasks:** Following Nikankin et al. (2025), the model is given a two-operand 3-digits arithmetic calculation (e.g., "156 + 223"), either as text or as a white-background image. We treat each operation as a unique task (addition, subtraction, multiplication and division).
- **Counting:** The model is given a sequence of objects and asked to count colors (e.g., "how many green objects are in the sequence?"), shapes (e.g., "how many stars ..."), full objects combinations (e.g., "how many green stars ..."), or icons, representing more complex object (e.g., "how many rockets ..."). A task is defined by the number of objects it has, and the query type: **C**olor, **Sh**ape, **O**bjects or **I**cons.
- **Grid:** Following Wang et al. (2024), the model is given a sequence of icons arranged in a grid, and asked which object is in a specific cell. Tasks differ by their grid size.
- **Pattern Matching:** The model is given a list of letters showing some pattern (e.g. "ABABA") and asked to output the next letter in the sequence based on the pattern (e.g. "B"). Tasks differ by the length of the pattern prefix given to the model.



(a) Task formats.

(b) Base vs. finetuned accuracy on Gemma-3-4b.

Figure 2: **Experimental Setup: Task examples and model performance. (a)** Example inputs for a subset of the tested tasks. Each task has a language and a vision–language variant. **(b)** Accuracy of the base model (open circles) and the gain after fine-tuning (filled circles), for language (blue) and vision–language (orange) tasks on Gemma-3-4b.

Examples for all tasks with full prompts are shown in Appendix A. For each task, we generate $5,000$ samples with corresponding textual and visual data, and split them into $4,000/500/500$ training/validation/test splits.

3.2 Circuit Discovery and Evaluation

A circuit (Elhage et al., 2021) is a minimal subset of interconnected model components that perform the computations required for a specific task or prompt. We follow Nikankin et al. (2024; 2025) and define a component as either an entire attention head or an individual multilayer perceptron (MLP) neuron, at a specific input position.

For each task, we sample a discovery set of 150–500 samples from the training set and an evaluation set of 100 samples from the validation set. We pair each sample with a random counterfactual that yields a different answer. For each component we compute an attribution score, with higher values indicate greater component importance for the analyzed task. Concretely, we compute attribution patching with integrated gradients (AP-IG) (Nanda, 2022; Hanna et al., 2024), which can be calculated once per input for all model components in parallel. We take the final attribution score of a component as the average score across all discovery prompts. For a task t, we construct a circuit C_t^p by selecting the top-p highest-scoring components for different values of p. We evaluate each C_t^p by its faithfulness (Wang et al., 2022), the proportion of the full model's task performance explained by the circuit. Finally, we take the minimal p such that C_t^p achieves a faithfulness of over 80%, denoting this circuit simply as C_t.

3.3 Fine-tuning

To fine-tune models on the various tasks, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2022) to all weight matrices of the language model, including the MLP up, gate, and down projections, and attention query, key, value and output projections, in each layer. The vision encoder stays frozen.

We perform a task-specific hyper-parameter search, optimizing the learning rate, weight decay, and LoRA rank. We select the best configuration based on performance on the validation set; see Appendix B for details. For each task and optimal configuration, we train 5 models using different random seeds. Fine-tuning performance is taken as the average across these seeds. The base model performance and fine-tuning gains from same-task training in Gemma-3-4B are shown in Figure 2b. The remaining models are shown in Appendix B.

4 Predicting Transfer Using Cross-Task Circuit Accuracy

We aim to use circuits discovered in the base model to predict how fine-tuning transfers between tasks. For this to be possible, we first establish that the circuits we analyze are largely preserved by

fine-tuning. If so, a circuit discovered once in the base model should remain representative of the task in the fine-tuned model. Given this stability, studying a test task's base-model circuit and its interactions with candidate training tasks should let us predict transfer without training. Then, we quantify how much one task's circuit contributes to another's computation using *cross-task circuit accuracy*, and show that it predicts the gain that fine-tuning on different candidate training tasks yields on the test task.

4.1 Circuits Remain Unchanged During Fine-Tuning

Before circuits can predict properties of fine-tuning, they must be stable enough that a circuit found on the base model still describes the fine-tuned one. In line with prior work (Prakash et al., 2024), we find that fine-tuning largely preserves circuits.

On a representative subset of our tasks (half the tasks in each family), we use the optimal fine-tuning configuration for one model from each family, Qwen-2.5-VL-3B and Gemma-3-4B, and recover circuits at a range of training steps, from early updates through convergence. We compare each circuit to the one found in the base model along two axes: component overlap (intersection over union; IoU), and preservation of component importance (Spearman correlation between node attribution scores; Chen et al. 2026), each at the top 5% and top 20% of components.

Across all tested tasks, we observe the same trajectory: circuits shift over the first few updates (before roughly step 50, out of a total of $1,000$) and then stabilize for the remainder of training. Once stabilized, the top 20% of components reach 60–90% IoU with the base-model circuit, and their attribution scores remain highly correlated. The top 5% shows more variation, but taken together, these results suggest that components and their ranks remain mostly stable across fine-tuning. Full curves are given in Appendix C.

4.2 Cross-Task Circuit Accuracy

Next, we hypothesize that if tasks share similar functionality, training on one will improve the other. Since circuits capture the functionalities the model implements to solve a task, they offer a natural way to test this. Critically, models were shown to implement similar functionality using different components, appearing as low component overlap (Nikankin et al., 2025). Thus, we test functional overlap by applying the circuit of one task to the other, and measuring its performance. In our experiments, we study the capability transfer from a training task s to a test task t. Thus, the object we want to study is the test circuit, evaluating it on s. In Appendix E we detail additional metrics we considered, including component recall and evaluating the training task circuit on the test tasks.

Since circuits assign components to specific input positions, we map between the positions of each pair of tasks. We build on previous studies (Kaduri et al., 2024; Nikankin et al., 2025) and divide the circuit into three position types: data positions, query positions, and the generation position, which is the last token position, and apply a many-to-many mapping per position type (see Appendix D). For a training task s and a test task t with circuit C_t, we denote the mapped circuit as $C_{t\to s}$.

To compare circuit performance across different tasks, we opt to use circuit *accuracy* instead of *faithfulness*, which is not normalized by the full model performance on the task. Concretely, for a training task s with prompts $x_i \in X$ and gold answers $y_i \in Y$, we compute:

$$Accuracy_t(s) = \frac{\sum_i \mathbb{1}[C_{t\to s}(x_i) = y_i]}{|X|}, \tag{1}$$

where $C_{t\to s}(x_i)$ denotes the top token computed by the model when only computing the mapped circuit components, and mean ablating the rest (using s). Since the many-to-many mapping may increase circuit size, we normalize by subtracting the accuracy of a random circuit of the same (mapped) size, denoted $R_{t\to s}$. Altogether, we define cross-task circuit accuracy (XCA) gain as:

$$XCA_t(s) = Accuracy(C_{t\to s}) - Accuracy(R_{t\to s}). \tag{2}$$

4.3 Cross-Task Circuit Accuracy Predicts Fine-Tuning Gains

To understand the relationship between cross-task circuit accuracy and capability transfer, we compute its correlation with fine-tuning gains. We denote a model M fine-tuned on a training task s as

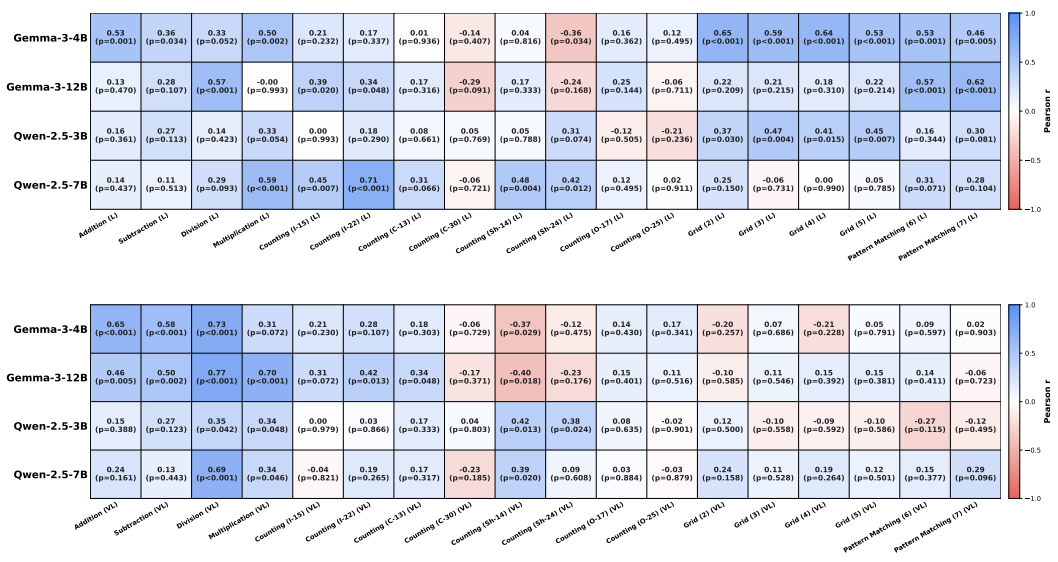


Figure 3: Correlation between our circuit-based metric $XCA_t(s)$ and fine-tuning gains across all 36 tasks and four models. Each cell reports the correlation r for one task–model pair. Top: language tasks; bottom: vision-language tasks. The metric attains high correlations for language tasks and, in vision-language, mostly for the arithmetic family. Counting tasks correlate more weakly across most models.

M_s. We denote a model's base accuracy on a test task t as $M(t)$ and its accuracy on the test task t after fine-tuning on s, as $M_s(t)$. The fine-tuning gain for task t after training on task s is defined as: $FT_s(t) = M_s(t) - M(t)$. For a test task t, we compute its gain across each training task s ($s \neq t$), $FT_s(t)$, and correlate it with the cross task circuit accuracy of C_t on s, $XCA_t(s)$.

Figure 3 shows the correlation values for all 36 tasks across the four analyzed models. Overall, our circuit-based metric achieves high correlation values for the language variations of the tasks (e.g., $r = 0.57$ for division in Gemma-3-12B, $r = 0.65$ for grid (2) in Gemma-3-4B and $r = 0.62$ for pattern matching (7) in Gemma-3-12B). In vision-language variations, our method shows very high correlations for the arithmetic task family (e.g., $r = 0.77$ for Division and $r = 0.70$ for Multiplication in VL in Gemma-3-12B), but only a limited success in other families. Across both modalities and in three out of the four models, we observe that the object counting tasks exhibit lower correlation values compared to other tasks.

Figure 4 shows examples of a positive relationship between $XCA_t(s)$ and fine-tuning gains across different samples. Our circuit-based metric reveals several non-trivial transfer predictions. First, our method orders tasks within a family and predicts how tasks in the same family transfer to one another. In Gemma-3-4B, for addition (VL) we predict that multiplication training yields the largest fine-tuning gains, while for subtraction (VL) we predict division, matching the fine-tuning gains in practice. For pattern matching and Grid tasks in the language modality, XCA predicts that arithmetic training tasks will improve performance across all models, in addition to predicting same-family gains. Our metric also captures modality-dependent transfer: for arithmetic tasks in VL, in Gemma models, our metric predicts that other VL arithmetic tasks yield larger gains than their language counterparts, whereas in Qwen models, XCA predicts language arithmetic transferring more strongly.

4.4 Comparison to Task-Relatedness Estimations

We compare our circuit-based predictions against an established transferability estimation method that is closest to our setting, relying on task-relatedness (Mehra et al., 2024). To compute task-relatedness $\Gamma(s, t)$ between a test task t and a candidate training task s, we collect the activations of both tasks at the final token and final layer of the model, denoted as h_s and h_t, and define $\Gamma(s, t)$ as:

$$\Gamma(s, t) := \|\mu_s - \mu_t\|_2^2 + \lambda \, \|\Sigma_s - \Sigma_t\|_2^2, \tag{3}$$

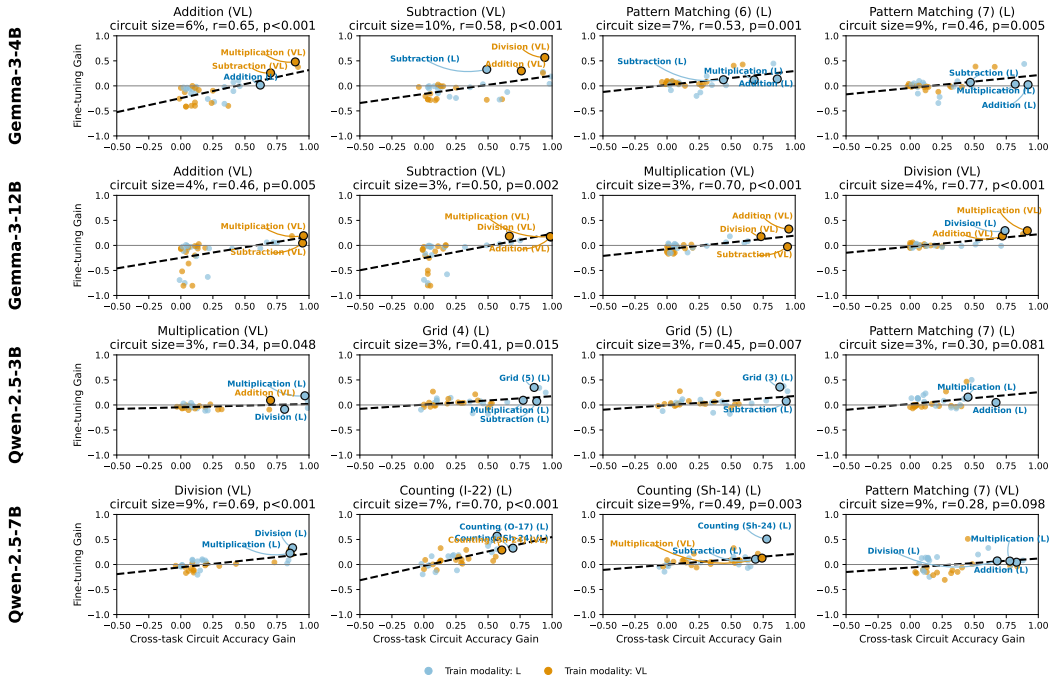


Figure 4: Examples where our metric successfully predicts cross-task transfer. Each panel plots circuit-based metric $XCA_t(s)$ against fine-tuning gain on task t from training on task s; the positive trend shows that higher predicted cross-circuit alignment corresponds to larger gains. Beyond this overall trend, XCA captures several non-trivial patterns: it orders tasks within a family (e.g. in Gemma-3-4B, predicting multiplication as the strongest source for addition (VL) and division for subtraction (VL)), and captures cross-domain transfer, with arithmetic training tasks predicted to improve pattern-matching and Grid tasks in the language modality across all models.

where:

$$\mu_{s/t} := \frac{1}{n}\sum h_{s/t}, \Sigma_{s/t} := \frac{1}{n}\sum (h_{s/t} - \mu_{s/t})^T (h_{s/t} - \mu_{s/t}) \tag{4}$$

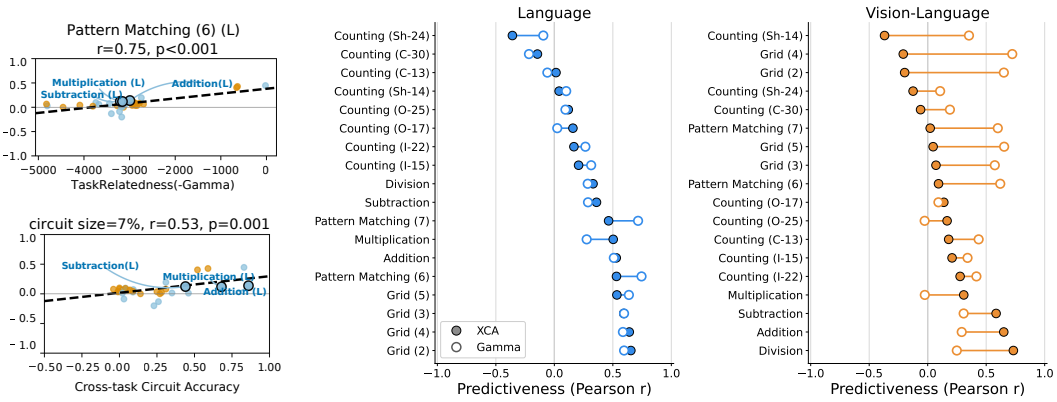
λ is a regularization coefficient, set to 0.001 as in Mehra et al. (2024). n is the number of collected representations. We use $n = 500$, which is the same as the maximal number of prompts used for circuit discovery, to allow a fair comparison. To easily compare between the methods, we take the correlation of $-\Gamma(s,t)$ against fine-tuning gains.

Figure 5 reports, for each task in Gemma-3-4B, the Pearson correlation between a method's score and the observed fine-tuning gain; a filled marker denotes our method (XCA) and an open marker denotes task-relatedness (Γ), with rightward markers indicating better prediction of gains. The two methods are broadly comparable at our prompt budget in the language setting. The vision–language (VL) setting is more split: Γ is the stronger predictor on the grid and pattern-matching tasks, whereas XCA is preferable for arithmetic tasks.

The remaining models are shown in Appendix G. On Gemma-3-12B XCA predicts gains better on a majority of tasks, again most decisively for arithmetic under VL. On the Qwen-2.5 models, XCA is a weaker predictor overall and tracks Γ less closely, although in some cases our circuit-based method still shows better predictiveness.

5 Analysis and Limitations of Circuit-based Predictions

While cross-task circuit accuracy is predictive of fine-tuning gains for many tasks, it is not universally so. Figure 6(a) shows two such cases in Gemma-3-4B where the correlation between $XCA_t(s)$ and fine-tuning gains is low or negative ($r = -0.20$ and $r = 0.09$). This raises a natural question:



(a) (b)

Figure 5: **Circuit-based predictiveness (XCA) vs. task-relatedness (Γ) on Gemma 3 4B.** (a) Predicting fine-tuning gains on *Pattern Matching (6)* (language) from task-relatedness ($-\Gamma$, top) and our circuit score (XCA, bottom). Γ correlates more strongly overall, but XCA successfully predicts gains from the arithmetic tasks. (b) Prediction comparison across all tasks. Horizontal position gives the Pearson correlation between each method's score and the observed fine-tuning gain (rightward = better). Filled = XCA, open = Γ. In language, the methods are broadly comparable, while in vision–language we observe some variation.

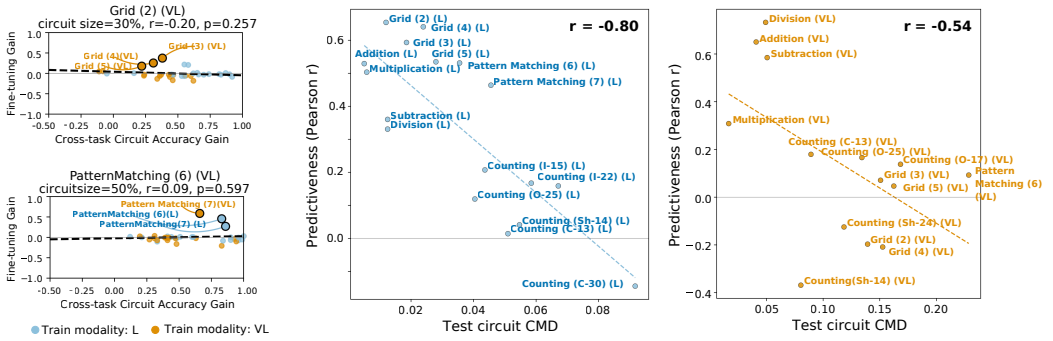
what distinguishes the test tasks for which our circuit metric is predictive from those for which it is not? In the next section, we carry out a *second-order* analysis to better understand the conditions that lead to successful predictions, and the limitations of the circuit-based approach.

5.1 Prediction vs. Localization

We begin with circuit *localization*: the extent to which the model's behavior on a task can be attributed to a small subset of its components. A standard tool for assessing localization is the *faithfulness curve*, which plots faithfulness against circuit size. A well-localized behavior yields a curve that rises steeply and approaches $f = 1$ at small percentages of model components, whereas a poorly localized behavior requires a large fraction of components before recovering model performance. Following Mueller et al. (2025), we compute two metrics to measure localization: the *circuit performance ratio (CPR)*: CPR $= \int_0^1 f(C_k)\,dk$, the area under the faithfulness curve, for which higher values indicate a better-localized circuit; and the *circuit–model distance (CMD)*: CMD $= \int_0^1 |1 - f(C_k)|\,dk$, the area between the curve and $f = 1$, for which lower values (0 is best) indicate a better-localized circuit.

For each test task t, we treat the first-order correlation between $XCA_t(s)$ and fine-tuning gains as the *predictiveness* of the test circuit, C_t, and compute the correlation between predictiveness and its CMD and CPR scores. Figure 6(b) shows this second-order correlation with the CMD score for each modality for Gemma-3-4B. As a lower CMD score indicates a more well-localized circuit, we see a strong negative correlation in language tasks ($r = -0.8$) and a milder but negative correlation for vision-language variants ($r = -0.54$). In Qwen-2.5-VL-3B, CPR correlates more strongly than CMD. As higher CPR score indicates better localization, we observe a positive correlation, with $r = 0.70$ for language and $r = 0.31$ for vision-language.

In both models, vision-language circuits are less well-localized than their language counterparts, with lower CPR and higher CMD scores. We hypothesize that this stems from the model's overall weaker performance on the visual variant: where the algorithm is less robust, the mechanism underlying it may be correspondingly less well defined. Consequently, although XCA remains predictive for well-localized vision-language tasks, predictiveness is lower for vision-language tasks overall. Additionally, we observe that in larger models from both families, the correlations to circuit-localization metrics weaken, as all tasks become more well-localized with scale.



(a) (b)

Figure 6: **Predictiveness of the circuit metric vs. circuit localization (Gemma-3-4B). (a)** Two failure cases: Grid (2) and Pattern Matching (6), both in VL, where the cross-task circuit accuracy XCA is essentially uncorrelated with fine-tuning gains. **(b)** Second-order analysis: for every test task we plot its *predictiveness*, the first-order correlation between XCA and fine-tuning gains, against the test circuit's *integrated circuit-model distance* (CMD), for L (left) and VL (right). Predictiveness decreases with CMD: XCA is most reliable when the test circuit is well localized.

5.2 Conflicting Tasks May Have Similar Circuit Functionalities

Next, we identify a second failure mode: tasks that are superficially similar but require conflicting computations can still show high cross-task circuit accuracy. This is shown by the counting tasks family, where three of the variations ask about different attributes of simple colored shapes (e.g., a purple heart, a red star): **Sh**apes asks to count shapes alone (e.g., how many stars), **C**olors for a given color (e.g., how many red objects), and full **O**bjects for a shape–color conjunction (e.g., how many red stars). Although all variants share a counting operation, they require attending to different, and at times conflicting, features of the input. Figure 7 shows this conflict on the object counting variation with 25 objects. We observe different object counting variations, like shapes and colors, sit at high cross-task circuit accuracy yet yield negative fine-tuning gains.

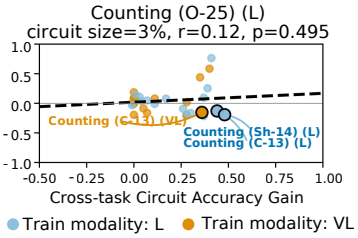


Figure 7: **Conflicting tasks may share functionality.** On object counting, conflicting variations (**Sh**apes, **C**olors) attain high XCA but negative fine-tuning gains.

6 Conclusion

We asked whether circuit analysis, usually applied post-hoc, can guide training decisions, and studied this through capability transfer. We establish three findings. First, fine-tuning largely preserves task circuits: circuits shift over the first few updates, then stabilize, so a circuit discovered once on the base model remains representative after training. Second, we introduced cross-task circuit accuracy (XCA), which evaluates a test task's circuit on a candidate training task. XCA correlates strongly with the realized fine-tuning gain (up to $r = 0.77$), predicts the ordering of source tasks within a family, and surfaces interpretable cross-family and cross-modality transfer. Against an output-based task-relatedness method (Mehra et al., 2024), XCA is on par at predicting fine-tuning gains, making non-trivial predictions that the black-box method misses.

We further examine the second-order correlation of XCA's predictiveness, and find it reliable when the test circuit is well-localized, and weaker for vision-language tasks, fitting these models' lower performance and consistently less-localized circuits on these variations. Lastly, we identify a failure mode in which tasks share some mechanisms but diverge in a small, critical computation, yielding high XCA alongside negative transfer. Taken together, these findings show that a model's internal computation, measured before fine-tuning, carries actionable signal about how training on one task transfers to another.

AI USE STATEMENT

We used generative AI tools to draft and edit parts of this paper and to write parts of the experiment software; all AI-generated code and text was reviewed and verified by the authors. Generative AI was not used for any tasks requiring disclosure under the ICLR policy (e.g., developing theory, proving claims, designing methodology, or interpreting results). We take responsibility for the final content of this work.

ETHICS STATEMENT

This work does not involve human subjects, and no personally identifiable or sensitive data was collected or used. We do not foresee any direct negative societal impacts, conflicts of interest, or fairness, privacy, or security concerns arising from this research.

REPRODUCIBILITY STATEMENT

We detail our setup in section 3 and our method in Section 4 of this work. Full technical details, hyper-parameters, and design choices appear in our code and datasets, which are included in the supplementary materials will be made public. All models in this work are open-weights and are publicly available.

REFERENCES

Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models. *arXiv preprint arXiv:2110.07483*, 2021.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Hang Chen, Jiaying Zhu, Hongyang Chen, Hongxu Liu, Xinyu Yang, and Wenya Wang. Navigating by old maps: The pitfalls of static mechanistic localization in llm post-training. *arXiv preprint arXiv:2605.06076*, 2026.

Andrea Gregor de Varda, Sana Pandey, Pengrui Han, Jacob Andreas, and Evelina Fedorenko. Shared circuits predict whether llms generalize across formats in arithmetic reasoning. *arXiv preprint arXiv:2609.04463*, 2026.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*, 2024.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, et al. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

Mohsen Gholami, Mohammad Akbari, Xinglu Wang, Behnam Kamranian, and Yong Zhang. Etran: Energy-based transferability estimation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18567–18576. IEEE, 2023.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models. *arXiv preprint arXiv:2411.17491*, 2024.

Prafful Kumar Khoba, Zijian Wang, Chetan Arora, and Mahsa Baktashmotlagh. Feature space perturbation: A panacea to enhanced transferability estimation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1299–1308. IEEE, 2025.

Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Certifying model accuracy under distribution shifts. *arXiv preprint arXiv:2201.12440*, 2022.

Trung Le, Tuan Nguyen, Nhat Ho, Hung Bui, and Dinh Phung. Lamda: Label matching deep domain adaptation. In *International Conference on Machine Learning*, pp. 6043–6054. PMLR, 2021.

Yongqi Leng and Deyi Xiong. Towards understanding multi-task learning (generalization) of llms via detecting and exploring task-specific neurons. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 2969–2987, 2025.

Xiaotong Li, Zixuan Hu, Yixiao Ge, Ying Shan, and Ling-Yu Duan. Exploring model transferability through the lens of potential energy. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5406–5415. IEEE, 2023.

Akshay Mehra, Yunbei Zhang, and Jihun Hamm. Understanding the transferability of representations via task-relatedness. *Advances in Neural Information Processing Systems*, 37:116513–116546, 2024.

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.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Circuit component reuse across tasks in transformer language models. *arXiv preprint arXiv:2310.08744*, 2023.

Philipp Mondorf, Sondre Wold, and Barbara Plank. Circuit compositions: Exploring modular structures in transformer-based language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14934–14955, 2025.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, et al. The quest for the right mediator: Surveying mechanistic interpretability through the lens of causal mediation analysis. *arXiv preprint arXiv:2408.01416*, 2024.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fiotto-Kaufman, Tal Haklay, Michael Hanna, et al. Mib: A mechanistic interpretability benchmark. *arXiv preprint arXiv:2504.13151*, 2025.

Neel Nanda. Attribution patching: Activation patching at industrial scale, 2022. URL `https://www.neelnanda.io/mechanistic-interpretability/attribution-patching`.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International conference on machine learning*, pp. 7294–7305. PMLR, 2020.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. *arXiv preprint arXiv:2410.21272*, 2024.

Yaniv Nikankin, Dana Arad, Yossi Gandelsman, and Yonatan Belinkov. Same task, different circuits: Disentangling modality-specific mechanisms in vlms. *arXiv preprint arXiv:2506.09047*, 2025.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Hadas Orgad, Fazl Barez, Tal Haklay, Isabelle Lee, Marius Mosbach, Anja Reusch, Naomi Saphra, Byron Wallace, Sarah Wiegreffe, Eric Wong, et al. Interpretability can be actionable. *arXiv preprint arXiv:2605.11161*, 2026.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Nikhil Prakash, Tamar Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *International Conference on Learning Representations*, volume 2024, pp. 8057–8082, 2024.

Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. Out-of-distribution generalization via composition: a lens through induction heads in transformers. *Proceedings of the National Academy of Sciences*, 122(6):e2417182122, 2025.

Alan Sun. Circuit stability characterizes language model generalization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9025–9040, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. Llm circuit analyses are consistent across training and scale. *Advances in neural information processing systems*, 37:40699–40731, 2024.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 75392–75421. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/89cc5e613d34f90de90c21e996e60b30-Paper-Conference.pdf`.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2022.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

Yihao Xue, Rui Yang, Xiaohan Chen, Weibo Liu, Zidong Wang, and Xiaohui Liu. A review on transferability estimation in deep transfer learning. *IEEE Transactions on Artificial Intelligence*, 5(12):5894–5914, 2024.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International conference on machine learning*, pp. 12133–12143. Pmlr, 2021.

Zhihao Zhang, Jun Zhao, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Unveiling linguistic regions in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6228–6247, 2024.

Yaoyan Zheng, Huiqun Wang, Nan Zhou, and Di Huang. Implicit modeling for transferability estimation of vision foundation models. *Advances in Neural Information Processing Systems*, 38: 36718–36745, 2026.

A Data Examples

Task Family	Language Prompt Example	Vision-Language Prompt Example
Arithmetic	Question: 005+004. What is the result of the given arithmetic calculation? Answer with only a single number:	<image> What is the result of the given arithmetic calculation? Answer with only a single number:
Counting	red star, red star, blue star, blue star, blue star, red star, red star, blue star, red star, blue star, red star, orange star, blue star. How many blue objects are in the sequence? Answer with only a single number:	<image> How many blue objects are in the sequence? Answer with only a single number:
Grid	Consider a 2x2 grid (2 rows and 2 columns) containing various objects/icons, where each 1x1 cell is considered a block and each block contains an object. The distribution of objects in the grid is as follows: hat,heart hat,heart What is the object of the block located at the first row, first column of the grid? Answer with only a single object:	<image> What is the object of the block located at the first row, first column of the grid? Answer with only a single object:
Pattern Matching	D A X I D A What is the next letter in the sequence? Answer with only a single letter:	<image> What is the next letter in the sequence? Answer with only a single letter:

B Fine-Tuning Details and Results

To select the optimal fine-tuning configuration per-task, we search over different learning rates $lr \in [5e-5, 1e-4, 2e-4, 3e-4]$, weight decay values $wd \in [0.0, 0.001, 0.01, 0.1]$ and LoRA ranks $r \in [4, 8, 16]$. The LoRA scaling factor is taken as twice the rank. We train for up to $1,000$ steps with a batch size of 4, and evaluate every 100 for early stopping. The optimal configuration for each task is given in the supplementary material.

We report the base model performance, as well as the mean gain after fine-tuning, and the standard deviation across seeds for all models and tasks in Figure 8.

C Circuits across Fine-Tuning

Figures 9 and 10 give the per-task stability curves for Qwen-2.5-VL-3B and Gemma-3-4B. At each step $s \in [10, 20, 30, 40, 50, 100, 200, 400, 600, 800, 1000]$, we compare the circuit recovered at that fine-tuning step to the base-model circuit, along two axes: component IoU, and the Spearman correlation between node attribution scores, and two thresholds: 5% and 20% of components.

across all tasks and both models, the curves share the same shape: a shift over the first few updates followed by a plateau.

D Position Mapping

A circuit C is formally defined as a set of components:

$$C = \{(l, i, p) \mid l \in \mathbb{N}, i \in \mathbb{N}, p \in \mathbb{N}\}$$

where each component is represented by its layer (l), index (i), and position (p). Each component represents either an attention head (where i denotes the attention head index within layer l) or an MLP neuron (where i denotes the neuron index within layer l).

When decomposing a circuit into its data (C^D), query (C^Q), and generation (C^G) sub-circuits, the circuit components are partitioned according to their position indices, such that $C = C^D \cup C^Q \cup C^G$. Formally, this decomposition is defined as:

$$C^D = \{(l, p, i) \in C \mid p \in P_D\}$$
$$C^Q = \{(l, p, i) \in C \mid p \in P_Q\}$$
$$C^G = \{(l, p, i) \in C \mid p \in P_G\}$$

where P_D, P_Q, and P_G are disjoint sets of position indices corresponding to the data, query, and generation regions respectively. In our case, the generation position is always defined as the last token position of the input. These positions are manually defined for each task, and are available in the supplementary materials.

In order to map a circuit from task t with position indices (P_D^t, P_Q^t, P_G^t) to a different task s with position indices (P_D^s, P_Q^s, P_G^s). We apply a many-to-many mapping within each position type, such that:

$$C_{t\to s}^D = \{(l, p_s, i) \mid (l, p_t, i) \in C_t, p_t \in P_D^t, p_s \in P_D^s\} \tag{5}$$

With query and generation mapping defined similarly. Together, the mapped circuit $C_{t\to s}$ is defined as: $C_{t\to s} = C_{t\to s}^D \cup C_{t\to s}^Q \cup C_{t\to s}^G$.

E Additional Circuit Metrics

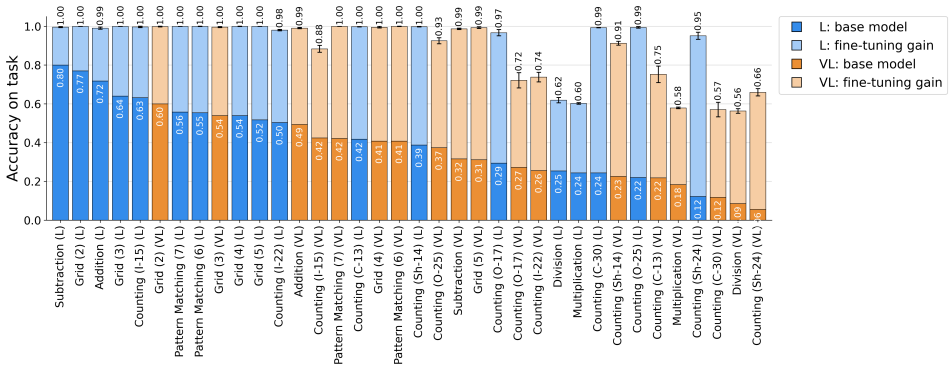
In addition to $XCA_t(s)$ as defined in Section‘4, we experimented with additional metrics: cross-task circuit accuracy from the training task to the test task ($XCA_s(t)$) and component recall (intersection of components divided by the total number of components in the test task circuit). The comparison is given in Figure 11. Overall, we observe that $XCA_t(s)$ gives the highest correlations for the largest number of tasks. For some of our failure cases, discussed in 5, specifically less localized vision-language tasks, $XCA_s(t)$ and recall do achieve high correlations.

F Full Cross-task circuit accuracy results

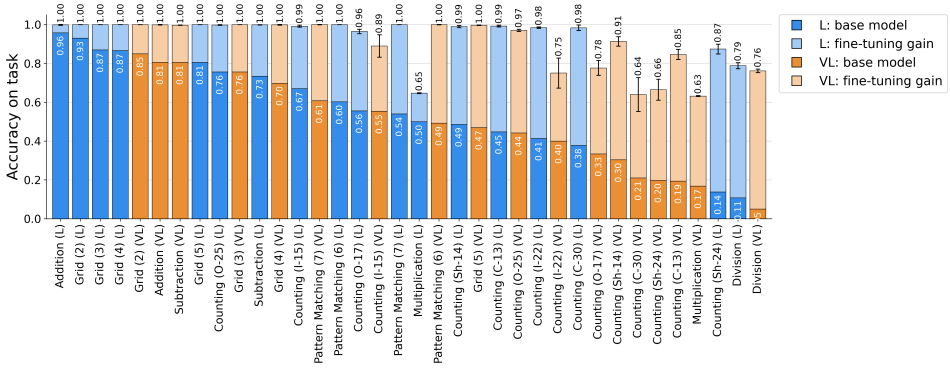
Figures 12–15 plot cross-task circuit accuracy against fine-tuning gains for all tasks across all models.

G Task Relatedness Results

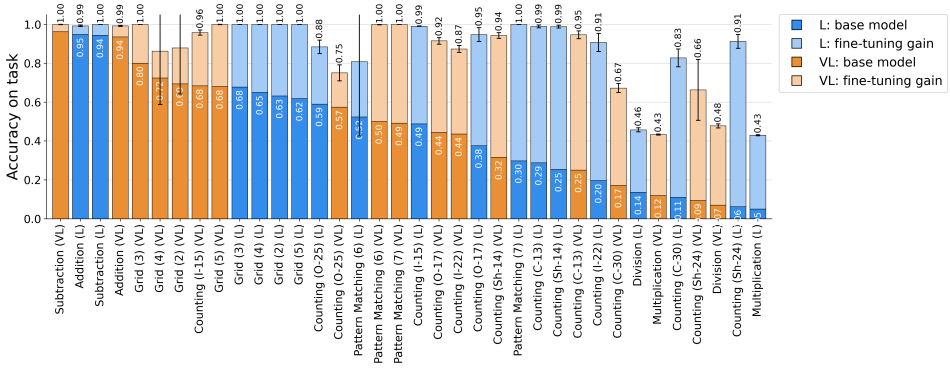
Figures 16–19 plot task relatedness against fine-tuning gains for all tasks across all models. Figures 20–22 compared between the correlation achieved by task relatedness and XCA.



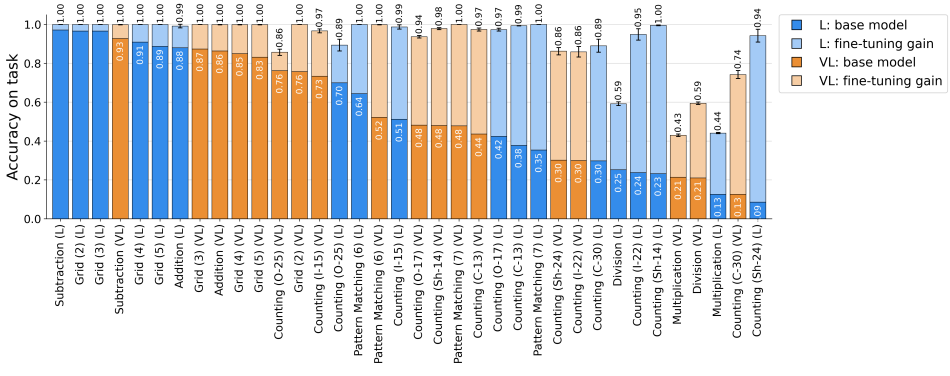
(a) Gemma-3-4B



(b) Gemma-3-12B

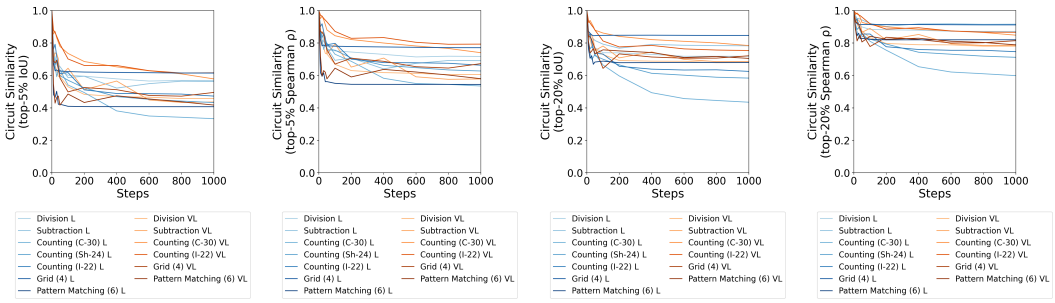


(c) Qwen-2.5-VL 3B



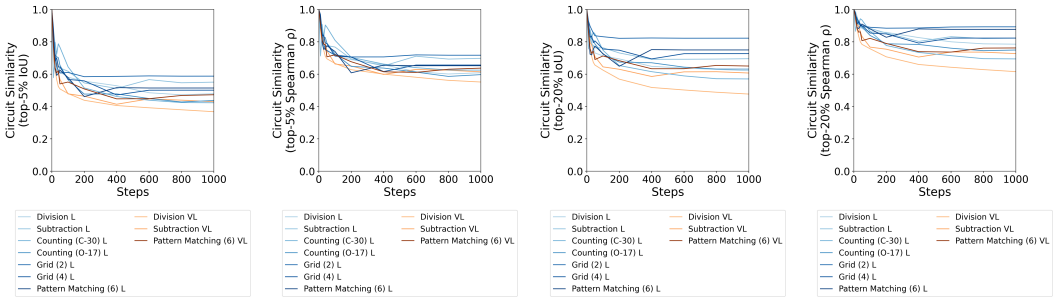
(d) Qwen-2.5-VL 7B

Figure 8: Base accuracies and same-task fine tuning gains across all tasks and models.



(a) Top 5% IoU (b) Top 5% Spearman ρ (c) Top 20% IoU (d) Top 20% Spearman ρ

Figure 9: **Qwen-2.5-VL-3B.** Circuits change early in fine-tuning, then stabilize. Language-only tasks are shown in blue and vision–language tasks in orange.



(a) Top 5% IoU (b) Top 5% Spearman ρ (c) Top 20% IoU (d) Top 20% Spearman ρ

Figure 10: **Gemma-3-4B.** Circuits change early in fine-tuning, then stabilize. Language-only tasks are shown in blue and vision–language tasks in orange.

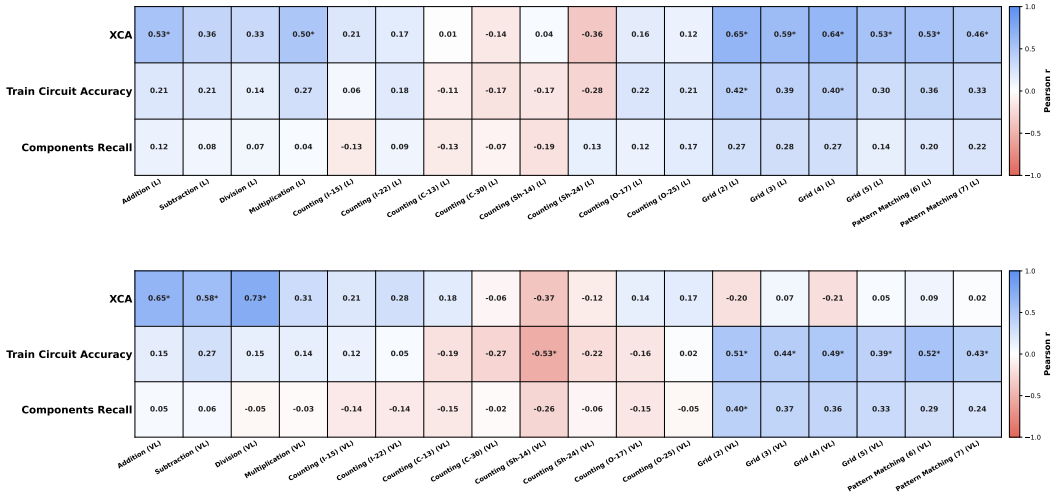


Figure 11: Comparison between different metrics for transfer predictions. Overall, XCA works best across the largest number of tasks.

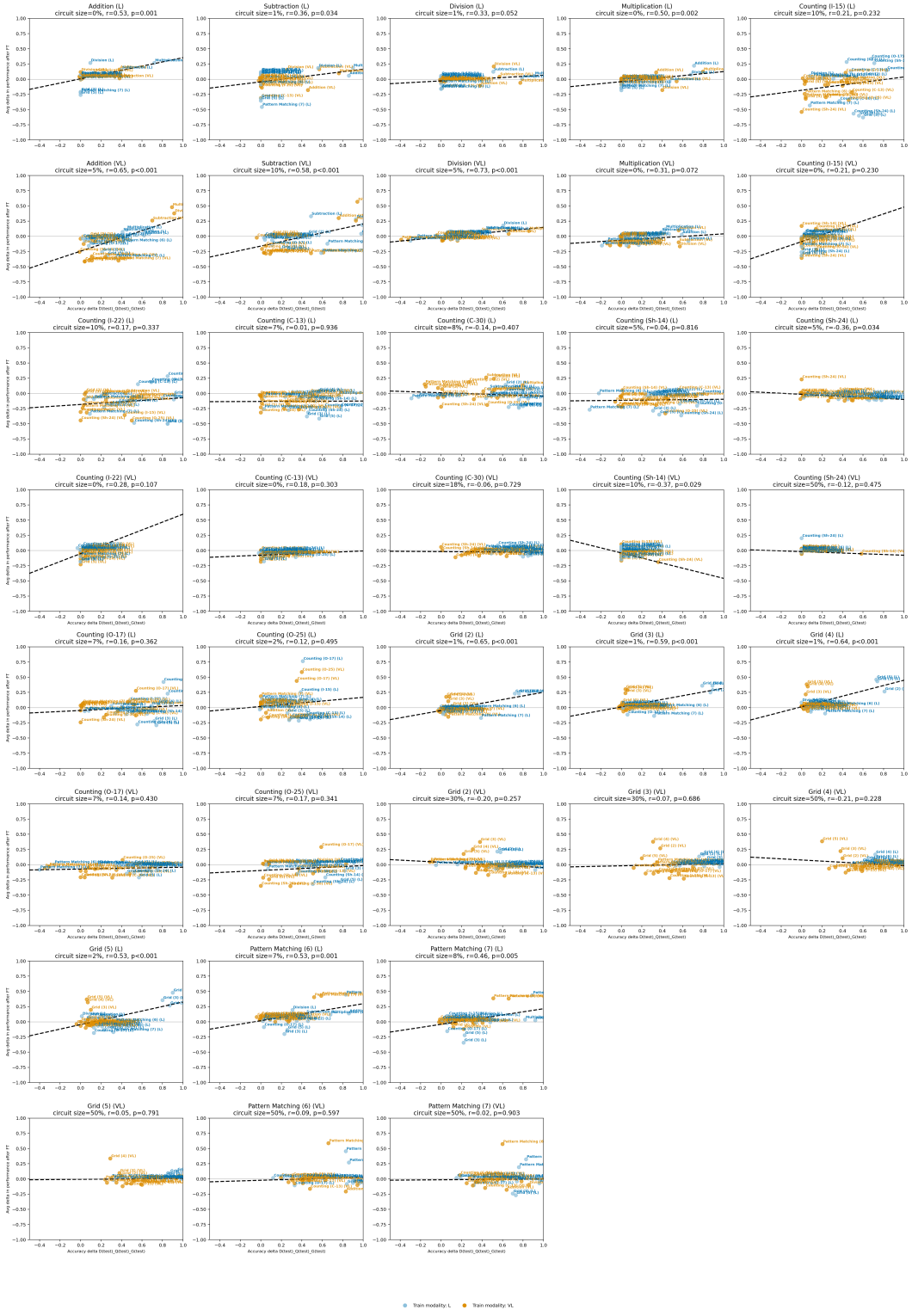


Figure 12: Gemma-3-4B. Circuit-based metric $XCA_t(s)$ against the realized fine-tuning gains.

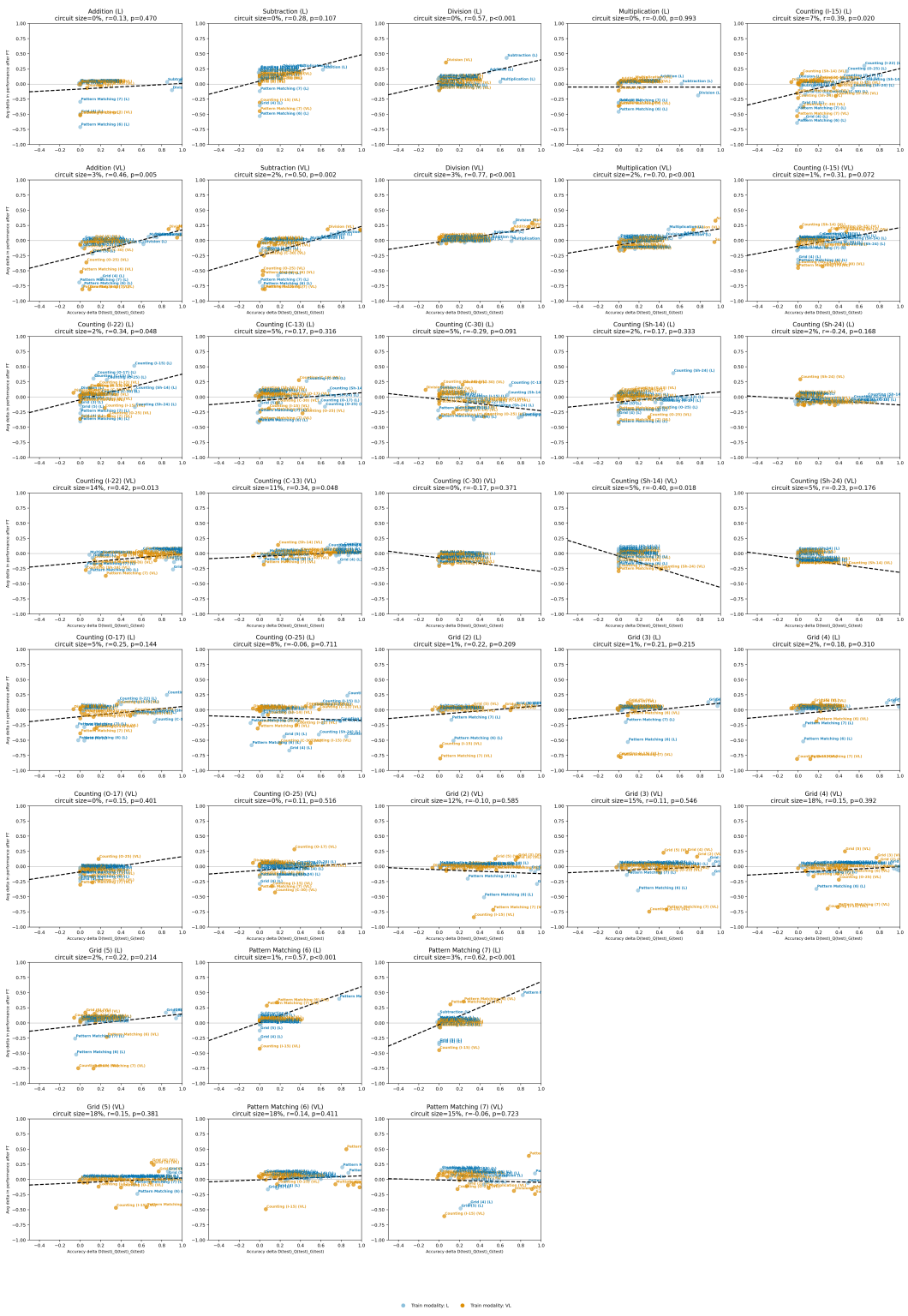


Figure 13: Gemma-3-12B. Circuit-based metric $XCA_t(s)$ against the realized fine-tuning gains.

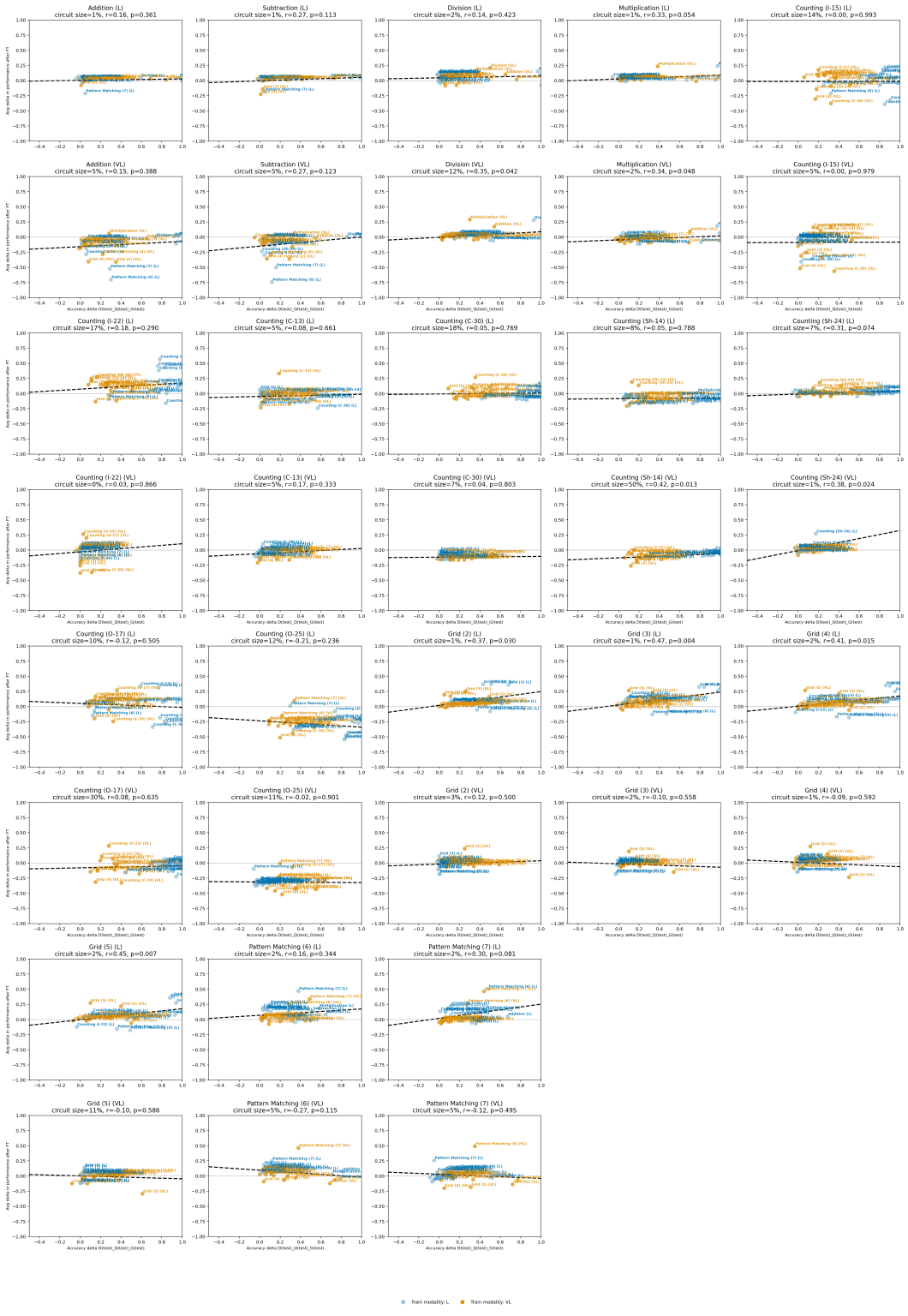


Figure 14: Qwen-2.5-VL-3B. Circuit-based metric $XCA_t(s)$ against the realized fine-tuning gains.

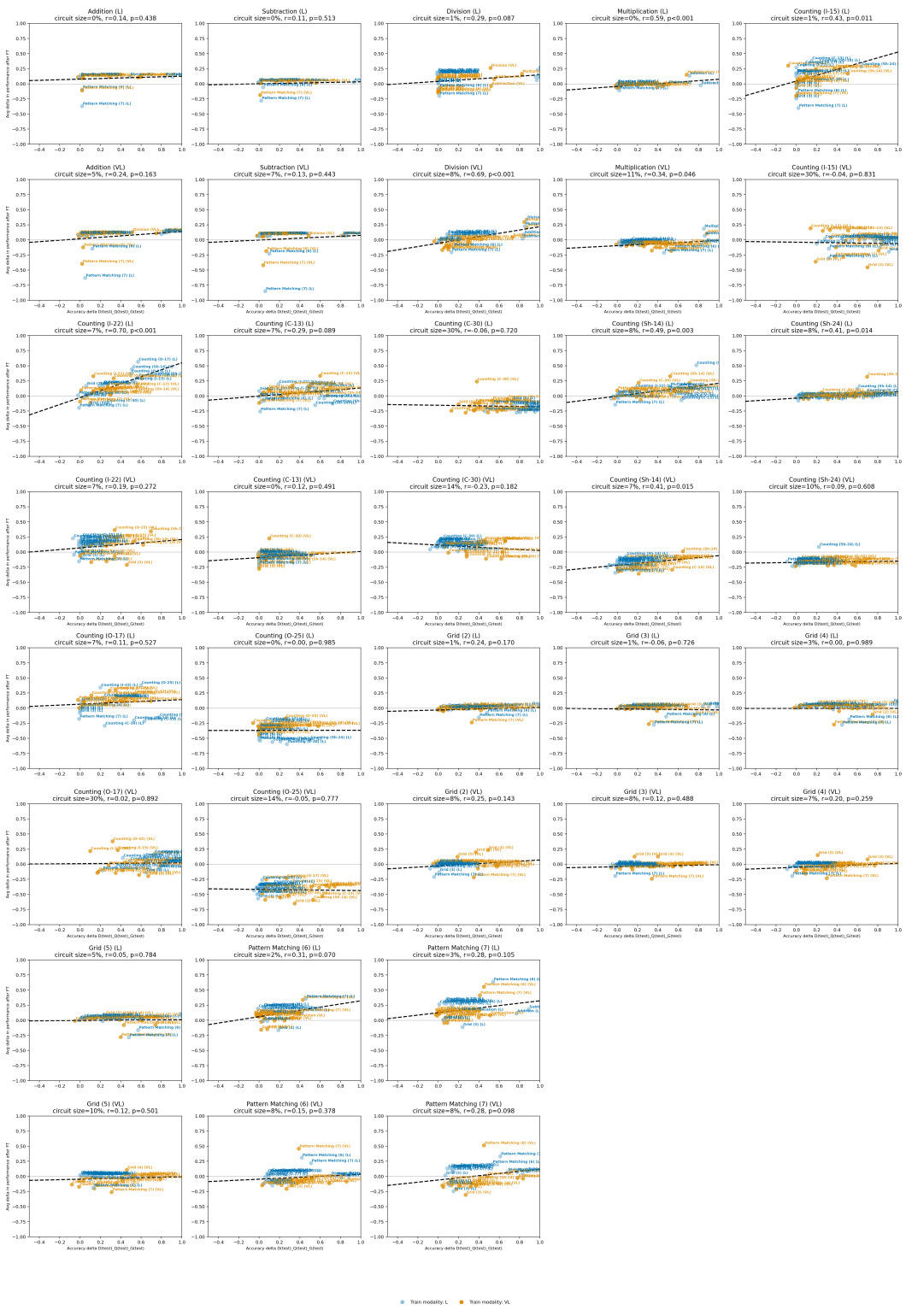


Figure 15: Qwen-2.5-VL-7B. Circuit-based metric $XCA_t(s)$ against the realized fine-tuning gains.

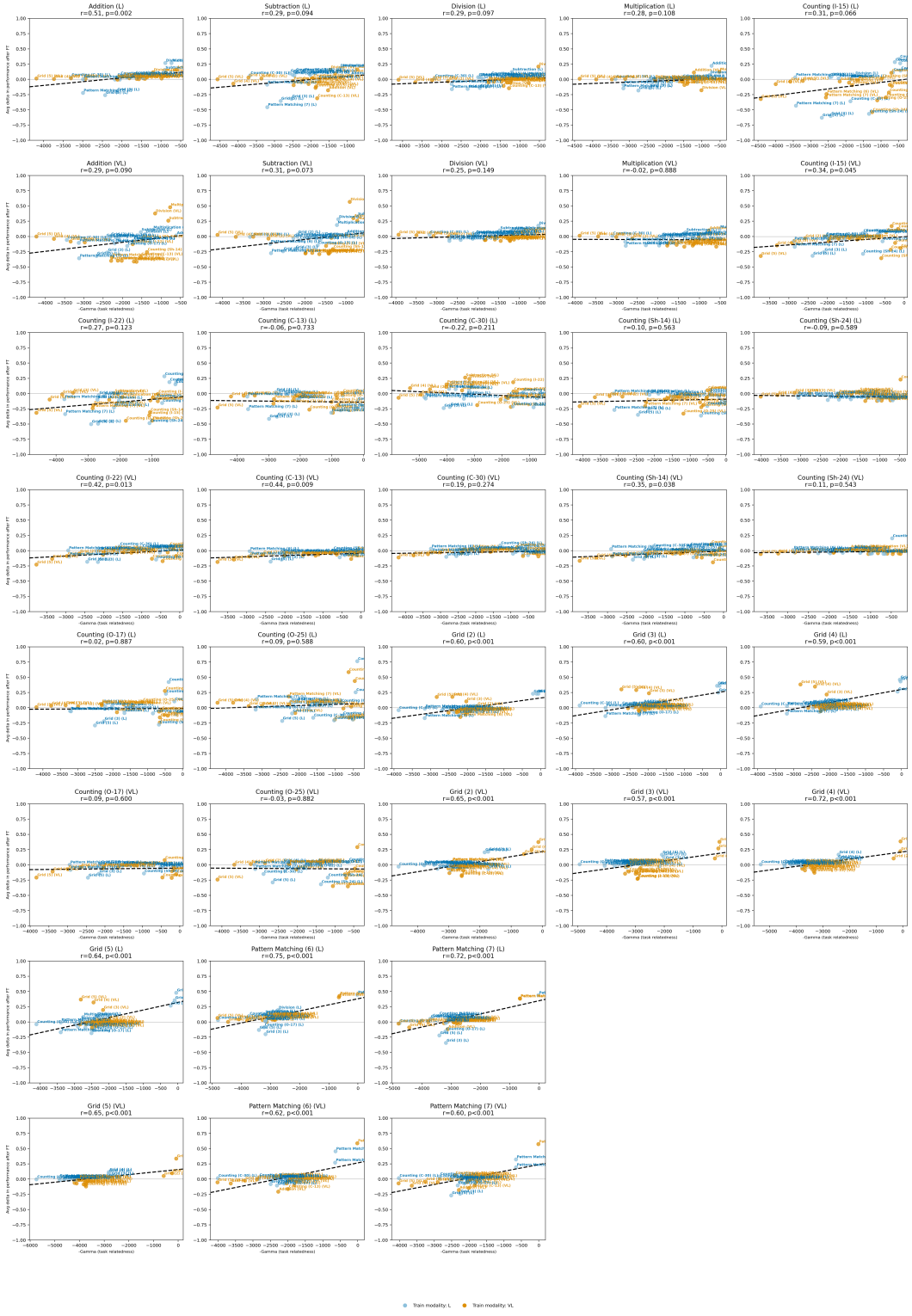


Figure 16: Gemma-3-4B. Task relatedness against the realized fine-tuning gains.

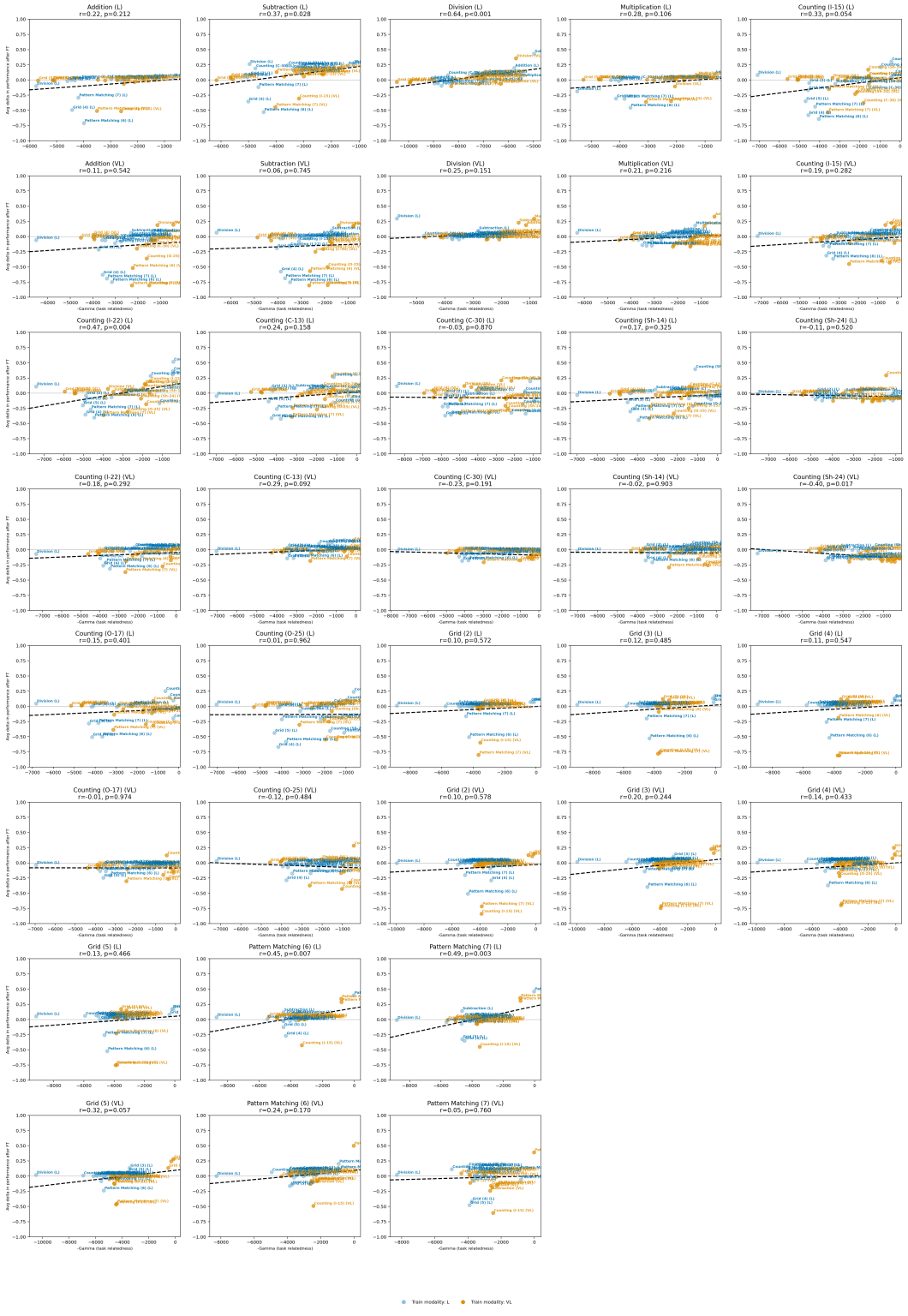


Figure 17: Gemma-3-12B. Task relatedness against the realized fine-tuning gains.

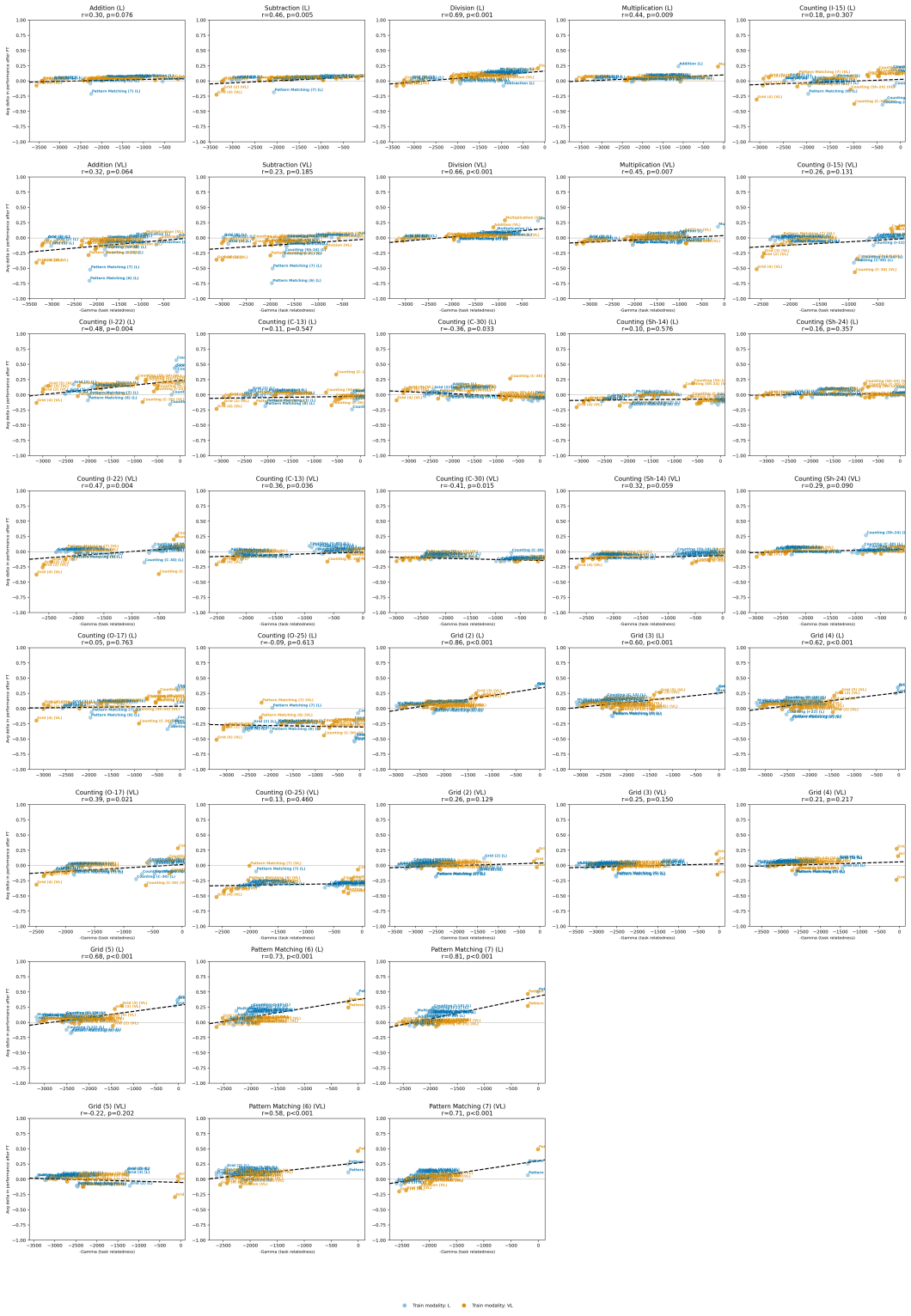


Figure 18: Qwen-2.5-VL-3B. Task relatedness against the realized fine-tuning gains.

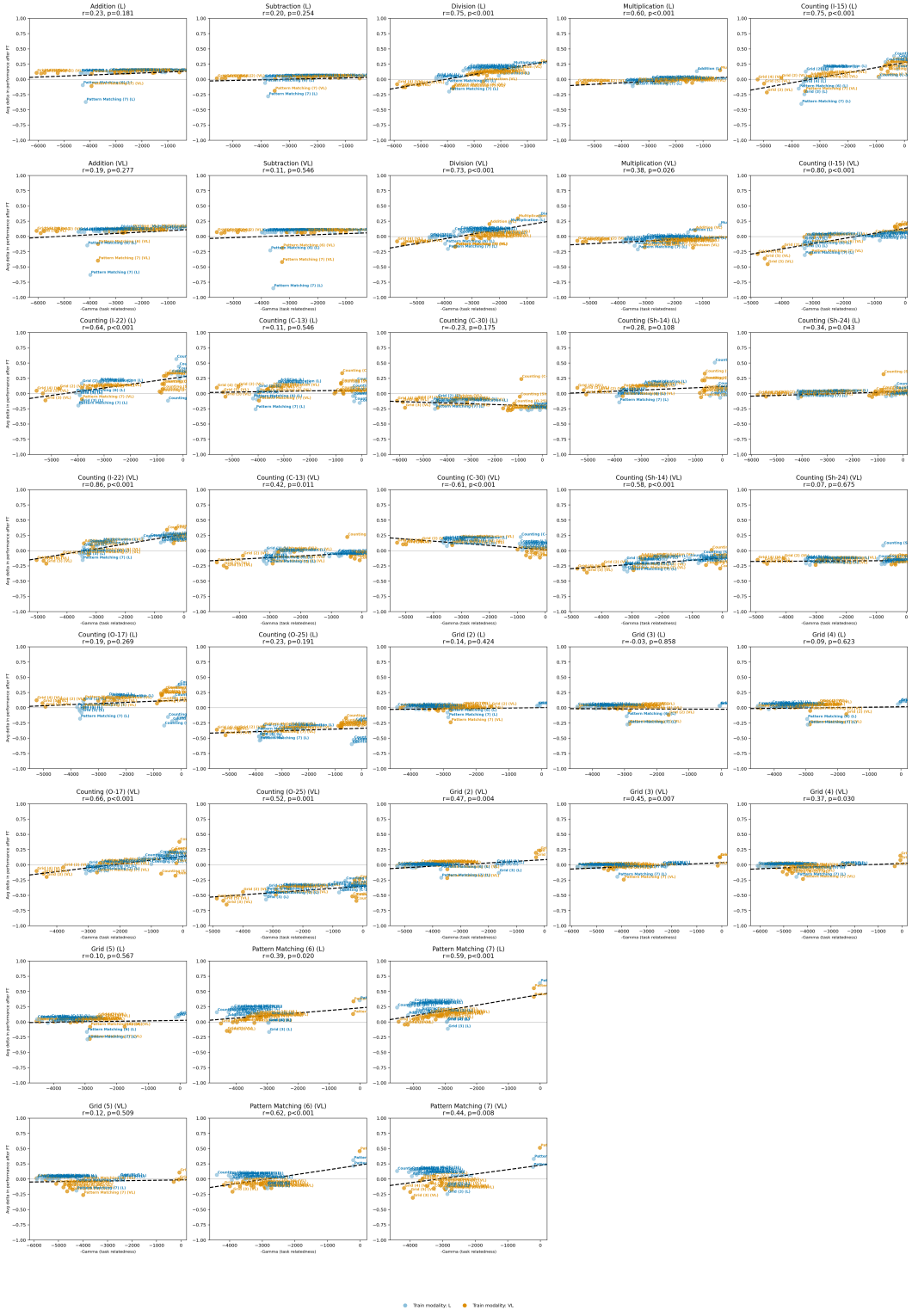


Figure 19: Qwen-2.5-VL-7B. Task relatedness against the realized fine-tuning gains.

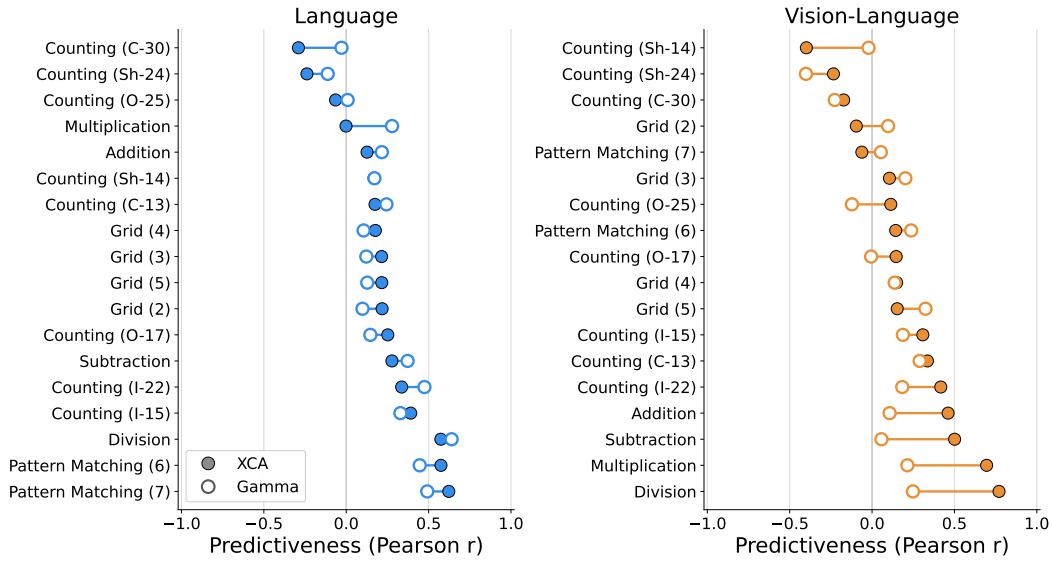


Figure 20: Gemma-3-12B. Task relatedness compared to XCA.

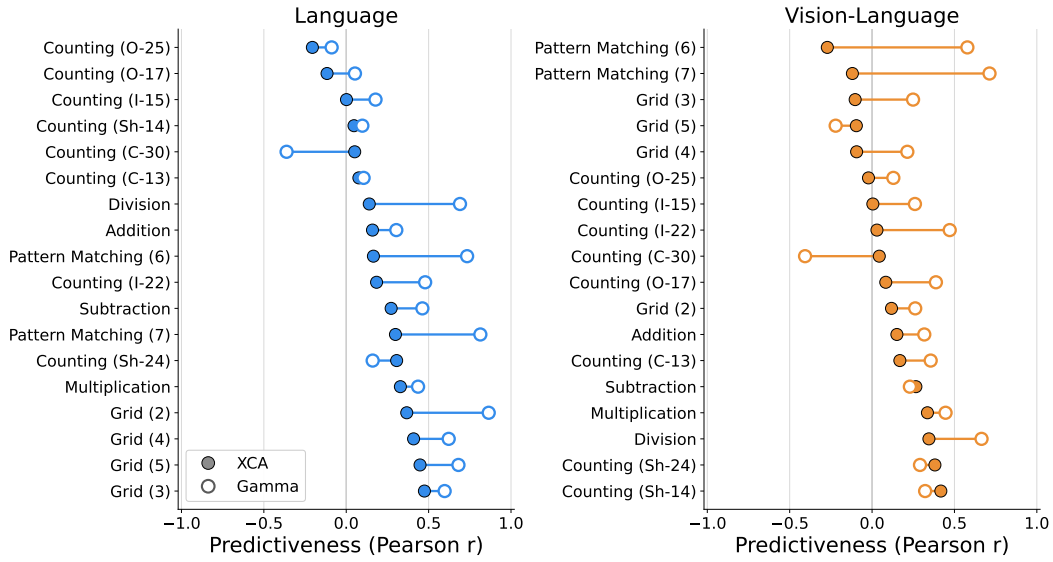


Figure 21: Qwen-2.5-VL-3B. Task relatedness compared to XCA.

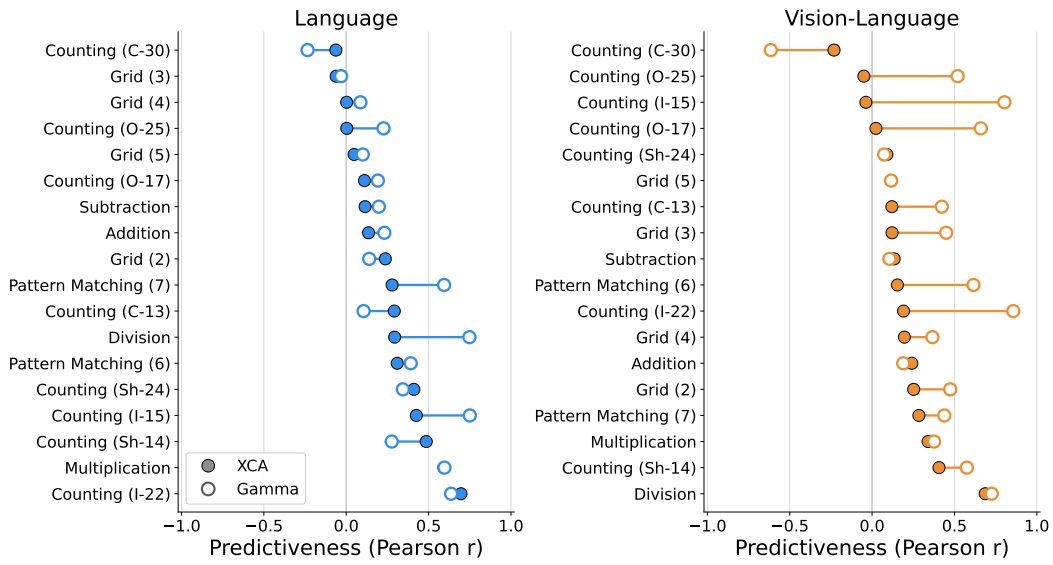


Figure 22: Qwen-2.5-VL-7B. Task relatedness compared to XCA.