

In-Context Compilation: Encoding Context-Induced Behavioral Shift into Model Weights

Michael Merzin¹ Yehonatan Elisha² Oren Barkan¹
Yonatan Toib² Noam Koenigstein²

¹The Open University, Israel

²Tel Aviv University, Israel

Abstract

In-context learning (ICL) adapts LLM behavior by placing task-relevant context in the prompt. While powerful, this mechanism can be inefficient: the context must be reprocessed for every query, and when it is substantially longer than the query itself, it dominates inference cost in both latency and token overhead. This paper contends that the *behavioral shift* induced by an in-context signal can be compiled directly into the model weights, enabling subsequent inference without including the original context in the prompt. To this end, we propose In-Context Compilation (ICC), a gradient free method that encodes a context-induced behavioral shift as a sparse closed-form edit to attention output-projection weights. ICC is calibrated solely from the activation differences between no-context and with-context forward passes on a small set of behaviorally informative examples, requiring neither backpropagation nor trainable parameters. We evaluate ICC in the few-shot setting, where labeled contexts provide controlled behavioral targets. Across diverse LLM architectures and classification and reasoning benchmarks, the ICC-adapted LLM closely approximates the performance of the same LLM prompted with the full context, while omitting that context at inference time. In some cases, it even surpasses the with-context model. Moreover, on most benchmarks, ICC is competitive with and often outperforms gradient-based PEFT methods (LoRA, DoRA, IA3, Prompt/Prefix Tuning, LoReFT) and training-free activation-injection baselines (Function Vectors, ICV), while completing compilation in 10–18s with only ~64–120 MB of overhead. Our code is available at <https://github.com/michaelmerzin/in-context-compilation>.

1 Introduction

Adapting a language model to a new task currently forces a choice between two costly extremes: *In-context learning* (ICL) (Brown et al., 2020) re-

quires no training but reprocesses the entire conditioning context (demonstrations, instructions, or retrieved documents, etc.) on every query. For a model serving millions of requests, this is a recurring tax: long context can dominate both latency and memory bandwidth, despite carrying no new information after the first query. The natural alternative, *parameter-efficient fine-tuning* (PEFT), embeds task knowledge directly into weights and pays zero per-query overhead. However, PEFT is gradient-driven: methods such as LoRA (Hu et al., 2021), IA³ (Liu et al., 2022), and prefix tuning (Li and Liang, 2021) all require iterative backpropagation, optimizer state, and enough labeled data to avoid the instability that can arise in low-data regimes (Chen et al., 2022). Neither option is fully satisfactory when labeled data is scarce and inference cost matters.

A third path: compiling in-context model behavior into weights. We observe that when an in-context signal improves model outputs, it does so by inducing a consistent and measurable shift in the model’s internal activations. Prior work has shown that this *behavioral shift* concentrates in a small subspace of late-layer attention heads (Hendel et al., 2023; Todd et al., 2024; Yin and Steinhardt, 2025). If this shift is localized and consistent, it should be possible to encode it analytically into the weights themselves, allowing the model to behave *as if* the context were present without including it in the prompt at inference time.

Unlike activation-vector methods, which inject the same static task direction at inference, task arithmetic, which reuses parameter deltas obtained through fine-tuning, and factual model editors, which rewrite specific associations, ICC starts from the same model with and without the context, measures the behavioral shift induced by the prompt, and solves directly for a sparse, persistent modification of the model weights that acts as a dynamic intervention, without gradient training or runtime

injection.

We formalize this intuition as **In-Context Compilation (ICC)**, a gradient-free method that performs a one-time offline compilation step. Given a small calibration pool of examples for which the in-context signal demonstrably improves or significantly alters the predictions, ICC: (i) records last-token activations under zero-shot and few-shot conditions; (ii) identifies, via a sensitivity score, a sparse set of attention output-projection weights where the behavioral delta concentrates; and (iii) solves a closed-form regularized least-squares problem to obtain a weight update that approximates few-shot activation targets from zero-shot inputs. In its supervised form, ICC requires labels only for the calibration pool, while its unsupervised variant requires no labeled data at all. After compilation, the edited model runs at standard zero-shot cost: the context is removed entirely, with no cached state and no per-query overhead.

Selective compilation can improve accuracy. ICC reduces noise in two ways. First, supervised calibration uses only examples in G where context corrects the no-context prediction. Second, ICC edits only the most context-sensitive weight rows, leaving weakly affected weights unchanged. Together, these forms of selectivity can isolate the useful part of the in-context behavioral shift. Across 27 task-model pairs, ICC exceeds the few-shot teacher on 21, including a +17.08pp gain on Phi-3-mini with DBPedia-14.

Contributions. We make the following contributions and claims:

1. **In-context Behavior Compilation.** ICC facilitates a gradient-free closed-form weight edit targeting sparse layer attention rows that persistently embeds a context-induced behavioral shift into model weights, enabling inference without the context being present thereafter. We formulate ICC in general terms and instantiate it on the few-shot ICL setting.
2. **Label-Free and Sweep-Free Calibration.** An unsupervised KL-divergence ranking that replaces the need for labeled corrections, and a cross-fit auto-sparsity selector that determines the optimal edit scope directly from activation deltas - eliminating oracle evaluation sweeps entirely.
3. **Teacher Approximation and Noise Isolation.** Across Mistral-7B-Instruct-v0.2, Phi-3-

mini, and Gemma-2-2B on classification and reasoning benchmarks, ICC reliably matches few-shot teacher accuracy at zero-shot cost - and occasionally surpasses it by filtering context-token noise from inference.

2 Related Work

2.1 Gradient-based parameter-efficient adaptation

One approach to compiling task behavior into weights is parameter-efficient fine-tuning. LoRA (Hu et al., 2021) and its quantized variant QLoRA (Dettmers et al., 2023) inject low-rank updates into attention projections. IA³ (Liu et al., 2022) learns multiplicative scaling vectors. Adapters (Houlsby et al., 2019; He et al., 2021) insert bottleneck MLPs between layers. Prompt and prefix tuning (Lester et al., 2021; Li and Liang, 2021) train continuous embeddings prepended to the input. All of these are gradient-driven: they require iterative backpropagation, optimizer state, and in practice, enough labeled data to escape the noisy-loss regime. Chen et al. (2022) document this instability empirically, showing that PEFT methods produce unreliable validation signals in low-data regimes and frequently underperform full finetuning when training data is scarce. ICC targets precisely this regime: rather than improving gradient stability with better initialization or regularization, it eliminates gradients entirely, replacing the optimization loop with a single closed-form ridge solve over activation deltas.

2.2 Activation steering and function vectors

Function vectors (Todd et al., 2024), in-context vectors (Liu et al., 2023), and task vectors (Hendel et al., 2023) extract task-level directions from few-shot activations and inject them during zero-shot decoding. Activation steering (Turner et al., 2023; Rinsky et al., 2024) and representation engineering (Zou et al., 2023) extend this to behavioral attributes such as sycophancy or refusal. These methods share two properties: they leave model weights unchanged, and they apply the *same* additive shift to every input regardless of whether that input resembles the calibration distribution. Our function-vector baseline (Section 4.3) makes this concrete: it injects the fixed mean delta $\bar{\Delta}_\ell = \frac{1}{|G|} \sum_{i \in G} (z_{\ell,i}^w - z_{\ell,i}^n)$, which recovers only part of the few-shot gain. ICC instead encodes the behavioral delta into the attention output-projection matrices themselves, yielding a persistent dynamic

intervention: the resulting shift is a linear function of the hidden state of each new input, rather than the same fixed direction being applied to every example.

Task arithmetic. Task arithmetic (Ilharco et al., 2022) defines a task vector as the parameter difference between a fine-tuned model and its pretrained initialization. The task is therefore first learned through gradient-based fine-tuning, after which the resulting parameter delta can be reused or combined. ICC instead requires no fine-tuned model or training-derived parameter delta: it measures the behavioral shift induced by context and directly solves for a sparse persistent edit of the model weights.

2.3 Closed-form weight editing

ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) edit factual associations into MLP layers via rank-one or batched least-squares updates; PMET (Li et al., 2024) similarly targets FFN weights. ICC shares the high-level idea of an analytic weight edit but differs fundamentally in purpose and mechanism. ROME rewrites isolated facts, requires a causal-tracing sweep plus an inner optimization for the target activation, and enforces an equality constraint on a single MLP layer. ICC compiles a *behavioral distribution* - the task-conditional output shift induced by an in-context signal - reads its targets directly from forward passes with no causal interventions, and applies a Tikhonov-regularized sparse update across attention output-projection rows. To the best of our knowledge, ICC is the first to introduce a closed-form gradient-free weight edit targeting attention projections to compile context-induced behavior.

2.4 Prompt compression and KV caching

Gist tokens (Mu et al., 2023), AutoCompressors (Chevalier et al., 2023), and LLMLingua (Jiang et al., 2023b; Pan et al., 2024) reduce the context’s footprint; prompt caching and KV-cache reuse (Gim et al., 2024; Liu et al., 2024b) skip recomputation by storing intermediate attention states. These methods meaningfully reduce overhead but keep the context alive in some form - as soft tokens, a compressed representation, or cached key-value pairs-meaning per-query cost scales with context length and cached state must be maintained across requests. ICC enables context removal entirely, after compilation, with no cached state and

no per-query overhead beyond a standard context-free forward pass.

2.5 ICL as implicit optimization

A theoretical thread argues that attention implements implicit gradient descent during in-context learning (Von Oswald et al., 2023; Akyürek et al., 2022; Dai et al., 2023; Garg et al., 2022), providing a mechanistic account of why a context induces a coherent behavioral shift. ICC is agnostic to this explanation: it does not analyze *why* ICL works but extracts and encodes *what it produces* - the activation delta between no-context and with-context forward passes-directly into the weights.

3 Method

Let $\mathcal{D} = (x_i, y_i)_{i=1}^N$ be a task dataset, where x_i is a query and y_i is the ground-truth label. For each input x_i , we obtain two predictions from the same base model: $\hat{y}_i^n$, the no-context prediction (predicted class) obtained when the context is omitted from the prompt, and $\hat{y}_i^w$, the with-context prediction obtained when the context is combined with the query as the model input. In general, the context may contain any source of task-relevant information, including instructions, demonstrations (few-shot examples), retrieved knowledge, or any other signal intended to guide the model toward the desired behavior.

ICC can operate in either supervised mode, where the labels y_i are used, or unsupervised mode, where labels are not required.

In supervised mode, ICC first constructs a correction set G , consisting of examples from $\mathcal{D}$ for which the model is correct with context but incorrect without context:

$$G = \{i \mid \hat{y}_i^w = y_i \wedge \hat{y}_i^n \neq y_i\}. \quad (1)$$

Thus, G contains examples on which the in-context signal changes the model prediction from incorrect to correct. These examples provide a measurable behavioral signal from which ICC calibrates its weight edit.

We partition the dataset $\mathcal{D}$ into three disjoint subsets: a *demonstration pool*, a *calibration pool*, and an *evaluation set*. The demonstration pool provides the in-context examples used to construct the conditioning context, such as few-shot demonstrations. The calibration pool is used to extract the correction set G , and the evaluation set is held out for final accuracy measurement.

Unsupervised Construction of G When labels are unavailable, G cannot be constructed using Eq. 1. Instead, we compute for each example i the predicted class¹ distribution obtained with context P_i^w and without context P_i^n . Then, we compute the KL-divergence between these distributions, $D_{\text{KL}}(P_i^w|P_i^n)$. We then run one-dimensional K -means with $K = 2$ on the D_{KL} scores and set G to include calibration examples whose D_{KL} value exceeds the mean of the two cluster centroids.

This simple construction is robust in practice and forms an effective correction set G when labels are unavailable, without requiring hyperparameter tuning. Our experiments in Sec. 4.5 show that this label-free alternative for constructing G achieves results competitive with the supervised construction of G .

This label-free criterion assumes that the with-context prompt has already been validated to improve performance on the target task. KL divergence serves as a distribution-shift signal for identifying examples on which the with-context prompt most strongly changes the model, but does not by itself determine whether the induced behavior is correct or desirable.

3.1 Weight Selection via Sensitivity Scores

ICC edits a sparse subset of model weights corresponding to rows of the attention output-projection matrices (o_{proj}) across the model layers. To select these rows, we compute a per-row sensitivity score that measures how strongly the scalar output produced by applying each row (acting as a weight filter), to the **last**² token hidden representation changes in response to the presence of the in-context signal relative to its absence.

Denote by $o_{\text{proj}}^\ell \in \mathbb{R}^{d \times d}$ the attention output-projection matrix in layer ℓ , where d is the hidden representation dimension. For each layer-row pair (ℓ, j) , we record the scalar output activation obtained by applying row j of o_{proj}^ℓ to the hidden representation of the last token under two conditions: with context and without context. We denote these activations by $z_{\ell,i}^w[j]$ and $z_{\ell,i}^n[j]$, respectively. The sensitivity score is then defined as the mean absolute activation difference across the examples

in G :

$$s_{\ell,j} = \frac{1}{|G|} \sum_{i \in G} |z_{\ell,i}^w[j] - z_{\ell,i}^n[j]|. \quad (2)$$

A high value of $s_{\ell,j}$ indicates that the scalar output associated with row (ℓ, j) changes substantially when the in-context signal is added, making this row a candidate carrier of the behavioral shift that ICC aims to compile. Restricting the edit to high-sensitivity rows serves two purposes. First, it concentrates the update on components most affected by the in-context signal while leaving weakly affected weights unchanged, thereby limiting unnecessary interference with the base model. Second, sparsity reduces the number of parameters involved in the closed-form edit, which is particularly useful when the calibration set is small. This selectivity may also provide one possible explanation for cases in which the compiled model surpasses the with-context teacher.

We pool all (ℓ, j) pairs across the model and select the global top- k rows with the largest $s_{\ell,j}$ values. The edit budget is set to $k = p \times L \times d$, where L is the number of layers, and $p \in (0, 1)$ is a hyperparameter controlling the sparsity fraction. Selection is *global across layers*: ICC does not impose a fixed per-layer quota, and the empirical sensitivity ranking determines which layers receive more edited rows. The sparsity level p can be swept on an evaluation set or chosen automatically using the cross-fit selector described in Section 3.3.

3.2 Closed-Form Weight Edit

Given the set S of index pairs (ℓ, j) corresponding to the rows selected by the procedure in Sec. 3.1, our goal is to edit each selected row so that, when applied to no-context inputs, it approximates the corresponding activations induced by the in-context signal across all examples in G .

Denote by $W_S^o \in \mathbb{R}^{|S| \times d}$ the matrix obtained by stacking all selected weight rows from the relevant o_{proj} matrices in the model. Let $H^n \in \mathbb{R}^{|G| \times d}$ denote the matrix of last-token hidden representations for the no-context inputs in the calibration set G , and let $H^w \in \mathbb{R}^{|G| \times d}$ denote the corresponding matrix for the with-context inputs. These representations are taken at the layers containing the selected rows³.

¹The classes correspond to a closed set of tokens symbolizing the task specific classes.

²This design choice is made for alignment purposes and is discussed in Appendix A.5

³Since the hidden dimension d is the same across layers, we can stack hidden representations from different layers into a single matrix. The same applies to the selected rows from the o_{proj} matrices associated with different layers.

The goal is then to find an edited weight matrix W_S^* that, when applied to the no-context representations H^n , reproduces the with-context activations $H^w(W_S^o)^\top$ on the calibration examples while remaining close to the original weights W_S^o . This objective can be formulated as a regularized least-squares problem as follows:

$$W_S^* = \arg \min_W \underbrace{\|H^n W^\top - H^w(W_S^o)^\top\|_F^2}_{\text{reproduce with-context output}} + \underbrace{\mu \|W - W_S^o\|_F^2}_{\text{stay close to original weights}} \quad (3)$$

The first term encourages the edited rows to reproduce the target with-context activations from no-context inputs. The second term is a Tikhonov regularizer that anchors the solution to the original weights, preventing the edit from drifting arbitrarily far from W_S^o when the calibration set is small. The hyperparameter $\mu > 0$ controls the strength of this anchor (in all experiments, we set $\mu = 0.01$).

3.3 No-Sweep Auto-Sparsity Variant

The swept variant picks the sparsity p by trying many values on the evaluation set - an oracle that requires labeled held-out data. We remove this oracle with a cross-fit selector that chooses p^* from the activations on G alone.

Idea. A row is worth editing only if its solved correction does two things: it **generalizes** to examples in G it was not fit on, and it **carries a meaningful shift** between the no-context and with-context passes. We measure both directly and score every row by their product.

Setup. For each example $i \in G$ and each row (ℓ, j) , the *activation shift* is

$$\delta_{\ell,j}^{(i)} = z_{\ell,i}^w[j] - z_{\ell,i}^n[j], \quad (4)$$

the change in the row’s scalar output at the last token between the with-context and no-context forward passes. We write $\delta_{\ell,j}$ for the collection of shifts across G .

Cross-fit procedure. Split G into two folds, A and B . On fold A , solve the ridge problem (Eq. 3) to obtain, for each row, a predicted shift $\hat{\delta}_{\ell,j}^{(i)}$. Then test how well this prediction matches the true shifts $\delta_{\ell,j}^{(i)}$ on fold B via the held-out coefficient of determination:

$$R_{\ell,j}^2 = 1 - \frac{\sum_{i \in B} (\delta_{\ell,j}^{(i)} - \hat{\delta}_{\ell,j}^{(i)})^2}{\sum_{i \in B} (\delta_{\ell,j}^{(i)} - \bar{\delta}_{\ell,j})^2}, \quad (5)$$

$$\bar{\delta}_{\ell,j} = \frac{1}{|B|} \sum_{i \in B} \delta_{\ell,j}^{(i)}. \quad (6)$$

$R_{\ell,j}^2$ asks whether the prediction tracks the held-out shifts better than just guessing their mean $\bar{\delta}_{\ell,j}$: $R_{\ell,j}^2 > 0$ means the correction generalizes across folds, while $R_{\ell,j}^2 \leq 0$ means the row carries only example-specific noise.

Row score. We score each row by

$$\text{score}_{\ell,j} = \underbrace{\max(0, R_{\ell,j}^2)}_{\text{transfers?}} \cdot \underbrace{\text{Var}(\delta_{\ell,j})}_{\text{moves enough?}}, \quad (7)$$

where $\text{Var}(\delta_{\ell,j}) = \frac{1}{|B|} \sum_{i \in B} (\delta_{\ell,j}^{(i)} - \bar{\delta}_{\ell,j})^2$ is the sample variance of the held-out shifts. The first factor keeps only rows whose corrections transfer; the second weights them by how much the row actually moves under the in-context signal.

Selecting p^* . Rank all rows by score and keep the shortest prefix whose scores sum to 90% of the total. The size of this prefix defines p^* automatically, with no evaluation-set access.

3.4 Computational Complexity

The offline compilation of ICC integrates naturally into standard prompt validation workflows. When deploying an in-context prompt, practitioners routinely execute both zero-shot and with-context inference over a small calibration pool to verify task performance. ICC reuses these exact forward passes; rather than discarding the activations after evaluation, it caches the last-token hidden representations to compute row-wise sensitivity scores and identify the target edit scope. Consequently, the marginal cost over the standard prompt-selection workflow is limited strictly to the closed-form ridge solve. Unlike other methods, this process completely sidesteps backpropagation, gradient tracking, and iterative optimization loops. Runtime and memory usage comparisons are reported in Sec. 4.3.

4 Experiments

All experiments and runtime and memory measurements were conducted on a NVIDIA DGX machine equipped with 4XA100-80GB GPUs.

4.1 Experimental Setup

In our experiments, the with-context condition (w) is instantiated as few-shot prompting with k labeled demonstrations, and the no-context condition

Model	Task	No-context	With-context	Best ICC	ΔW	Best p
Mistral-7B	DBPedia-14	78.44	84.10	89.07	+4.97	0.25
	SST-2	79.01	86.47	88.65	+2.18	0.10
	MRPC	75.98	77.94	80.88	+2.94	0.10
	BoolQ	84.86	84.29	84.40	+0.11	0.05
	ARC-Challenge	76.20	75.60	76.40	+0.80	0.10
	GSM8K	36.00	34.00	42.00	+8.00	0.30
	MATH-500	4.67	6.44	6.89	+0.45	0.25
	MMLU	60.60	59.60	61.20	+1.60	0.20
	AG News	79.01	90.68	84.94	-5.74	0.10
Phi-3-mini	DBPedia-14	77.75	79.82	96.90	+17.08	0.30
	MRPC	60.29	74.75	76.72	+1.97	0.05
	BoolQ	80.50	85.44	86.12	+0.68	0.15
	ARC-Challenge	87.20	88.00	89.20	+1.20	0.30
	GSM8K	37.33	48.00	50.67	+2.67	0.15
	MATH-500	28.00	22.00	27.11	+5.11	0.35
	SST-2	69.27	94.04	92.32	-1.72	0.15
	AG News	69.61	71.90	75.46	+3.56	0.15
	MMLU	67.40	65.60	69.20	+3.60	0.25
Gemma-2-2B	DBPedia-14	25.57	91.86	95.07	+3.21	0.20
	MRPC	43.87	69.85	76.47	+6.62	0.20
	AG News	4.36	80.50	80.73	+0.23	0.05
	ARC-Challenge	71.20	75.00	72.20	-2.80	0.20
	BoolQ	69.27	83.72	80.85	-2.87	0.35
	MMLU	56.00	56.00	57.60	+1.60	0.20
	SST-2	71.33	95.76	91.06	-4.70	0.10
	GSM8K	61.33	62.67	62.00	-0.67	0.05
	MATH-500	11.33	9.78	14.67	+4.89	0.10

Table 1: Per-task accuracy of ICC against no-context and with-context baselines across three architectures. ΔW is the gap to the with-context teacher; p is the swept sparsity. Bold marks the best score per row.

(n) as zero-shot prompting. We evaluate on three instruction-tuned decoder-only models: **Mistral-7B-Instruct-v0.2** (Jiang et al., 2023a) (32 layers, $d=4096$), **Phi-3-mini-4k-instruct** (Abdin et al., 2024) (32 layers, $d=3072$), and **Gemma-2-2B-it** (Team et al., 2024) (26 layers, $d=2048$).

Tasks. We cover three task families: *classification* (SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), AG News (Zhang et al., 2015), DBPedia-14 (Zhang et al., 2015), BoolQ (Clark et al., 2019)), *multiple-choice* (ARC-Challenge (Clark et al., 2018); MMLU (Hendrycks et al., 2020) evaluated under the Hendrycks 5-shot subject-matched protocol, $n=500$), and *multi-token mathematical generation* (GSM8K (Cobbe et al., 2021) and MATH-500 (Lightman et al., 2024)).

Baselines. We compare against (i) the no-context baseline (zero-shot prompting); (ii) the with-context teacher (few-shot prompting), and (iii) gradient-based PEFT methods: LoRA (Hu et al., 2021) (rank = 8), DoRA (Liu et al., 2024a) (rank = 8), LoReFT (Wu et al., 2024) (rank = 4), IA³ (Liu et al., 2022), Prompt Tuning (Lester et al., 2021), and Prefix Tuning (Li and Liang, 2021). We

follow (Wu et al., 2024) and train them for 3 epochs on the same correction set; (iv) the function-vector baseline (Todd et al., 2024), which injects the mean no-context/with-context activation delta $\bar{\Delta}_\ell$ at inference; and (v) in-context vectors (ICV) (Liu et al., 2023) swept over λ . All methods use identical low-data calibration budgets.

Data splits. We enforce a strict three-way partition: a *demonstration pool* from which the in-context examples for the with-context condition (i.e., few-shot demonstrations) are sampled, a disjoint *calibration pool* used to construct the correction set G and fit the ridge solve, and a held-out *evaluation set* that is never seen during compilation. No calibration example appears in evaluation. Per-task split sizes are reported in Tab. 31 in Appendix D.

Hyperparameters. We sweep $p \in \{0.05, 0.10, \dots, 0.35\}$ unless the auto-sparsity variant (Sec. 3.3) is active. The ridge solve (Eq. 3) uses $\mu = 0.01$.

The underlying $\mathcal{D}_{\text{demo}}$, $\mathcal{D}_{\text{cal}}$, and $\mathcal{D}_{\text{eval}}$ splits are fixed across models and methods, whereas G , constructed from the calibration pool, is model-dependent and is therefore formed separately for

each model and demonstration seed. Within each comparison, ICC and all PEFT baselines use the same model-specific G .

4.2 Does ICC Successfully Compile Context-Induced Behavioral Shift?

Table 1 reports the with-context teacher, the no-context baseline, and the best-swept ICC accuracy across all three architectures and nine tasks each. **ICC exceeds the with-context teacher on 21 of 27 task-model pairs**, establishing that the behavioral shift induced by in-context prompting is analytically compilable into a sparse set of attention output-projection weights.

4.3 Performance and Efficiency Evaluation

We benchmark ICC against gradient-based PEFT (LoRA, DoRA, LoReFT, IA³, Prompt/Prefix Tuning) and activation-injection baselines (Function Vectors, ICV) on Mistral-7B under identical low-data calibration (Table 2). Several gradient-based baselines exhibit severe low-data failures, whereas ICC remains competitive across all four and achieves the best performance on MRPC and MMLU. In low-data regimes, gradient methods drop to near-random accuracy while ICC’s closed-form solve sidesteps the instability and still gains over the with-context teacher. It consistently outperforms Function Vectors and generally outperforms ICV, while the weight edit adapts to each input rather than applying a fixed shift. Efficiency is structural: ICC compiles in seconds with tens of MB of overhead - substantially faster and lower-overhead than the gradient-based baselines in our experiments.

PEFT with the full calibration pool. Because Table 2 evaluates the low-data regime targeted by ICC, we additionally train LoRA and DoRA on the full $\mathcal{D}_{\text{cal}}$. With independent shuffling each epoch, PEFT substantially improves and matches or exceeds ICC on several tasks, confirming that ICC’s main advantage is in the low-data regime rather than universal accuracy dominance. Full results are reported in Appendix A.3.

Compiling output-format behavior. The low no-context accuracy of Gemma-2-2B on AG News (4.36%) is primarily an output-format failure rather than below-random semantic classification. Evaluation uses free generation followed by a strict parser that accepts exactly one of the four dataset labels (World, Sports, Business, or Sci/Tech).

Without demonstrations, the model frequently produces semantically plausible but invalid labels such as “Environment”, “Weather”, “Labor”, or “Pensions”, yielding a parse rate of only 5.73%. Among parseable responses, approximately 76% are correct.

Few-shot demonstrations teach the required label mapping, increasing both parse rate and accuracy to 100% and 80.50%, respectively. After the demonstrations are removed, ICC preserves this induced behavior and reaches 80.73% accuracy. As a diagnostic, restricting prediction to the four valid labels raises Gemma-2-2B from 4.36% to 62.84% without context, while the with-context score remains 80.50%. This confirms that the low free-generation score primarily reflects failure to follow the task-specific verbalizer.

Multi-token generation. Because the edited weights remain active at every decoding step, ICC can affect the full autoregressive trajectory despite being calibrated from last-token activations. The results on GSM8K and MATH-500 show that the compiled behavior persists beyond the first generated token and remains effective throughout multi-token generation.

Post-edit language-model quality. We additionally measure WikiText-2 perplexity to test whether the weight edits degrade general language modeling. The unedited Mistral-7B model obtains a perplexity of 5.431. After compiling SST-2, MRPC, DBpedia-14, and MMLU behaviors, perplexities are 5.424, 5.420, 5.402, and 5.415, respectively. We therefore observe no degradation on this open-domain language-modeling measure.

Editing target. Under the same ICC protocol on Mistral-7B, editing o_{proj} outperforms editing mlp.down_proj on 6 of 7 tasks, with an average gain of +1.92 percentage points. This supports attention output projections as the primary editing target. Full per-task results are reported in Appendix A.4.

4.4 ICC without Oracle Sparsity Selection

The swept variant picks p by validation accuracy - an oracle that requires labeled held-out data. Our cross-fit selector (Sec. 3.3) removes this requirement: p^* is chosen directly from activation statistics on the correction set G , with no evaluation-set access.

Table 3 shows the auto-calibrated edit recovers a

Method	SST-2	MRPC	DBPedia	MMLU	Time	Extra Mem. (MB)
No-context	79.01	75.98	78.44	60.60	-	0
With-context	86.47	77.94	84.10	59.60	-	0
Function Vec.	81.77	78.92	78.56	60.60	avg.	64-120
ICV sweep performance range	92.66	79.66	82.22	59.60	λ sweep	64-120
	33.72-92.66	0.00-79.66	77.98-82.22	54.40-59.60		
DoRA	93.23	31.62	94.15	55.00	1m30s-3m19s	722-4178
LoReFT	93.35	31.62	91.17	40.00	5m12s-7m28s	458-549
LoRA	94.72	34.31	41.17	55.40	58-109s	680-879
IA ³	92.32	70.83	87.16	60.40	57-96s	634-832
Prompt	77.29	74.26	84.52	50.20	59-100s	548-764
Prefix	4.82	0.00	5.05	21.80	52-95s	407-2606
ICC	88.65	80.88	89.07	61.20	10.5-18.4s	64-120

Table 2: Accuracy, adaptation time, and extra memory of ICC versus PEFT, ReFT, and activation-injection baselines on Mistral-7B. Memory is reported above the loaded base model, not total GPU residency. All methods use the same correction set G . For ICV, the bold number is the best accuracy across a sweep over the injection-strength hyperparameter λ and the small range below it is the min-max across that sweep, illustrating the method’s hyperparameter sensitivity. Bold marks the best score per column.

substantial fraction of the no-context - with-context gap.

Model	Task	No-context	With-context	Auto-Edit
Mistral	DBPedia	78.44	84.10	84.17
Mistral	SST-2	79.01	86.47	85.09
Mistral	BoolQ	84.86	84.29	84.52
Phi-3	DBPedia	77.75	79.82	93.00
Phi-3	MRPC	60.29	74.75	79.90
Phi-3	BoolQ	80.50	85.44	82.91

Table 3: No-sweep auto-sparsity results. Auto-Edit is ICC run with the cross-fit selector (§3.3), which picks p^* from activations on G alone. The no-context and with-context columns are the baselines from Table 1. Bold marks rows where Auto-Edit exceeds the with-context teacher.

4.5 Label-Free ICC via Unsupervised KL-based Calibration

Table 4 reports the label-free KL-ranking variant, which frequently surpasses both the with-context teacher and the supervised edit. Table 5 isolates why: the +1.15pp jump from Supervised to Hybrid shows that KL-magnitude, not label correctness, drives the gain - high KL signals clean activation shifts and stronger least-squares updates.

Reliability under small correction sets. To assess the stability of KL-ICC in the smallest correction-set regimes, we repeat the Mistral experiments over three seeds. For SST-2 ($|G| = 3$), MRPC ($|G| = 4$), and DBPedia-14 ($|G| = 5$), KL-ICC obtains 84.19 ± 0.72 , 79.41 ± 1.12 , and 88.74 ± 4.04 , respectively.

5 Analysis and Interpretability

5.1 Layer Localization - Where Does the Edit Live?

Table 6 reports the distribution of selected rows across transformer layers for Mistral-7B at the auto-selected p^* . On Mistral, the final transformer blocks dominate the edit: layer 31 alone captures 15.6-28.6% of selected rows across tasks, layers 28-31 capture ~ 45 -60%, and layers 26-31 capture 55-72%. Phi-3-mini exhibits the same late-layer pattern: the top-sensitivity neuron for five of six tasks lies in layer 28 or later, with most concentrated in layer 31 (Table 7).

This concentration is a recovered property of the model. The sensitivity score $s_{\ell,j}$ (Equation 2) is computed uniformly across all layers and channels; late layers emerge as dominant purely from the magnitudes of the activation deltas. This finding is consistent with prior mechanistic work showing that in-context behavioral shifts concentrate in late attention heads (Hendel et al., 2023; Todd et al., 2024; Yin and Steinhardt, 2025), and provides empirical grounding for ICC’s claim to compile a localized, consistent behavioral delta.

5.2 The Noise-Filtering Property

In supervised ICC, G contains only examples where the with-context prompt corrects the model (Eq. 1), emphasizing useful context-induced shifts. Removing demonstrations at inference also removes formatting and ordering effects, while restricting the edit to high-sensitivity rows may fur-

Model	Task	$ G $	Prec.	No-context	With-context	Unsup edit
Mistral-7B	SST-2	3	100%	79.01	86.47	91.17
	MRPC	4	75%	75.98	77.94	80.64
	DBPedia-14	5	60%	78.44	84.10	90.02
	AG News	15	80%	79.01	90.68	80.85
	BoolQ	8	75%	84.86	84.29 (deg.)	84.52
Phi-3-mini	SST-2	8	50%	69.27	94.04	93.23
	MRPC	16	81%	60.29	74.75	76.72
	DBPedia-14	32	75%	77.75	79.82	98.17
	BoolQ	6	67%	80.50	85.44	85.78

Table 4: Unsupervised KL-based calibration across Mistral-7B and Phi-3-mini. $|G|$ denotes the number of selected calibration examples. Precision is measured post hoc and is not used during selection. Bold indicates the unsupervised edit exceeds the with-context teacher.

Method	$ G $	Prec.	Accuracy
Supervised (unranked)	64	100%	96.90%
Hybrid (Sup + KL)	32	100%	98.05%
Unsupervised (top-KL)	32	75%	98.17%

Table 5: Hybrid ablation on Phi-3 DBPedia-14.

Task	Top row	L31	L28–31	L26–31
AG News	L31-N3901	23.5%	~55%	~72%
DBPedia-14	L31-N2070	~22%	~53%	~70%
SST-2	L31-N3901	~20%	~50%	~65%
MRPC	L31-N2070	28.6%	~45%	~55%
GSM8K	L30-N3901	~20%	~51%	~71%
MMLU	L31-N2070	15.6%	59.9%	71.0%

Table 6: Layer localization at auto-selected p^* (Mistral-7B). Across these six tasks, layers 26–31 capture 55–72% of selected rows.

ther reduce interference by leaving weakly affected weights unchanged. Together, these effects may help explain why ICC can sometimes exceed its with-context teacher, as on Phi-3 DBPedia-14 (Table 1), where the compiled edit recovers performance that the prompt fails to elicit reliably.

6 Conclusion

We introduced In-Context Compilation (ICC), a gradient-free closed-form edit that compiles the behavioral shift of an in-context signal directly into a sparse set of attention output-projection weights. Across three architectures and nine tasks per architecture, ICC reliably approximates the with-context teacher at no-context inference time, without backpropagation, optimizer state, or trainable parameters.

Our central finding is that the behavioral shift induced by in-context learning can be analytically compiled into the model weights. Backed by a label-free unsupervised pipeline and a sweep-free

Task	Phi-3 top neuron	Sensitivity score	Late ($\geq L28$)?
SST-2	L12-N2046	6.42	No
DBPedia-14	L31-N525	16.34	Yes
AG News	L31-N525	14.80	Yes
MRPC	L31-N525	14.80	Yes
GSM8K	L29-N2812	15.36	Yes
ARC-Challenge	L29-N2024	11.53	Yes

Table 7: Top sensitivity neurons on Phi-3-mini.

auto-sparsity mechanism, ICC provides a gradient-free closed-form alternative to PEFT that avoids optimizer state overhead. It performs best in low-data regimes where gradient-based methods exhibit instability, and does not aim to replace LoRA when that method has sufficient data and stable convergence.

Limitations

Dependence on Context-Induced Signal. ICC compiles behavior that is already induced by the conditioning context; it does not create capabilities that the underlying model cannot express. Its effectiveness therefore depends on the context producing a sufficiently consistent and measurable behavioral shift. When the with-context behavior provides only a weak or highly variable signal, the resulting edit may be less effective and remain closer to the no-context baseline.

Alternative Weight Edits. Our targeted ablation compares attention output-projection edits with MLP down-projection edits on Mistral-7B, with o_{proj} performing better on 6 of 7 tasks. However, this comparison is not exhaustive. We do not systematically evaluate Q/K/V projections, combinations of multiple module types, or whether the same editing preference holds across architectures. Exploring these alternatives may further improve the effectiveness and generality of ICC.

Generative Scope. Beyond classification and multiple-choice benchmarks, we evaluate ICC on GSM8K and MATH-500, showing that the compiled behavior can persist through multi-token autoregressive generation. We have not yet evaluated fully open-ended long-form or knowledge-intensive generation, which we leave for future work.

Broader Impact

While ICC efficiently embeds task-specific behaviors into model weights, this mechanism is inherently agnostic to the nature of the behavior being compiled. Consequently, there is a theoretical risk that undesirable or biased behaviors could be inadvertently or intentionally encoded into the model. Because the compiled model operates without the original context at inference time, standard prompt-based safety filters might not detect these embedded traits. We encourage future work to explore auditing tools to verify the safety and integrity of the compiled weights prior to deployment.

Acknowledgments

This work was supported by the Ministry of Innovation, Science & Technology, Israel, and by the Israeli Science Foundation (ISF grant 977/26).

References

- Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. [Phi-3 technical report: A highly capable language model locally on your phone](#). *Preprint*, arXiv:2404.14219.
- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2612–2626.
- Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598.
- In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. 2024. Prompt cache: Modular attention reuse for low-latency inference. *Proceedings of Machine Learning and Systems*, 6:325–338.
- Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.
- Roe Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333.

- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023a. *Mistral 7b*. Preprint, arXiv:2310.06825.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. LlmLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 13358–13376.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 3045–3059.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pages 4582–4597.
- Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohata, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in neural information processing systems*, 35:1950–1965.
- Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2023. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*.
- Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, and 1 others. 2024b. Cachegen: Kv cache compression and streaming for fast large language model serving. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 38–56.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.
- Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.
- Jesse Mu, Xiang Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352.
- Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor R  hle, Yuqing Yang, Chin-Yew Lin, and 1 others. 2024. LlmLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981.
- Nina Rimskey, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. *Recursive deep models for semantic compositionality over a sentiment treebank*. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. 2024. Function vectors in large language models. In *International conference on learning representations*, volume 2024, pages 17282–17333.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.
- Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.
- Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962.
- Kayo Yin and Jacob Steinhardt. 2025. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

In-Context Compilation: Encoding Context-Induced Behavioral Shift into Model Weights - Appendix

A Further Experimental Insights

A.1 Random-Fill Ablation

To verify that gains stem from the closed-form solve and not just row selection, we replace W_S^* with $\mathcal{U}[-1, 1]$ noise in the identical selected rows on Mistral DBPedia-14. Table 8 shows accuracy collapses to 0.00% across all sparsity values $p \in \{0.05, \dots, 0.30\}$ and all seeds, confirming that the ridge-anchored optimization injects highly specific geometric adaptations rather than benefiting from row selection alone.

The complete collapse to 0.00% - rather than to a random-guessing baseline near $1/C$ - is mechanistically expected. The natural weight magnitudes in attention output-projections are on the order of 10^{-2} , so $\mathcal{U}[-1, 1]$ noise is roughly two orders of magnitude larger than the values it replaces. Even at the smallest tested sparsity ($p = 0.05$, $\sim 6,500$ rows on Mistral-7B), this saturates the residual stream and prevents coherent token generation, producing degenerate outputs that match no valid class label. The complete collapse is therefore the expected behavior, reproducible across all three fill seeds and all six sparsity values.

Variant	Acc	ΔNC	ΔW
No-context baseline	78.44%	—	−5.66pp
With-context teacher (seeds 1–2)	84.10%	+5.66pp	—
ICC closed-form at $p = 0.25$	89.07%	+10.63pp	+4.97pp
Random $\mathcal{U}[-1, 1]$ fill (all p)	0.00%	−78.44pp	−84.10pp

Table 8: Random-weight ablation on DBPedia-14 (Mistral-7B, seeds 1–2, 3 fill seeds per p). Same rows as the supervised ICC edit, replaced with $\mathcal{U}[-1, 1]$ noise. The random-fill standard deviation is 0.00% at all points.

A.2 With-Context Quality and Latent Capability Recovery

Before interpreting the ΔW column, it is important to note that with-context accuracy is itself an imperfect ceiling. A with-context prompt can underperform due to volatile formatting, suboptimal demonstration ordering, or a demonstration pool that fails to surface the model’s latent capability for a given task. ICC is calibrated exclusively on examples where the with-context prompt demonstrably helps (Eq. 1), which has two consequences.

First, the calibration signal is high-quality by construction: the activation shifts in G are real, con-

sistent, and predictable by the ridge-anchored solve. Second, ICC sometimes exceeds a weak with-context teacher not by transcending the model’s capacity, but by more faithfully recovering a capability the prompt failed to elicit.

Conversely, where the with-context teacher is already near-optimal, ICC predictably falls slightly below, because the residual with-context to no-context shift it must encode is small relative to the signal lost by removing the full demonstration context. Readers should therefore interpret ΔW jointly with absolute with-context accuracy: a large positive ΔW above a low with-context baseline reflects capability recovery; a small negative ΔW below a near-ceiling with-context baseline reflects a reasonable tradeoff.

A related observation is that when the with-context teacher is itself weak, ICC nonetheless converges to teacher-level behavior rather than diverging or regressing to no-context. This is consistent with the method’s design intent: ICC is calibrated to reproduce the with-context teacher’s behavior, and it succeeds at this even when that behavior is mediocre. Reproducing a weak teacher is not a failure mode – it is evidence that the compilation step is doing what it claims to do.

Resource profile. PEFT baselines require iterative backpropagation, trainable adapter or prompt parameters, gradients, optimizer states, and training-time activation graphs. In our runs, LoRA adds 6.82M trainable parameters, Prefix Tuning 1.31M, IA³ 524K, and Prompt Tuning 81.9K, with measured GPU peaks of roughly 14.6–16.5 GB (Prefix Tuning on MMLU requiring the largest overhead because it must run without gradient checkpointing). ICC adds no trainable parameters, gradients, or optimizer state; its extra memory is dominated by cached calibration activations and closed-form solve buffers – approximately 64–120 MB on Mistral-7B for the SST-2/MRPC correction-set sizes in this comparison. ICC should therefore be viewed as a fast, closed-form, gradient-free behavior-compilation method rather than one that dramatically reduces total model residency.

Task	ICC (G)	LoRA ($\mathcal{D}_{\text{cal}}$)	DoRA ($\mathcal{D}_{\text{cal}}$)
SST-2	89.22	91.35 $\pm$ 2.16	94.65 $\pm$ 0.54
MRPC	77.86	79.64 $\pm$ 6.71	82.76 $\pm$ 0.93
DBPedia-14	88.23	92.42 $\pm$ 1.89	88.12 $\pm$ 1.48
MMLU	60.13	59.80 $\pm$ 1.91	57.87 $\pm$ 0.64

Table 9: PEFT sensitivity analysis on Mistral-7B using the full calibration pool. LoRA and DoRA are trained on $\mathcal{D}_{\text{cal}}$ with independent shuffling each epoch; mean $\pm$ standard deviation is reported over three seeds. ICC is shown for reference and uses the smaller correction set G .

A.3 PEFT with Full Calibration Data

To test whether the low-data PEFT failures in Table 2 arise from the small correction sets, we additionally train LoRA and DoRA for three epochs on the full $\mathcal{D}_{\text{cal}}$, independently shuffling the data before each epoch. Table 9 reports mean accuracy and standard deviation over three seeds. The ICC column is shown for reference; ICC continues to use its smaller correction set G , so this experiment gives PEFT access to substantially more training data and is not a matched-data comparison.

A.4 Editing-Target Ablation

To test whether attention output projections are particularly suitable for ICC, we repeat the same Mistral-7B protocol while editing `mlp.down_proj` instead of `o_proj`, using the same sparsity sweep and ridge-regression procedure. Table 10 reports the per-task difference in accuracy. Editing `o_proj` performs better on 6 of 7 tasks, with an average advantage of +1.92 percentage points.

Task	Δ Accuracy ($o_{\text{proj}} - \text{mlp.down_proj}$)
AG News	+5.39
MRPC	+4.65
DBPedia-14	+2.75
BoolQ	+0.42
MMLU	+0.20
SST-2	+0.11
ARC-Challenge	-0.07
Average	+1.92

Table 10: Editing-target ablation on Mistral-7B under the same ICC protocol. Positive values indicate higher accuracy when editing attention output-projection rows rather than MLP down-projection rows.

A.5 Token-Position Alignment

A natural concern when comparing no-context and with-context forward passes is how the two activation sequences – of very different lengths –

are aligned for the ridge solve. We resolve this with a single uniform convention: **both H^n and $H^w(W_S^o)^\top$ are extracted at the final input token of their respective prompts**, i.e., the token immediately preceding the position where the model will emit its first output token. No interior tokens of the with-context prompt enter the calibration tensors; the in-context examples influence the captured activations only through their effect on the hidden state at the last position.

This convention works uniformly across our task types because every prompt template ends with the same suffix in both the no-context and with-context conditions:

- **Single-label classification (SST-2, MRPC, AG News, DBPedia-14).** Both prompts end with a task-specific verbalizer cue: "Sentiment:" for SST-2, "Relationship:" for MRPC, "Topic:" for AG News, and "Category:" for DBPedia-14. The final token is identical between conditions; only the preceding context differs.
- **Multiple-choice (BoolQ, ARC-Challenge, MMLU).** Both prompts end with "Answer:" followed by the choice prefix. The captured activation is the one from which the model produces its letter (or yes/no) answer.
- **Chain-of-thought generation (GSM8K, MATH-500).** For both tasks, the no-context and with-context prompts end with the same task-specific generation cue ("Solution: [/INST]" on Mistral-Instruct, with analogous model-specific serialization where applicable). The captured activation is the one from which the model begins decoding its reasoning chain.

This last-token convention is dimensionally clean ($H^n \in \mathbb{R}^{|G| \times d}$ with one row per calibration example), avoids any need for sequence-level alignment between prompts of different lengths, and respects the autoregressive structure of the model: the activation at the final input token is exactly the state that determines what the model emits next. The ridge solve in Eq. 3 is therefore well-posed regardless of whether the underlying task is classification, multiple-choice, or open-ended generation.

A.6 Prompt Templates

We report the no-context and with-context prompt templates used in our experiments. Braced terms denote example-specific fields. Demonstration and query blocks are separated by one blank line. In demonstration blocks, the answer follows the output cue with a leading space (e.g., Topic: World), whereas the query block terminates at the cue itself (e.g., Topic:). The number of demonstrations used for each task is denoted by k .

AG News ($k = 6$). No-context template.

You are a news topic classifier.
Classify the following news article
into exactly one of these labels:
World, Sports, Business, Sci/Tech.
Answer with only the label name.

Article: {text}
Topic:

With-context template.

You are a news topic classifier.
Classify the following news article
into exactly one of these labels:
World, Sports, Business, Sci/Tech.
Answer with only the label name.

Article: {demo_text}
Topic:
{World|Sports|Business|Sci/Tech}
... ($k = 6$ demonstration blocks)
...
Article: {text}
Topic:

DBPedia-14 ($k = 14$). No-context template.

You are a topic classifier.
Classify each text into exactly
one of these categories: Company,
Educational Institution, Artist,
Athlete, Office Holder, Mean of
Transportation, Building, Natural
Place, Village, Animal, Plant,
Album, Film, Written Work.
Answer with only the category name.

Title: {title}
Text: {content[:100]}
Category:

With-context template.

You are a topic classifier.
Classify each text into exactly
one of these categories: Company,
Educational Institution, Artist,
Athlete, Office Holder, Mean of
Transportation, Building, Natural
Place, Village, Animal, Plant,
Album, Film, Written Work.
Answer with only the category name.

Title: {demo_title}
Text: {demo_content[:100]}
Category: {category}
... ($k = 14$ demonstration blocks)
...
Title: {title}
Text: {content[:100]}
Category:

The Title: line is omitted when the title is empty. Text content is truncated to 100 characters.

SST-2 ($k = 4$). No-context template.

Classify the sentiment of the
following sentence as Positive or
Negative.
Answer with only the sentiment
label.

Sentence: {text}
Sentiment:

With-context template.

Classify the sentiment of the
following sentence as Positive or
Negative.
Answer with only the sentiment
label.

Sentence: {demo_text}
Sentiment: {Positive|Negative}
... ($k = 4$ demonstration blocks)
...
Sentence: {text}
Sentiment:

MRPC ($k = 4$). No-context template.

Determine whether the two sentences
are semantically equivalent.
Answer with only 'Equivalent' or
'Not Equivalent'.

Sentence 1: {sentence1}
Sentence 2: {sentence2}
Relationship:

With-context template.

Determine whether the two sentences
are semantically equivalent.
Answer with only 'Equivalent' or
'Not Equivalent'.

Sentence 1: {demo_sentence1}
Sentence 2: {demo_sentence2}
Relationship: {Equivalent|Not
Equivalent}
... ($k = 4$ demonstration blocks)
...
Sentence 1: {sentence1}
Sentence 2: {sentence2}
Relationship:

BoolQ ($k = 4$). No-context template.

Read the passage and answer the question with exactly No or Yes.

Passage: {passage}
 Question: {question}
 Label:

With-context template.

Read the passage and answer the question with exactly No or Yes.

Passage: {demo_passage}
 Question: {demo_question}
 Label: {No|Yes}
 ... ($k = 4$ demonstration blocks)
 ...
 Passage: {passage}
 Question: {question}
 Label:

ARC-Challenge ($k = 4$). No-context template.

Answer each multiple-choice question with one letter: A, B, C, or D.

Question: {question}
 A. {choice_0}
 B. {choice_1}
 C. {choice_2}
 D. {choice_3}
 Answer:

With-context template.

Answer each multiple-choice question with one letter: A, B, C, or D.

Question: {demo_question}
 A. {demo_choice_0}
 B. {demo_choice_1}
 C. {demo_choice_2}
 D. {demo_choice_3}
 Answer: {A|B|C|D}
 ... ($k = 4$ demonstration blocks)
 ...
 Question: {question}
 A. {choice_0}
 B. {choice_1}
 C. {choice_2}
 D. {choice_3}
 Answer:

MMLU ($k = 5$). No-context template.

The following are multiple choice questions (with answers) about {subject}.

Question: {question}
 A. {choice_0}
 B. {choice_1}
 C. {choice_2}
 D. {choice_3}
 Answer:

With-context template.

The following are multiple choice questions (with answers) about {subject}.

Question: {demo_question}
 A. {demo_choice_0}
 B. {demo_choice_1}
 C. {demo_choice_2}
 D. {demo_choice_3}
 Answer: {A|B|C|D}
 ... ($k = 5$ demonstration blocks)
 ...

Question: {question}
 A. {choice_0}
 B. {choice_1}
 C. {choice_2}
 D. {choice_3}
 Answer:

Here {subject} is the evaluation example's subject with underscores replaced by spaces, defaulting to general. The subject is included in both conditions.

GSM8K ($k = 8$). The following shows the common task-level textual prompt. Model-specific chat serialization tokens are omitted.

No-context template.

Solve the following math word problem step by step. At the end write the answer as: #### <number>
 Question: {question}
 Solution:

With-context template.

Solve math word problems step by step. At the end write the answer as: #### <number>
 Here are solved examples:
 Question: {demo_question}
 Solution: {demo_reasoning}
 #### {demo_answer}
 ... ($k = 8$ demonstration blocks)
 ...
 Now solve this:
 Question: {question}
 Solution:

The eight demonstrations are fixed canonical examples and are identical across evaluation examples and seeds. Unlike the classification and multiple-choice tasks, GSM8K uses slightly different instruction wording in the no-context and with-context conditions.

MATH-500 ($k = 4$). No-context template.

[INST] Solve the math problem step by step. Put the final answer in \boxed{ }.
 Problem: {problem}
 Solution: [/INST]

With-context template.

```
[INST] Solve the math problem step
by step. Put the final answer in
\boxed{.}.

Here are solved examples:

Problem: {demo_problem}
Solution: {demo_solution}
... (k = 4 demonstration blocks)
...

Now solve this problem:
Problem: {problem}
Solution: [/INST]
```

If a demonstration solution does not contain a boxed answer, the formatter appends Final answer: `\boxed{answer}`.

B Detailed Main Results

This section reports per-task results for all three models swept over sparsity p . Table 11 (Mistral-7B), Table 12 (Phi-3-mini), and Table 13 (Gemma-2-2B) each list the no-context baseline, the with-context teacher, ICC’s accuracy at the best p in the oracle sparsity sweep, the gap to the with-context teacher (Δ_W), and the selected p^* .

Task	No-context	With-context	Best edit	Δ_W	Best p
AG News	79.01	90.68	84.94	-5.74	0.10
DBPedia-14	78.44	84.10	89.07	+4.97	0.25
SST-2	79.01	86.47	88.65	+2.18	0.10
MRPC	75.98	77.94	80.88	+2.94	0.10
ARC-Challenge	76.20	75.60	76.40	+0.80	0.10
GSM8K	36.00	34.00	42.00	+8.00	0.30
MMLU	60.60	59.60	61.20	+1.60	0.20
BoolQ	84.86	84.29	84.40	+0.11	0.05
MATH-500	4.67	6.44	6.89	+0.45	0.25

Table 11: Mistral-7B detailed swept results. ICC surpasses the with-context teacher on 8 of 9 tasks.

Task	No-context	With-context	Best edit	Δ_W	Best p
SST-2	69.27	94.04	92.32	-1.72	0.15
DBPedia-14	77.75	79.82	96.90	+17.08	0.30
MRPC	60.29	74.75	76.72	+1.97	0.05
AG News	69.61	71.90	75.46	+3.56	0.15
ARC-Challenge	87.20	88.00	89.20	+1.20	0.30
GSM8K	37.33	48.00	50.67	+2.67	0.15
BoolQ	80.50	85.44	86.12	+0.68	0.15
MMLU	67.40	65.60	69.20	+3.60	0.25
MATH-500	28.00	22.00	27.11	+5.11	0.35

Table 12: Phi-3-mini detailed swept results. The behavioral shift transfers cleanly to a smaller architecture: ICC surpasses the with-context teacher on 8 of 9 tasks.

Task	No-context	With-context	Best edit	Δ_W	Best p
SST-2	71.33	95.76	91.06	-4.70	0.100
MRPC	43.87	69.85	76.47	+6.62	0.200
AG News	4.36	80.50	80.73	+0.23	0.050
DBPedia-14	25.57	91.86	95.07	+3.21	0.200
ARC-Challenge	71.20	75.00	72.20	-2.80	0.200
BoolQ	69.27	83.72	80.85	-2.87	0.350
MMLU	56.00	56.00	57.60	+1.60	0.200
GSM8K	61.33	62.67	62.00	-0.67	0.050
MATH-500	11.33	9.78	14.67	+4.89	0.100

Table 13: Gemma-2-2B detailed swept results. ICC surpasses the with-context teacher on 5 of 9 tasks.

C Detailed Capacity Sweeps

Tables 14–30 report ICC’s accuracy as a function of sparsity p for every (model, task) pair. “Top acc.” uses the sensitivity-ranked rows obtained from the score $s_{\ell,j}$ defined in Eq. 2 (§3.1); “Random acc.” is a matched-budget control in which the same number of rows is selected uniformly at random. The gap between the two isolates the contribution of the sensitivity score from the closed-form ridge solve of Eq. 3 (§3.2). “GI” (where reported) is the per-task gradient-instability indicator carried over from the PEFT runs of §4.3; it is included for reference and is not used by ICC. The corresponding swept best- p results are summarized per model in Table 1 of the main paper and detailed in Appendix B (Tables 11–13).

C.1 Mistral-7B Sweeps

p	Top acc.	Random acc.	Gap
0.05	82.71	80.98	+1.73
0.10	84.94	82.20	+2.73
0.15	84.71	83.01	+1.70
0.20	84.33	83.73	+0.59
0.25	83.83	83.97	-0.13
0.30	81.77	84.69	-2.92

Table 14: AG News, Mistral-7B. The sensitivity ranking helps most at small p ; as the edit budget grows, random rows catch up.

p	Top acc.
0.05	5.11
0.10	5.11
0.15	3.56
0.20	4.89
0.25	6.89
0.30	5.33
0.35	6.89

Table 15: MATH-500 sparsity sweep on Mistral-7B with $|G| = 16$. Accuracy is averaged over three seeds. The best fixed- p accuracy is 6.89% at both $p = 0.25$ and $p = 0.35$; Table 1 reports $p = 0.25$.

p	Top acc.	Random acc.	Gap
0.05	81.19	78.92	+2.27
0.10	85.40	79.77	+5.63
0.15	85.13	80.90	+4.23
0.20	86.62	81.31	+5.31
0.25	89.07	81.33	+7.73
0.30	88.84	81.97	+6.87

Table 16: DBPedia-14, Mistral-7B. The sensitivity ranking holds a +5–8pp gap over random across the full sweep, indicating the behavioral shift is concentrated rather than diffuse.

p	Top acc.	Random acc.	GI
0.05	83.94	80.39	26.9
0.10	88.65	80.28	59.7
0.15	88.30	79.59	64.2
0.20	88.30	81.88	64.2
0.25	87.61	83.94	65.7
0.30	87.39	83.83	64.2

Table 17: SST-2 capacity sweep on Mistral-7B. Performance peaks at $p = 0.10$, indicating that the useful with-context behavior is localized in a small subset of high-sensitivity rows. Larger edit budgets slightly degrade accuracy, while random row selection remains below the sensitivity-ranked edit at the optimum.

p	Top acc.	Random acc.	GI
0.05	78.92	78.43	34.2
0.10	80.88	78.43	43.9
0.15	79.17	78.92	56.1
0.20	77.94	79.41	58.5
0.25	76.96	79.90	65.9
0.30	75.25	79.90	73.2

Table 18: MRPC, Mistral-7B. Peaks early and degrades beyond $p = 0.10$: editing too many rows overwrites useful signal.

p	Top acc.	Random acc.
0.05	75.60	76.10
0.10	76.40	76.30
0.15	75.60	76.70
0.20	76.00	76.40
0.25	76.20	76.20
0.30	76.40	75.70

Table 19: ARC-Challenge, Mistral-7B. The edit has minimal room to operate when no-context already nears the with-context ceiling.

p	Top acc.	Random acc.	GI
0.05	33.33	36.00	18.5
0.10	36.67	34.00	25.9
0.15	36.00	31.33	29.6
0.20	36.00	35.33	32.1
0.25	40.00	32.00	33.3
0.30	42.00	35.33	37.0

Table 20: GSM8K, Mistral-7B. Generative reasoning requires larger edit budgets, consistent with the chain-of-thought signal being spread across more layers.

C.2 Phi-3-mini Sweeps

p	Top acc.	GI	Rows
0.05	42.00	34.71	4,915
0.10	40.00	34.71	9,830
0.15	50.67	41.32	14,746
0.20	47.33	42.15	19,661
0.25	50.67	52.89	24,576
0.30	50.67	48.76	29,491

Table 21: GSM8K, Phi-3-mini. “Rows” is the number of attention output-projection rows edited at each p .

p	Top acc.
0.05	24.22
0.10	22.00
0.15	22.67
0.20	24.00
0.25	24.67
0.30	24.44
0.35	27.11

Table 22: MATH-500 sparsity sweep on Phi-3-mini with $|G| = 8$. Accuracy is the mean over three seeds. The best fixed- p result, 27.11%, occurs at $p = 0.35$.

p	Top acc.	Random acc.	GI
0.05	76.72	71.49	72.5
0.10	71.08	69.77	85.0
0.15	50.49	65.93	81.7
0.20	60.29	64.95	81.7
0.25	46.57	63.15	81.7
0.30	43.38	56.37	81.7

Table 23: MRPC, Phi-3-mini. Sharp peak at $p = 0.05$ followed by collapse — a signature of a small, highly localized signal that gets diluted as the budget grows.

p	Top acc.	Random acc.
0.05	88.40	88.00
0.10	87.80	88.10
0.15	88.00	88.10
0.20	88.60	87.70
0.25	88.20	87.90
0.30	89.20	87.80

Table 24: ARC-Challenge capacity sweep on Phi-3-mini. Performance remains stable across edit budgets and peaks at $p = 0.30$. The small difference between top-ranked and random rows suggests that the useful with-context signal is less localized for this task.

p	Top acc.	Random acc.	GI
0.05	70.30	70.53	63.2
0.10	67.32	69.65	71.1
0.15	75.46	70.37	73.7
0.20	74.31	68.27	73.7
0.25	70.18	71.41	73.7
0.30	64.22	66.55	65.8

Table 25: AG News capacity sweep on Phi-3-mini. The edit peaks at $p = 0.15$, indicating that the useful behavioral shift is neither extremely localized nor fully diffuse. Larger budgets degrade performance, while random row selection does not recover the sensitivity-ranked peak.

p	Top acc.	Random acc.	Gap
0.050	80.96	59.23	+21.73
0.100	91.06	42.95	+48.11
0.150	63.30	42.32	+20.99
0.200	90.60	49.48	+41.11
0.250	90.37	42.72	+47.65
0.300	90.14	45.70	+44.44

Table 26: SST-2 capacity sweep on Gemma-2-2B. The edit peaks at $p = 0.10$ and consistently outperforms random row selection by large margins, indicating that the with-context behavior is concentrated in high-sensitivity rows. Larger budgets remain competitive but do not improve over the optimum, suggesting that additional edited rows add little useful signal.

p	Top acc.
0.05	11.78
0.10	14.67
0.15	12.44
0.20	10.44
0.25	12.67
0.30	14.22
0.35	11.33

Table 27: MATH-500 sparsity sweep on Gemma-2-2B with $|G| = 4$. Accuracy is the mean over three seeds. The best fixed- p result, 14.67%, occurs at $p = 0.10$.

p	Top acc.	Random acc.	Gap
0.050	55.39	68.63	−13.24
0.075	51.23	74.75	−23.53
0.100	53.68	74.39	−20.71
0.150	72.55	76.10	−3.55
0.200	76.47	73.90	+2.57
0.250	75.98	73.28	+2.70
0.300	74.02	73.90	+0.12

Table 28: MRPC capacity sweep on Gemma-2-2B. Performance peaks at $p = 0.20$, where the sensitivity-ranked edit slightly outperforms random row selection. The weaker gaps compared to other tasks suggest that the MRPC behavioral shift is less sharply localized and requires a moderate edit budget to recover.

p	Top acc.	Random acc.	Gap
0.050	80.73	39.16	+41.57
0.100	79.01	57.57	+21.44
0.150	74.43	70.81	+3.61
0.200	63.42	71.44	−8.03
0.250	44.72	68.58	−23.85
0.300	51.26	70.93	−19.67

Table 29: AG News, Gemma-2-2B. Strong peak at the smallest p tested (0.050), then steady collapse — the relevant rows are very few.

p	Top acc.	Random acc.	Gap
0.050	90.48	43.12	+47.36
0.100	89.33	43.52	+45.81
0.150	94.38	37.67	+56.71
0.200	95.07	58.60	+36.47
0.250	93.12	49.94	+43.18
0.300	94.27	61.47	+32.80

Table 30: DBPedia-14, Gemma-2-2B. Large and consistent gap across the full sweep — the sensitivity score finds rows that random selection misses by 30–65pp.

D Per-Task Split Sizes

Table 31 reports the exact size of each split for every task. $|D|$ is the demonstration pool

from which few-shot demonstrations are sampled, $|P|$ is the calibration pool from which the correction set G is constructed, $|G|$ is the upper bound on G used by the ridge solve, and n_{eval} is the held-out evaluation set on which final accuracy is reported. The three pools are pairwise disjoint.

Task	$ \mathcal{D}_{\text{demo}} $	$ \mathcal{D}_{\text{cal}} $	$ G $	$ \mathcal{D}_{\text{eval}} $
SST-2	100	500	4	872
MRPC	100	400	4	408
AG News	128	500	32	872
DBPedia-14	126	560	14	872
BoolQ	100	400	16	872
ARC-Challenge	120	300	16	500
GSM8K	128	500	8	150
MMLU	285	500	32	500
MATH-500	16	250	16	150

Table 31: Per-task data split sizes. $|\mathcal{D}_{\text{demo}}|$: demonstration pool size; $|\mathcal{D}_{\text{cal}}|$: calibration pool size; $|G|$: calibration budget for the correction set used by the ridge solve; $|\mathcal{D}_{\text{eval}}|$: evaluation set size. All three pools are sampled disjointly from the standard public splits; no calibration example appears in evaluation.

Reproducibility and Artifacts

Licenses and Intended Use. Our work utilizes standard open-source models (Mistral-7B, Phi-3-mini, and Gemma-2-2B) and canonical academic benchmarks (MMLU, GSM8K, SST-2, MRPC, AG News). All artifacts were used in strict accordance with their respective open-source licenses and intended research purposes. These models and datasets are widely adopted in the NLP community, and their usage here is consistent with standard academic practice.

Documentation and Statistics. The datasets used are standard benchmarks; their linguistic coverage, language (English), and demographic representations are documented in their respective foundational papers, which we cite in §4.1. The data partitioning strategy (demonstration pool, calibration pool, and held-out evaluation set) is detailed in §4.1. All evaluation sets are held-out subsets drawn from the standard public dataset splits.