HYPERREPAIR: RECOVERING FORGOTTEN GENERAL CAPABILITIES OF FINE-TUNED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning improves a language model's performance in a target domain but can weaken its general capabilities. Existing methods often face a trade-off between general capabilities and domain specialization, making it difficult to preserve both. We introduce HyperRepair, a framework for recovering general capabilities while retaining domain specialization at only inference cost. To make recovery generalizable across models sharing a pretrained backbone, HyperRepair learns a hypernetwork from repairs trained on source models. The hypernetwork uses activation fingerprints that summarize changes from the pretrained backbone to generate a tailored Repairing Side Net (RSN), whose prompt-conditioned gate applies corrections selectively. This enables recovery on models from unseen adaptation domains without target-domain examples or additional optimization. We evaluate HyperRepair across two model families, 12 LoRA adaptation datasets, and 15 general benchmarks. On the default split, HyperRepair achieves 93.4% of the base models' average general score while retaining 99.5% of the fine-tuned models' average domain score.

1 INTRODUCTION

The widespread use of pretrained large language models creates a need to adapt them to diverse downstream tasks, domains, and application requirements. Fine-tuning improves downstream performance but can degrade pretrained knowledge and general capabilities (Wang et al., 2024; Kotha et al., 2024; Luo et al., 2025), a phenomenon known as catastrophic forgetting (McCloskey & Cohen, 1989). Such degradation can impair responses beyond the adaptation task, limiting the usefulness of specialized models when applications also require broad knowledge and reasoning. Although low-rank adaptation (LoRA) trains only a small number of additional parameters (Hu et al., 2022), studies show that catastrophic forgetting can still occur (Kalajdzievski, 2024; Yang et al., 2024b).

Existing approaches mitigate forgetting during fine-tuning through parameter regularization (Kirkpatrick et al., 2017), rehearsal and distillation (Huang et al., 2024b; Li & Hoiem, 2016), or changes to adapter construction and loss weighting (Yang et al., 2024b; Sanyal et al., 2025; Lin et al., 2026). However, these approaches intervene during adaptation and cannot directly repair an already fine-tuned model without further training. Post-hoc approaches instead modify trained weights through model merging (Ilharco et al., 2023; Yadav et al., 2023; Huang et al., 2024a; Yu et al., 2024), layerwise rescaling (Wang et al., 2025), or forgetting-aware pruning (Huang et al., 2025). Post-hoc rescaling and pruning can recover general performance without repeating the original fine-tuning process. However, methods that apply a single set of modified weights to all inputs can face a trade-off between recovering general capabilities and retaining downstream specialization. We therefore pose and study the following question:

> *Given a fine-tuned language model with degraded general capabilities, how can we efficiently recover its general performance at inference cost only while retaining downstream task performance?*

This motivates a selective approach to recovery: we keep the fine-tuned weights fixed and place the correction in a separate pathway. This retains the original model as a fallback and makes it possible to decide from the prompt whether a correction should be used. Side networks provide a natural starting point for this design because they can reuse intermediate representations from a

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.

frozen backbone without rewriting its parameters (Zhang et al., 2020; Sung et al., 2022). A separate pathway alone, however, does not solve the entire problem. If a new repair must be trained for every newly released adapter, the cost of recovery still grows with the number of models. What is missing is a way to learn from repaired models and use that experience to repair another model.

To make recovery reusable across models, we consider models adapted independently from the same pretrained backbone. Some of these models serve as sources from which repairs can be learned. The target model comes from an adaptation domain that was not used to learn the repair mechanism. Its domain data is unavailable, and no optimization is allowed on the target model. To our knowledge, this is the first work to study amortized recovery of general capabilities across independently fine-tuned language models. The goal is not merely to find a better compromise for one checkpoint, but to learn a repair process that generalizes to an unseen downstream domain.

Building on this formulation, we introduce HyperRepair, which learns this repair process. To prepare data for learning how to repair, we first train a lightweight Repairing Side Net (RSN) for each fine-tuned source model. The RSN reads intermediate hidden states and produces a residual correction before the language model head. Labeled general examples teach it how to improve general predictions. A prompt-conditioned gate determines how much of the correction should be applied. The gate is trained to activate on general prompts and to remain inactive on unlabeled domain prompts. When the correction is inactive, the model produces the original fine-tuned predictions. The model can therefore use the repair where it is helpful without giving up direct access to the behavior learned for a downstream domain.

The remaining question is how to obtain a suitable RSN for a new model without training one. We represent each fine-tuned model by an activation fingerprint. The fingerprint records how its hidden representations differ from those of the pretrained model on a fixed group of unlabeled general prompts. We train a hypernetwork, HyperNet (Ha et al., 2017; Aliev et al., 2026), to map this fingerprint to the complete parameters of the RSN, including its prompt gate. Training uses the parameters of directly trained source repairs together with supervision on the behavior of the generated repairs. At deployment, HyperNet receives the fingerprint of a model from a held-out adaptation domain and generates its RSN in one forward pass. No target-domain examples or optimization are required.

We summarize our main contributions as follows:

1. We formulate post-hoc recovery as the activation-conditioned generation of model-specific Repairing Side Nets (RSNs) that recover general capabilities while retaining downstream performance. Given a fine-tuned checkpoint, our method generates its repair from activation fingerprints without additional optimization or target-domain examples.
2. We design HyperRepair, a hypernetwork-based framework that predicts the weights of a lightweight RSN, including its prompt-conditioned gate. The RSN applies selective residual corrections to the frozen backbone, while HyperNet learns from source repairs and generalizes to new fine-tuned models.
3. We show that generated RSNs can recover general capabilities while preserving domain performance. We evaluate two backbones across 12 adaptation datasets and 15 general benchmarks, training HyperNet on source models and testing on held-out domains. On the default split, HyperRepair reaches 93.4% of the base models' average general score and retains 99.5% of the fine-tuned models' average domain score.

2 Related Work

Forgetting in Fine-Tuned LLMs. Domain-adaptive pretraining (Gururangan et al., 2020) and supervised fine-tuning (Howard & Ruder, 2018) can improve target-task performance, while multitask instruction tuning (Wei et al., 2022; Sanh et al., 2022) can improve generalization to unseen tasks. Fine-tuning can nevertheless degrade existing capabilities. Longer pretraining can increase sensitivity to parameter updates (Springer et al., 2025), while smaller fine-tuning learning rates can reduce degradation (Lin et al., 2026). Aggregate scores can conceal simultaneous gains and losses on individual examples (Harmon et al., 2026). Analyses of topic and style shifts (Zhang & Wu, 2024), task inference (Kotha et al., 2024), function activation (Jiang et al., 2025), and task alignment (Zheng et al., 2025) suggest that some declines reflect how retained knowledge is accessed and applied.

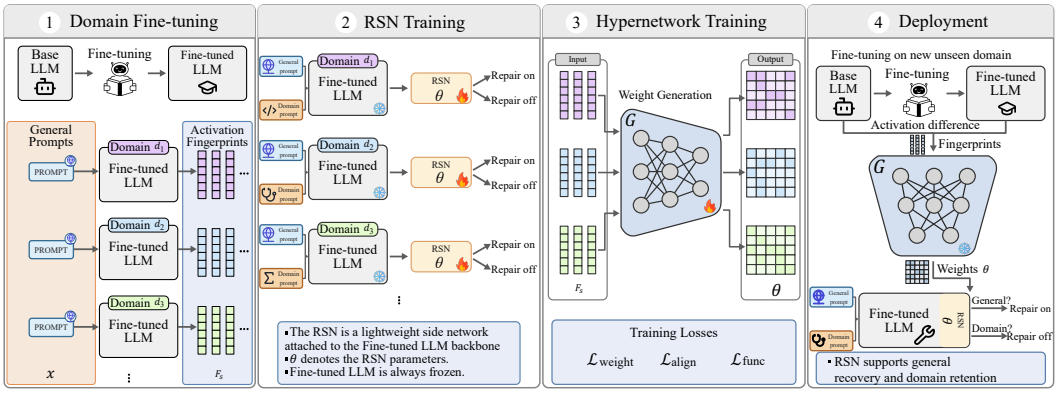


Figure 1: Overview of HyperRepair. (1) Domain fine-tuning produces source models, whose activations on general prompts are used to construct fingerprints. (2) The gated RSN is trained for each source model while keeping its backbone frozen. (3) A HyperNet learns to generate RSN parameters from activation fingerprints using weight, representation-alignment, and functional supervision. (4) For a fine-tuned model from an unseen adaptation domain, the frozen HyperNet generates RSN parameters without further optimization or target-domain data. The gate controls whether the repair is applied.

We study whether general capabilities can be recovered after domain fine-tuning while retaining the domain capabilities acquired during adaptation.

Mitigating Fine-Tuning-Induced Forgetting. Classical continual learning mitigates forgetting through parameter regularization (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018), distillation (Li & Hoiem, 2016), gradient constraints (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019), and replay (Shin et al., 2017; Buzzega et al., 2020). LAMOL (Sun et al., 2020) and Self-Synthesized Rehearsal (Huang et al., 2024b) generate examples for language-model rehearsal. CorDA's knowledge-preserved mode (Yang et al., 2024b) initializes adapters using calibration activations, and FLOW (Sanyal et al., 2025) upweights examples with low pretrained-model loss. Post-hoc approaches include weight interpolation for visual robustness (Wortsman et al., 2022b), checkpoint averaging (Wortsman et al., 2022a), and merging with Fisher information (Matena & Raffel, 2022) or matching layer outputs (Jin et al., 2023). Several model-merging methods combine task updates (Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Huang et al., 2024a; Yang et al., 2024a), while LiNeS (Wang et al., 2025) scales updates by layer depth. Side-Tuning (Zhang et al., 2020) trains a task-specific side network and blends its features with those of a frozen pretrained model. Our RSNs recover general capabilities after fine-tuning and are generated by HyperNet for new models sharing the same backbone. HyperRepair generates a gated repair pathway alongside frozen fine-tuned weights, preserving the original predictions when correction is disabled.

3 METHOD

HyperRepair aims to recover general capabilities lost during domain fine-tuning while retaining domain performance. Figure 1 gives an overview of the framework. The framework combines an RSN, a gated side network that corrects the frozen model's final hidden states, with HyperNet, which predicts the repair parameters from the model's activation fingerprint. We train HyperNet using fingerprints paired with RSN parameters learned for source models. For a model from an unseen domain, the trained generator produces a repair without further optimization or examples from that domain. The following sections introduce the problem, the two networks, and the training and deployment procedure.

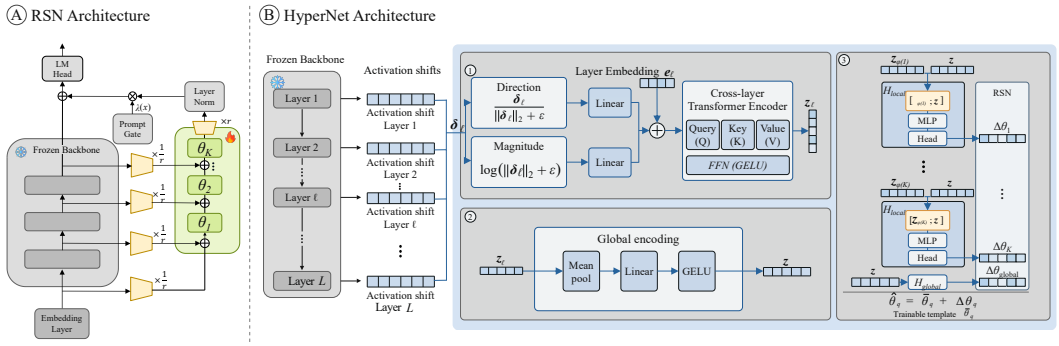


Figure 2: Architectures of the RSN and HyperNet. (A) The RSN processes intermediate states from the frozen backbone. The prompt-conditioned gate $\lambda(x)$ scales its final output before addition to the backbone state. The backbone and language-model head remain frozen throughout repair training. (B) HyperNet encodes layerwise activation shifts with a Transformer and pools the encodings into a global representation. It adds generated local and global updates to a trainable shared template.

3.1 Problem Formulation

Let M_0 denote a pretrained language model. Fine-tuning M_0 on domain d produces the model M_d. This adaptation can improve domain performance while reducing performance on general tasks. Our goal is to recover general capabilities and retain the domain capabilities acquired during fine-tuning. All parameters of M_d remain frozen during repair.

We seek a HyperNet G that generates an RSN for a fine-tuned model from a potentially unseen domain. Given the model's activation fingerprint $\mathbf{F}_d$, parameter generation and prompt gating are expressed as

$$\widehat{\theta}_d = G(\mathbf{F}_d), \qquad \lambda(x) = f_{\hat{\theta}_d}(x; M_d). \tag{1}$$

Here $\mathbf{F}_d$ summarizes how M_d's activations differ from those of M_0 on fixed unlabeled general prompts. The generated parameters $\widehat{\theta}_d$ specify the complete RSN, including its correction network and prompt gate. The function $f_{\hat{\theta}_d}$ denotes its gate computation using prompt x and the frozen model's activations. Its output $\lambda(x) \in [0, 1]$ controls the strength of the correction described in Section 3.2. Each generator is trained on source models and applied to held-out models from domains excluded from that training. These models share the same pretrained backbone M_0.

3.2 Repairing Side Net

The RSN attaches to the frozen backbone and produces a corrective residual together with a prompt gate. We adopt the side tuning technique to modify the backbone's final hidden states (Zhang et al., 2020; Sung et al., 2022). Figure 2A shows how the correction enters the model's predictions.

Side Transformer. The RSN uses K Transformer layers with hidden dimension $h_s = h/r$, where h is the backbone hidden size and r is the reduction factor. We index the backbone layers by $\ell = 1, \ldots, L$ and the side layers by $k = 1, \ldots, K$. The initial side state $\mathbf{h}_0^{\text{side}}$ is obtained by projecting the backbone's input embeddings from h to h_s dimensions.

Ladder Connections. Each side layer combines a projected backbone hidden state with the output of the preceding side layer. Omitting projection biases, the input to side layer k is

$$\mathbf{h}_k^{\text{in}} = \mu_k \mathbf{W}_k^{\text{down}} \mathbf{h}_{\phi(k)}^{M_d} + (1 - \mu_k)\mathbf{h}_{k-1}^{\text{side}}. \tag{2}$$

Here $\phi : \{1, \ldots, K\} \to \{1, \ldots, L\}$ maps each side layer to its connected backbone layer, and $\mathbf{W}_k^{\text{down}} \in \mathbb{R}^{h_s \times h}$ projects that backbone layer's hidden state to the side-network width. The learned scalar $\mu_k \in (0, 1)$ controls the contribution of the backbone state. The side Transformer block maps $\mathbf{h}_k^{\text{in}}$ to its output $\mathbf{h}_k^{\text{side}}$.

Selective Repair. After the final side layer, we project its output from h_s to h and normalize it to obtain the correction $\mathbf{h}^{\text{side}}_{\text{final}}$. We average these states over prompt tokens and pass the result through a small multilayer perceptron (MLP) and sigmoid to produce $\lambda(x)$. For each token, we add the gated correction to the backbone's final hidden state $\mathbf{h}^{M_d}_{\text{final}}$ before the frozen language-model head:

$$\mathbf{h}^{\text{repaired}}_{\text{final}} = \mathbf{h}^{M_d}_{\text{final}} + \lambda(x)\mathbf{h}^{\text{side}}_{\text{final}}. \tag{3}$$

The gate should be near one on general prompts to apply the correction. On domain prompts, it should be near zero to preserve the fine-tuned model's behavior. When the gate is zero, the correction is disabled and the model produces the original fine-tuned predictions. The gate depends only on the prompt and is shared when scoring answer candidates or generating a response. During training, we use the unthresholded sigmoid gate.

3.3 Activation-Conditioned Hypernetwork

HyperNet generates the parameters of an RSN from the activation fingerprint of a fine-tuned model (Figure 2B). We encode activation shifts by direction and magnitude and use low-rank factors to generate matrix updates.

Activation Fingerprint Extraction. To compare models under the same inputs, we use a fixed group of N unlabeled general prompts $\{x_i\}_{i=1}^{N}$. Let $\mathbf{h}^{M}_{\ell}(x)$ denote model M's raw hidden state at decoder layer ℓ, evaluated at the final token of prompt x. We measure the mean activation change relative to the pretrained model:

$$\boldsymbol{\delta}_\ell = \frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{h}^{M_d}_{\ell}(x_i) - \mathbf{h}^{M_0}_{\ell}(x_i)\right]. \tag{4}$$

Using general prompts allows us to characterize a new model without data from its domain.

Direction-Magnitude Decomposition. We represent each layer's activation change by its direction and log magnitude:

$$\mathbf{d}_\ell = \frac{\boldsymbol{\delta}_\ell}{\|\boldsymbol{\delta}_\ell\|_2 + \epsilon}, \qquad m_\ell = \log(\|\boldsymbol{\delta}_\ell\|_2 + \epsilon)\,. \tag{5}$$

Here ϵ is a small constant for numerical stability. The fingerprint $\mathbf{F}_d$ collects $\mathbf{d}_\ell$ and m_ℓ across all backbone layers. A Transformer encodes these features together with learned layer embeddings, producing a local fingerprint encoding $\mathbf{z}_\ell$ for each backbone layer. Pooling the local encodings gives the global fingerprint encoding $\mathbf{z}$.

Layerwise Factorized Generation. HyperNet generates RSN parameters as a shared template plus an update conditioned on the fine-tuned model:

$$\widehat{\theta}_d = \bar{\theta} + \Delta\theta_d. \tag{6}$$

The trainable template $\bar{\theta}$ is shared across models. The update $\Delta\theta_d$ depends on the fine-tuned model's fingerprint. For side layer k, we generate its update from the corresponding local encoding and the global encoding:

$$\Delta\theta_k = H_{\text{local}}\big(\mathbf{z}_{\phi(k)}, \mathbf{z}\big)\,. \tag{7}$$

Here $\mathbf{z}_{\phi(k)}$ is the local fingerprint encoding of the backbone layer connected to side layer k. The local generator H_{local} concatenates this local encoding with the global encoding $\mathbf{z}$ and processes them with an MLP followed by parameter heads. The MLP and heads for each parameter role share weights across side layers. Matrix updates are products of low-rank factors predicted by the heads, reducing the number of generated values. Vector and scalar updates are predicted directly.

Global Parameter Generation. A separate generator H_{global} takes $\mathbf{z}$ and produces updates for the output projection, output normalization, and prompt-gate network. All updates are added to the corresponding shared template parameters.

3.4 Training and Deployment

We first train an RSN separately for each frozen source model. These trained networks then supply the parameter targets used to train HyperNet.

Source RSN Training. For each source model M_d, we train an RSN and use its learned parameters $\theta_d^\star$ as supervision targets. Labeled general examples teach it to correct predictions. General and unlabeled domain prompts provide gate supervision, with targets one and zero, respectively. Let $\mathcal{L}_{\text{general}}$ denote cross-entropy over candidate answers for discriminative tasks or mean response-token cross-entropy for generative tasks. The training objective is

$$\mathcal{L}_{\text{func}} = \mathcal{L}_{\text{general}} + \mathcal{L}_{\text{gate}}. \tag{8}$$

Here $\mathcal{L}_{\text{gate}}$ is the binary cross-entropy loss that supervises the prompt gate on general and domain inputs. Domain answers are not required because domain prompts contribute only gate supervision.

Hypernetwork Training. Each source model supplies a fingerprint $\mathbf{F}_d$ and ground-truth RSN parameters $\theta_d^\star$. We train HyperNet to reconstruct these parameters and optimize the predictions made with the generated RSN. The combined objective is

$$\mathcal{L}_{\text{hyper}} = \beta_{\text{weight}}\mathcal{L}_{\text{weight}} + \beta_{\text{align}}\mathcal{L}_{\text{align}} + \beta_{\text{func}}\mathcal{L}_{\text{func}}, \tag{9}$$

where $\mathcal{L}_{\text{weight}}$ is the mean-squared error between the generated and ground-truth RSN weights ($\widehat{\theta}_d$ and $\theta_d^\star$). $\mathcal{L}_{\text{align}}$ is the mean-squared error between the normalized gate-input features of the generated and ground-truth RSNs. The functional loss $\mathcal{L}_{\text{func}}$ applies Equation 8 to the generated RSN attached to its frozen source model. Because the prompt gate reads the correction states, errors in generated RSN parameters can affect both the correction and the gating decision. Representation alignment constrains the features used by the gate, complementing weight reconstruction and functional supervision. Training updates only HyperNet, including the shared template. The backbones and ground-truth RSNs remain frozen.

Deployment. For a held-out M_d, we compute its fingerprint using the fixed general prompts, evaluate HyperNet once, and attach the generated RSN. The repair modifies predictions through Equation 3, using the generated prompt gate to control its strength. Generating this repair requires no additional optimization or examples from the held-out model's domain.

4 Experiments

4.1 Experimental Setup

Models and Domains. We evaluate the base checkpoints of Qwen2.5-7B (Qwen Team, 2024) and Llama-3.1-8B (Grattafiori et al., 2024) on 12 domains spanning classification, question answering, semantic parsing, and code summarization. We treat each dataset–task setting as an adaptation domain. For each backbone and domain, we fine-tune a LoRA adapter (Hu et al., 2022), which remains active during repair.

General Benchmarks. We measure general capabilities on 15 general benchmarks: 12 discriminative tasks and three generative tasks, MATH, HumanEval, and MBPP. Domain-task scores measure retention of the capabilities acquired during fine-tuning. We report individual general-benchmark scores and their unweighted mean as the average general score.

Evaluation Protocol. We use four splits, each holding out three of the 12 domains. For each backbone and split, we train a separate HyperNet on the remaining nine domains and evaluate its generated RSNs on the three held-out models. The main comparison uses the default split (Split D), which holds out LEDGAR-Top10, SNIPS-7, and DROP.

Parameter and inference overhead. The RSN adds 9.57M parameters to Qwen2.5-7B and 10.56M to Llama-3.1-8B, approximately 0.13% of each backbone. For each adapted model, repair generation requires fingerprint extraction on 150 unlabeled general prompts and one HyperNet forward pass. The generated parameters are reused across subsequent requests, while the RSN runs at every decoding step.

Table 1: General benchmark test scores for Qwen2.5-7B and Llama-3.1-8B on Split D. Base is shared across held-out domains within each backbone. HyperRepair uses RSNs generated by a HyperNet trained on the other nine domains.

Method	ARC-Challenge	HellaSwag	MMLU	OpenBookQA	PIQA	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA	WiC	COPA	MATH	HumanEval	MBPP
Qwen2.5-7B															
Base	51.4	78.8	71.8	47.4	79.8	95.2	73.1	53.0	54.7	35.8	57.8	91	39.3	56.1	63.8
LEDGAR-Top10															
Fine-tuned	53.8	73.1	55.0	46.0	79.4	88.1	70.8	48.5	49.0	33.6	50.0	88	3.6	22.0	28.8
CorDA-KPM	45.7	65.7	24.0	44.2	75.7	78.1	69.6	33.0	46.8	27.0	50.0	90	0.0	4.3	0.0
FAPM	55.4	75.0	**63.1**	48.0	80.5	91.3	71.9	49.0	**51.1**	33.3	50.0	89	28.5	**34.8**	**44.0**
FLOW	27.6	41.5	28.9	31.8	60.4	50.5	50.7	31.4	37.9	29.3	**50.2**	57	0.3	9.8	25.0
HyperRepair	**65.4**	**84.3**	62.9	**51.2**	**84.8**	**93.3**	**75.4**	**54.7**	**51.1**	**34.1**	50.0	**95**	**39.4**	**34.8**	41.8
SNIPS-7															
Fine-tuned	44.8	70.3	61.8	41.8	75.1	87.3	68.7	37.1	38.0	31.0	50.3	85	0.0	1.8	0.0
CorDA-KPM	32.6	53.2	24.0	38.2	67.8	71.5	62.4	32.2	37.6	27.5	50.0	85	0.0	0.0	0.0
FAPM	46.7	73.1	68.2	42.4	76.6	92.3	69.8	**42.1**	39.3	31.6	52.4	82	0.0	14.0	12.4
FLOW	53.0	74.6	**69.6**	43.6	77.7	**95.6**	70.3	41.7	**46.5**	**33.8**	**61.8**	88	9.8	32.3	**27.6**
HyperRepair	**55.7**	**80.9**	64.8	**46.6**	**78.1**	89.5	**71.3**	**42.1**	38.0	30.9	50.3	**90**	**35.8**	**36.0**	23.2
DROP															
Fine-tuned	59.9	76.1	64.8	46.8	79.8	**97.3**	75.1	47.0	54.9	44.9	65.8	90	33.4	51.8	58.8
CorDA-KPM	50.8	66.1	43.6	44.2	76.6	96.5	67.8	36.1	51.7	27.5	52.7	90	0.0	8.5	10.6
FAPM	61.2	76.6	**66.4**	45.2	80.5	**97.3**	74.4	47.6	54.9	**46.1**	**66.3**	91	39.7	50.6	**60.6**
FLOW	61.3	76.5	66.2	47.8	79.9	97.0	74.0	44.9	55.6	43.2	59.4	91	25.7	**53.0**	**60.6**
HyperRepair	**70.4**	**87.6**	**66.4**	**52.8**	**84.2**	**97.3**	**80.7**	**52.3**	**59.3**	42.7	62.7	**95**	**40.1**	52.4	56.8
Llama-3.1-8B															
Base	54.7	79.3	64.0	45.4	81.0	95.6	74.4	35.1	48.4	31.0	51.1	92	19.9	36.6	47.6
LEDGAR-Top10															
Fine-tuned	40.1	62.6	28.1	42.8	74.0	69.0	71.0	36.2	40.7	**31.2**	**50.0**	86	0.0	0.0	0.0
CorDA-KPM	48.8	73.6	25.5	43.6	79.0	**88.7**	73.6	33.3	**48.0**	29.5	**50.0**	90	0.0	**22.6**	**20.2**
FAPM	41.5	67.1	**31.3**	42.6	76.2	75.6	71.7	36.3	41.8	30.7	**50.0**	88	0.0	0.0	0.0
FLOW	39.8	69.2	24.2	43.8	76.5	81.3	73.4	32.5	42.8	**31.2**	**50.0**	**91**	0.0	1.8	0.0
HyperRepair	**49.5**	**75.9**	28.4	**44.6**	**79.2**	81.3	**76.8**	**36.9**	43.7	29.6	**50.0**	**91**	**10.8**	14.6	17.6
SNIPS-7															
Fine-tuned	38.1	68.3	23.5	41.0	75.9	76.7	70.7	33.2	40.3	29.3	49.8	88	0.0	12.2	1.2
CorDA-KPM	**47.7**	76.4	34.0	43.4	78.6	**93.4**	70.9	31.2	**43.7**	29.6	50.0	89	0.1	19.5	16.4
FAPM	39.4	70.5	23.7	41.0	77.1	80.2	71.1	33.2	41.7	30.1	49.5	89	0.0	16.5	6.2
FLOW	41.2	68.2	**34.8**	40.4	76.6	84.2	70.0	32.6	39.6	27.6	50.8	86	0.0	3.7	0.0
HyperRepair	**47.7**	**76.7**	28.1	**47.2**	**79.0**	81.5	**75.4**	**41.4**	42.4	**30.6**	**51.1**	**90**	**9.4**	**20.7**	**16.8**
DROP															
Fine-tuned	55.2	76.2	33.2	47.2	79.8	96.3	74.5	32.1	51.5	27.5	53.1	92	0.0	13.4	25.4
CorDA-KPM	54.9	74.7	35.0	42.8	79.2	96.9	72.1	31.6	51.0	27.8	50.3	88	0.0	20.1	18.6
FAPM	56.1	76.9	35.8	47.0	80.3	96.4	74.7	32.4	51.9	28.3	54.7	92	0.0	17.1	30.2
FLOW	55.2	76.1	29.6	46.0	79.9	96.9	75.0	33.0	50.9	**28.6**	50.0	92	0.0	15.9	33.4
HyperRepair	**60.6**	**83.2**	**38.5**	**53.4**	**83.1**	**97.2**	**79.8**	**41.8**	**53.0**	28.1	**61.0**	**93**	**8.2**	**22.6**	**34.4**

4.2 Main Results

General recovery and domain retention. HyperRepair improves the average general score of all six held-out models in Table 1, reaching 93.4% of the base models' average general score. Across the 12 held-out domains, domain performance remains close to that of the fine-tuned models (Figure 3); on Split D, HyperRepair retains 99.5% of their average domain score. The gating ablation on Qwen2.5-7B supports the role of selective repair: keeping the repair always active raises the average general score but reduces the average domain score (Section 4.3).

Comparison with baselines. HyperRepair achieves the highest average general score on both backbones among the compared methods (Table 1). Averaged over 15 benchmarks and three held-out domains, it reaches 61.2 on Qwen2.5-7B and 51.2 on Llama-3.1-8B, compared with 57.0 and

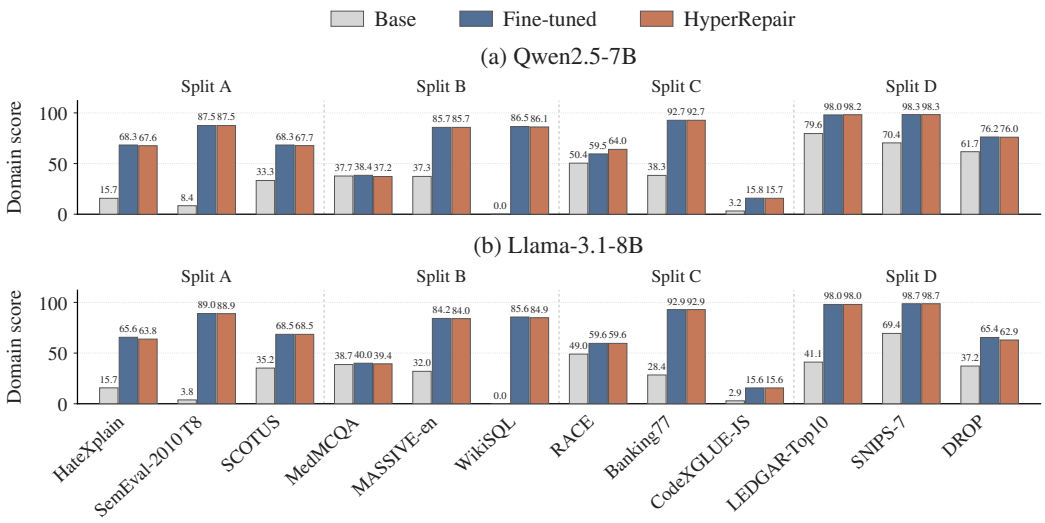


Figure 3: Domain test performance across the 12 held-out adaptation domains. HyperRepair generally retains the scores of the fine-tuned models on both backbones.

48.7 for the strongest respective baselines, FAPM and CorDA-KPM. Because these models come from adaptation domains excluded from HyperNet training, the gains demonstrate that the learned repair mechanism generalizes beyond its source models. Activation fingerprints let HyperNet tailor this mechanism to the changes induced by fine-tuning in each target model. On Qwen2.5-7B, a shared RSN without fingerprint conditioning reduces the average general score (Section 4.3), supporting the value of model-specific repairs.

4.3 Ablation Studies

We conduct ablation studies on Qwen2.5-7B, reporting the changes in average general score and domain score, averaged across the three held-out domains in Split D. The full system achieves the best overall performance. Removing any component degrades at least one metric.

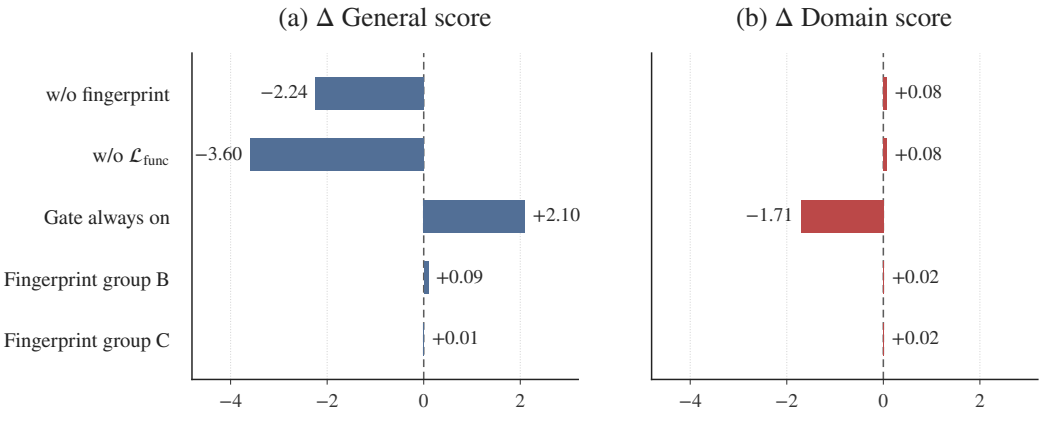


Figure 4: Ablation study on Qwen2.5-7B across the three held-out domains in Split D. Values are changes in general and domain scores relative to the full model, measured in score points.

Loss components. Removing $\mathcal{L}_{\text{func}}$ while retaining $\mathcal{L}_{\text{weight}}$ and $\mathcal{L}_{\text{align}}$ reduces general recovery. Matching source RSN weights and features alone is less effective than also supervising the predictions and gate of the generated repair.

Activation fingerprints. We remove the fingerprint-conditioned updates and train only the shared RSN template, keeping the three losses and training budget unchanged. All held-out models use the same repair parameters, while prompt-conditioned gating is retained. Using a shared RSN reduces the average general score by 2.24 points, with an increase of 0.08 points in domain score. This supports the value of tailoring repair parameters to individual fine-tuned models.

Fingerprint sampling sensitivity. Aggregate scores change little when the fingerprint is recomputed from different prompts. We compare three disjoint groups of 150 general prompts with matched task composition, denoted A, B, and C. Group A is used during HyperNet training; evaluation substitutes B or C while holding the fine-tuned model and HyperNet fixed. These results support low sensitivity to the tested prompt substitutions.

Prompt-conditioned gating. Keeping the repair always active ($\lambda(x) = 1$) reduces domain performance, showing that applying corrections to every prompt can interfere with domain capabilities. These results highlight the importance of controlling when and how strongly the repair is applied.

4.4 Qualitative Results

Table 2 shows two representative recovery examples on Qwen2.5-7B. On ARC-Challenge, fine-tuning on HateXplain changes the correct prediction, while HyperRepair restores it. On MATH, fine-tuning on MASSIVE-en leads to an error in handling repeated letters, and HyperRepair recovers the correct answer of 1260. More examples are provided in Section E.

Table 2: Recovery examples on Qwen2.5-7B for ARC-Challenge and MATH. HyperRepair restores the correct outputs after fine-tuning errors.

Base (correct)	**Fine-tuned (incorrect)**	**HyperRepair (correct)**
ARC-Challenge (discriminative)		*Adaptation: HateXplain (Split A)*
Prompt. An astronomer observes that a planet rotates faster after a meteorite impact. Which is the most likely effect of this increase in rotation?		
Planetary days will become shorter.	Planetary gravity will become stronger.	Planetary days will become shorter.
MATH (generative)		*Adaptation: MASSIVE-en (Split B)*
Prompt. Determine the number of ways to arrange the letters of the word ELLIPSE.		
The word ELLIPSE has 7 letters, with 2 L's and 2 E's. The number of ways to arrange the letters is $\frac{7!}{2!2!} = \boxed{1260}$.	The word ELLIPSE has 7 letters, but the letter E appears twice. Therefore, the number of ways to arrange the letters is $\frac{7!}{2!} = \boxed{2520}$.	The word ELLIPSE has 7 letters, with 2 L's and 2 E's. The number of ways to arrange the letters is $\frac{7!}{2!2!} = \boxed{1260}$.

5 Conclusion

HyperRepair generates model-specific RSNs that recover general capabilities in frozen fine-tuned language models across diverse adaptation domains and two model families while maintaining strong domain performance. Two findings highlight its practical value: a hypernetwork trained on source models generates effective repairs for models from unseen adaptation domains without target-domain examples or additional optimization, and prompt-conditioned gating preserves domain capabilities that deteriorate when the repair is always active.

Limitations. The extent of recovery may vary across tasks and adaptation domains. Further work could examine how task characteristics and fine-tuning-induced changes influence recovery, with the aim of achieving more consistent performance across different settings.

AI USE STATEMENT

Generative AI tools assisted with interpreting experimental results. They also supported literature search and summarization, manuscript drafting and revision, and LaTeX editing, including table formatting and figure placement. AI tools were not used for method or experiment design, method implementation, data generation or processing, or translation. The authors take responsibility for the final manuscript, including AI-assisted text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

We describe the RSN and HyperNet architectures, training objectives, and deployment procedure in Section 3, with implementation and training settings in Section A. The datasets, scoring protocol, held-out domain partitions, and fingerprint prompt construction appear in Sections B to D.

REFERENCES

Wasi Uddin Ahmad, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Vahid Noroozi, Somshubra Majumdar, and Boris Ginsburg. OpenCodeInstruct: A large-scale instruction tuning dataset for code LLMs. *arXiv preprint arXiv:2504.04030*, 2025.

Aznaur Aliev, Carlos Hinojosa, Abdelrahman Eldesokey, Bang An, Bernard Ghanem, and Yibo Yang. HyperSafe: Inference-time safety recovery for fine-tuned language models. *arXiv preprint arXiv:2607.11475*, 2026.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision (ECCV)*, 2018.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. SLURP: A spoken language understanding resource package. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024. URL https://arxiv.org/abs/2405.14782v2.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Workshop on Natural Language Processing for Conversational AI (NLP4ConvAI)*, 2020.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations (ICLR)*, 2019.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

David Ha, Andrew Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations (ICLR)*, 2017.

Jackson Harmon, Andreas Hochlehnert, Matthias Bethge, and Ameya Prabhu. Mapping post-training forgetting in language models at scale. In *International Conference on Learning Representations (ICLR)*, 2026.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *International Workshop on Semantic Evaluation (SemEval)*, 2010.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2021b.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

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

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. EMR-Merging: Tuning-free high-performance model merging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024b.

Wei Huang, Anda Cheng, and Yinggui Wang. Mitigating catastrophic forgetting in large language models with forgetting-aware pruning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Gangwei Jiang, Caigao Jiang, Zhaoyi Li, Siqiao Xue, Jun Zhou, Linqi Song, Defu Lian, and Ying Wei. Unlocking the power of function vectors for characterizing and mitigating catastrophic forgetting in continual instruction tuning. In *International Conference on Learning Representations (ICLR)*, 2025.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Damjan Kalajdzievski. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations (ICLR)*, 2024.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, 2016.

Jiacheng Lin, Zhongruo Wang, Kun Qian, Tian Wang, Arvind Srinivasan, Hansi Zeng, Ruochen Jiao, Xie Zhou, Jiri Gesi, Dakuo Wang, Yufan Guo, Kai Zhong, Weiqi Zhang, Sujay Sanghavi, Changyou Chen, Hyokun Yun, and Lihong Li. SFT doesn't always hurt general capabilities: Revisiting domain-specific fine-tuning in LLMs. In *International Conference on Learning Representations (ICLR)*, 2026.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. CodeXGLUE: A machine learning benchmark dataset for code understanding and generation. In *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2021.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/taslpro.2025.3606231.

Michael S Matena and Colin Raffel. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. HateXplain: A benchmark dataset for explainable hate speech detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Elsevier, 1989. doi: 10.1016/s0079-7421(08)60536-8.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning (CHIL)*, 2022.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

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

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense)*, 2011.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng-Xin Yong, Harshit Pandey, Michael McKenna, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations (ICLR)*, 2022.

Sunny Sanyal, Hayden Prairie, Rudrajit Das, Ali Kavis, and Sujay Sanghavi. Upweighting easy samples in fine-tuning mitigates forgetting. In *International Conference on Machine Learning (ICML)*, 2025.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. Overtrained language models are harder to fine-tune. In *International Conference on Machine Learning (ICML)*, 2025.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: LAnguage MOdeling for lifelong language learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. LST: Ladder Side-Tuning for parameter and memory efficient transfer learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Language Resources and Evaluation Conference (LREC)*, 2020.

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. LiNeS: Post-training layer scaling prevents forgetting and enhances model merging. In *International Conference on Learning Representations (ICLR)*, 2025.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit Dhillon, and Sanjiv Kumar. Two-stage LLM fine-tuning with less specialization and more generalization. In *International Conference on Learning Representations (ICLR)*, 2024.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*, 2022.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Workshop on Noisy User-generated Text (WNUT)*, 2017.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are Super Mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning (ICML)*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, 2017.

Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-Tuning: A baseline for network adaptation via additive side networks. In *European Conference on Computer Vision (ECCV)*, 2020.

Xiao Zhang and Ji Wu. Dissecting learning and forgetting in language model finetuning. In *International Conference on Learning Representations (ICLR)*, 2024.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2SQL: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

APPENDIX CONTENTS

A Implementation Details **17**

B Dataset Details **18**

C Splits **19**

D Fingerprint Sensitivity **19**

E Qualitative Recovery Examples **20**

F Additional Split Results **22**

F.1 Split A: Qwen2.5-7B 22

F.2 Split A: Llama-3.1-8B 23

F.3 Split B: Qwen2.5-7B 24

F.4 Split B: Llama-3.1-8B 25

F.5 Split C: Qwen2.5-7B 26

F.6 Split C: Llama-3.1-8B 27

G Split D Source-domain Results **28**

G.1 Qwen2.5-7B 28

G.2 Llama-3.1-8B 29

A Implementation Details

Table 3: Architecture, training, and inference settings of HyperRepair.

Parameter	Value
Repairing Side Net (RSN)	
Hidden dimension / layers	192 / 6
Attention heads / feedforward dimension	4 / 768
Backbone layers (Qwen / Llama)	5, 10, 15, 19, 24, 28 / 6, 11, 17, 22, 27, 32
Prompt gate	Mean pooling, MLP (hidden dimension 384), sigmoid
Ladder gate	Learned per layer; sigmoid temperature 0.1
Parameters (Qwen / Llama)	9.57M / 10.56M (approximately 0.13% of backbone)
Hypernetwork	
Encoder hidden dimension / layers	128 / 2
Attention heads / feedforward dimension	4 / 512
Local / global encoding dimension	128 / 256
Maximum matrix-update rank	8
Fingerprint	Layerwise activation change: direction and log magnitude
Fingerprint prompts	150 unlabeled general prompts; 10 per task
Shared template initialization	Mean of the source RSN parameters
Generated update initialization	Zero
Domain LoRA fine-tuning	
Rank / scaling parameter α	64 / 128
Target modules	Query, key, value, and output attention projections; gate, up, and down MLP projections
Epochs / effective batch size	2 / 16
Optimizer / learning rate	AdamW / 3×10^{-4}
Scheduler / weight decay	Linear decay / 0
Dropout / gradient clipping / seed	0.05 / 1 / 42
Ground-truth RSN training	
Training steps	1,500 per source model
Optimizer / learning rate	AdamW / 10^{-4}
Learning-rate schedule	200-step linear warmup, then constant
General examples per step	64 labeled examples from general datasets
Domain prompts per step	64 prompts without answer labels
Hypernetwork training	
Training steps	3,000 per backbone and split
Optimizer / learning rate	AdamW / 3×10^{-5} (constant)
General examples per step	64 labeled examples from general datasets
Domain prompts per step	64 source-domain prompts without answer labels
Loss weights $(\beta_{\text{func}}, \beta_{\text{weight}}, \beta_{\text{align}})$	$(1,\ 0.2,\ 0.1)$
Source / held-out domains	9 / 3 per split
Inference	
Repair generation	One HyperNet forward pass per model fingerprint
Total deployment time	$<$ 1 min on one NVIDIA A100 (40 GB), including fingerprint extraction
Prompt-gate threshold	Values below 0.1 are set to zero
Parameter reuse	The generated RSN is reused across requests

The source RSNs and HyperNet use the same general training data (Table 5). We evaluate their final training checkpoints.

Parameter generation. The fingerprint encoder adds projected direction and magnitude features to a learned layer embedding. Mean pooling of its layer encodings, followed by a nonlinear projection, gives the global encoding. Local parameter heads are shared across side layers; the input-embedding projection uses the first decoder layer's encoding together with the global encoding. Matrix heads predict low-rank factors, while vector and scalar heads predict their updates directly.

Initializing one factor of each matrix update and all vector and scalar updates to zero makes the initial generated RSN equal to the shared template. Activation means and template initialization use FP32, with $\epsilon = 10^{-8}$ in Equation 5.

Functional supervision and answer scoring. For discriminative tasks, candidate-answer scores sum continuation-token log-probabilities. For tasks using length-normalized accuracy, training divides each score by the candidate's character count and multiplies it by 20 before applying option cross-entropy. For generative tasks, training uses the mean cross-entropy over response tokens under teacher forcing, with prompt and formatting tokens masked. The general-task loss is averaged over general examples. The positive and negative gate losses are averaged separately over general and domain prompts, respectively; these three means are added with unit coefficients. Gate supervision uses logits for numerical stability, and training uses the unthresholded gate.

Reconstruction and representation alignment. We average the reconstruction MSE equally across parameter tensors and source domains. The error is normalized by its value at HyperNet initialization, computed with dropout disabled and kept fixed throughout training. Representation alignment compares the mean correction states over non-padding prompt tokens, before gating. It applies layer normalization without affine parameters ($\epsilon = 10^{-5}$) to each pooled vector and averages squared differences over hidden dimensions and an equal mixture of general and source-domain prompts. The ground-truth RSN remains frozen.

B Dataset Details

The domain datasets cover classification, question answering, semantic parsing, and code summarization (Table 4). Domain answers are used for fine-tuning the backbone; repair training requires domain prompts without answer labels. General capabilities are evaluated on the 15 benchmarks in Table 5, which reports the shared general training pools for source RSNs and HyperNet, along with the test set sizes used in this study. When a labeled official test set is unavailable, we use the official validation split for testing.

Evaluation. General evaluation follows the prompts, few-shot settings, filters, and metrics of `lm-evaluation-harness` v0.4.12 (Biderman et al., 2024), without a chat template. The average general score is the unweighted mean of the 15 task scores, computed before rounding. All scores are reported on a 0–100 scale.

Table 4: The 12 adaptation datasets.

Dataset	Task	Train	Test	Metric
HateXplain (Mathew et al., 2021)	Hate-speech detection	4,000	1,000	Macro-F1
SemEval-2010 T8 (Hendrickx et al., 2010)	Relation classification	4,000	2,717	Macro-F1
SCOTUS (Chalkidis et al., 2022)	Legal issue classification	5,000	1,400	Macro-F1
MedMCQA (Pal et al., 2022)	Medical QA	4,000	1,000	Accuracy
MASSIVE-en (FitzGerald et al., 2023; Bastianelli et al., 2020)	Intent recognition	4,000	1,000	Accuracy
WikiSQL (Zhong et al., 2017)	Text-to-SQL	4,000	1,000	Execution accuracy
RACE (Lai et al., 2017)	Reading comprehension	4,000	1,000	Accuracy
Banking77 (Casanueva et al., 2020)	Banking intent	4,000	1,000	Accuracy
CodeXGLUE-JS (Lu et al., 2021)	Code summarization	4,000	1,000	BLEU-4
LEDGAR-Top10 (Tuggener et al., 2020)	Contract classification	4,000	1,000	Macro-F1
SNIPS-7 (Coucke et al., 2018)	Assistant intent	4,000	695	Accuracy
DROP (Dua et al., 2019)	Discrete-reasoning QA	4,000	1,000	F1

Note. LEDGAR-Top10 selects ten classes from the 100-class, single-label LEDGAR version in LexGLUE (Chalkidis et al., 2022).

Table 5: The 15 general benchmarks.

Dataset	Category	Train	Test
Discriminative tasks			
ARC-Challenge (Clark et al., 2018)	Science QA	889	1,172
HellaSwag (Zellers et al., 2019)	Commonsense reasoning	1,970	10,042
MMLU (Hendrycks et al., 2021a)	Multitask knowledge	21,752	14,042
OpenBookQA (Mihaylov et al., 2018)	Science QA	1,970	500
PIQA (Bisk et al., 2020)	Physical reasoning	1,970	1,838
SciQ (Welbl et al., 2017)	Science QA	11,448	1,000
WinoGrande (Sakaguchi et al., 2020)	Commonsense reasoning	1,970	1,267
ANLI-R1 (Nie et al., 2020)	Natural language inference	1,970	1,000
COPA (Roemmele et al., 2011)	Causal reasoning	290	100
SocialIQA (Sap et al., 2019)	Social reasoning	1,970	1,954
LogiQA (Liu et al., 2020)	Logical reasoning	1,970	651
WiC (Pilehvar & Camacho-Collados, 2019)	Word sense disambiguation	1,970	638
Generative tasks			
MATH (Hendrycks et al., 2021b)	Mathematical reasoning	7,466	5,000
HumanEval (Chen et al., 2021)	Code generation	7,788	164
MBPP (Austin et al., 2021)	Code generation	6,004	500

Note. HumanEval uses a cleaned OpenCodeInstruct (Ahmad et al., 2025) proxy training pool; MBPP combines official training examples with OpenCodeInstruct augmentation.

C SPLITS

Each split holds out three adaptation domains and trains HyperNet on models from the remaining nine (Table 6). Both backbones use the same partition, and every domain is held out once. Split D defines the main comparison in Table 1.

Table 6: Held-out adaptation domains in each split.

Split	Domain 1	Domain 2	Domain 3
A	HateXplain	SemEval-2010 T8	SCOTUS
B	MedMCQA	MASSIVE-en	WikiSQL
C	RACE	Banking77	CodeXGLUE-JS
D	LEDGAR-Top10	SNIPS-7	DROP

D FINGERPRINT SENSITIVITY

The prompt-substitution experiment tests how the generated repair changes when its fingerprint is estimated from different general prompts. We construct three disjoint groups, $\mathcal{C}^{(0)}$, $\mathcal{C}^{(1)}$, and $\mathcal{C}^{(2)}$, denoted A, B, and C in Section 4.3. Each contains ten unlabeled prompts from each of the 15 general datasets, giving 150 prompts per group. The same groups are shared across adaptation domains and splits; all 450 prompts are excluded from HyperNet functional supervision.

HyperNet training and the primary evaluation use group A. At evaluation, we replace it with B or C and recompute the fingerprint, keeping the fine-tuned model and trained HyperNet fixed. These experiments use Qwen2.5-7B on the three held-out domains of Split D, evaluated with SGLang on the complete test sets used for that split.

E Qualitative Recovery Examples

Table 7: Recovery examples on 13 additional benchmarks, showing abbreviated prompts, selected answers, and verbatim response excerpts.

Base (correct)	Fine-tuned (incorrect)	HyperRepair (correct)
HellaSwag		*Adaptation: CodeXGLUE-JS (Split C)*
Prompt. Clean and jerk: A woman in a black shirt is bent over. She		
picks up a heavy weight.	is putting lotion onto her back.	picks up a heavy weight.
MMLU		*Adaptation: SCOTUS (Split A)*
Prompt. A wave transfers		
energy	amplitude	energy
OpenBookQA		*Adaptation: MASSIVE-en (Split B)*
Prompt. Which musical instrument is the same type as a guitar?		
cello	drum	cello
PIQA		*Adaptation: SemEval-2010 T8 (Split A)*
Prompt. To place a screw in a plastic lid, you can		
Use a drill	Use a toothbrush	Use a drill
SciQ		*Adaptation: HateXplain (Split A)*
Prompt. What is it called when ice changes to liquid water?		
melting	condensing	melting
WinoGrande		*Adaptation: Banking77 (Split C)*
Prompt. Adam loved dogs but Jason was afraid of them, so only _ petted the poodle.		
Adam	Jason	Adam
ANLI-R1		*Adaptation: DROP (Split D)*
Prompt. "Can't Help Thinking About Me" is a song written by David Bowie in 1965 and released as a single under the name David Bowie with The Lower Third. This was the first single released after he changed his name from David (also Davie) Jones to David Bowie. Question: David Bowie wrote the song can't help thinking about me in 1966. True, False, or Neither?		
False	True	False
COPA		*Adaptation: SNIPS-7 (Split D)*
Prompt. Everyone in the class turned to stare at the student. What was the cause?		
The student's phone rang.	The student took notes.	The student's phone rang.
SocialIQA		*Adaptation: HateXplain (Split A)*
Prompt. Jordan won a car and could not believe his luck. How would Jordan feel afterwards?		
thrilled	angry	thrilled
LogiQA		*Adaptation: MASSIVE-en (Split B)*
Prompt. Every politician has to please his voters. Although Ma Ying-jeou is an honest person, he is also a politician. No one can please his voters if he does not occasionally say something ambiguous. If the above statement is true, which of the following statements must be true?		
Ma Ying-jeou will say something ambiguous	The politicians who say ambiguously are not honest people.	Ma Ying-jeou will say something ambiguous

Table 7 (continued)

Base (correct)	Fine-tuned (incorrect)	HyperRepair (correct)
WiC		*Adaptation: SCOTUS (Split A)*
Prompt. Sentence 1: Burn a CD. Sentence 2: My eyes are burning. Question: Is the word 'Burn' used in the same way in the two sentences above?		
no	yes	no
HumanEval		*Adaptation: CodeXGLUE-JS (Split C)*
Prompt. def add(lst): Given a non-empty list of integers lst. add the even elements that are at odd indices.. Example: add([4, 2, 6, 7]) ==> 2		
`return sum([lst[i] for i in range(1, len(lst), 2) if lst[i] % 2 == 0])`	`return sum([lst[i] for i in range(len(lst)) if i % 2 == 0 and lst[i] % 2 == 0])`	`return sum(lst[i] for i in range(1, len(lst), 2) if lst[i] % 2 == 0)`
MBPP		*Adaptation: WikiSQL (Split B)*
Prompt. Write a function to add the given list to the given tuples. Provided test: assert add_lists([5, 6, 7], (9, 10)) == (9, 10, 5, 6, 7)		

`def add_lists(list1, tuple1):`
`  return tuple1 + tuple(list1)`
	`def add_lists(nums, tup):`
	`  return tup + nums`
		`def add_lists(test_list, test_tuple):`
		`  res = test_tuple + tuple(test_list)`
		`  return (res)`

F Additional Split Results

In the following result tables, each entry shows Fine-tuned → HyperRepair.

F.1 Split A: Qwen2.5-7B

Table 8: Qwen2.5-7B results on Split A.

Role	Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
Held-out	HateXplain	68.3→67.6	41.1→56.9	63.0→83.8	67.4→65.7	43.2→50.6	73.3→80.6
	SemEval-2010 T8	87.5→87.5	51.4→60.2	72.9→86.2	63.3→62.6	46.6→50.0	77.3→82.6
	SCOTUS	68.3→67.7	45.7→61.3	77.7→89.0	25.6→37.9	46.8→53.0	80.2→84.1
Source	MedMCQA	38.4→38.1	62.5→70.9	79.3→90.5	69.5→70.0	50.2→57.6	81.4→84.7
	MASSIVE-en	85.7→85.7	56.7→69.4	78.1→90.0	68.9→69.5	47.8→57.2	79.7→84.8
	WikiSQL	86.5→86.3	55.9→65.8	77.3→90.2	66.6→68.0	44.4→52.6	79.4→84.7
	RACE	59.5→59.6	60.4→70.1	79.5→90.1	68.9→70.3	48.0→56.6	79.9→85.2
	Banking77	92.7→92.7	53.4→64.4	76.7→89.3	27.3→49.5	45.4→56.8	77.4→83.0
	CodeXGLUE-JS	15.8→15.9	55.7→69.2	79.2→90.8	71.3→71.5	45.6→53.2	78.8→84.9
	LEDGAR-Top10	98.0→98.2	53.8→66.9	73.1→89.8	55.0→64.6	46.0→56.0	79.4→84.4
	SNIPS-7	98.3→98.3	44.8→63.3	70.3→88.1	61.8→64.9	41.8→52.8	75.1→82.8
	DROP	76.2→75.9	59.9→71.0	76.1→89.3	64.8→66.1	46.8→55.0	79.8→84.8

Role	Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
Held-out	HateXplain	77.6→83.3	65.6→76.2	40.5→46.8	40.5→42.3	31.6→32.9
	SemEval-2010 T8	80.9→85.8	69.4→78.5	50.6→51.9	45.9→47.5	33.8→32.0
	SCOTUS	85.0→88.6	73.6→78.5	40.4→43.7	47.6→49.3	31.8→33.6
Source	MedMCQA	95.7→97.6	72.0→79.2	53.1→59.7	59.3→60.3	35.5→46.7
	MASSIVE-en	96.0→96.9	74.2→79.0	44.7→54.6	50.0→53.2	34.9→42.2
	WikiSQL	96.9→97.7	73.0→80.7	51.9→60.9	53.2→57.1	34.1→38.2
	RACE	96.5→97.4	69.1→78.5	50.8→65.5	57.5→60.3	48.2→50.7
	Banking77	95.0→96.2	72.6→80.7	35.0→45.9	47.2→48.4	33.2→34.6
	CodeXGLUE-JS	96.5→97.3	70.7→79.2	52.4→61.0	54.7→57.2	39.3→45.9
	LEDGAR-Top10	88.1→94.5	70.8→78.2	48.5→53.6	49.0→52.0	33.6→33.8
	SNIPS-7	87.3→93.8	68.7→78.7	37.1→46.9	38.0→42.5	31.0→36.7
	DROP	97.3→97.1	75.1→81.1	47.0→61.2	54.9→59.5	44.9→43.3

Role	Domain	WiC	COPA	MATH	HumanEval	MBPP
Held-out	HateXplain	50.0→53.8	75→92	0.0→37.2	0.0→41.5	0.0→50.6
	SemEval-2010 T8	49.7→50.8	85→90	0.0→41.6	0.0→43.9	6.4→52.0
	SCOTUS	50.3→58.5	89→92	0.0→43.9	22.6→32.9	36.0→29.0
Source	MedMCQA	55.8→68.0	90→92	40.2→43.8	56.1→50.0	62.8→58.4
	MASSIVE-en	64.9→64.7	91→95	43.4→45.6	46.3→43.3	42.0→46.0
	WikiSQL	60.5→63.3	91→95	39.7→41.3	52.4→47.0	59.4→55.2
	RACE	51.6→63.8	84→92	34.7→43.4	54.3→47.6	62.4→60.4
	Banking77	56.6→58.2	90→91	15.0→40.1	43.3→42.1	47.6→41.0
	CodeXGLUE-JS	56.0→67.9	89→91	22.0→45.0	53.7→51.2	60.4→58.2
	LEDGAR-Top10	50.0→58.6	88→93	3.6→38.8	22.0→33.5	28.8→42.2
	SNIPS-7	50.3→54.7	85→93	0.0→35.4	1.8→34.1	0.0→28.2
	DROP	65.8→65.4	90→96	33.4→39.7	51.8→47.0	58.8→56.6

F.2 SPLIT A: LLAMA-3.1-8B

Table 9: Llama-3.1-8B results on Split A.

Role	Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
Held-out	HateXplain	65.6→63.8	35.5→43.8	51.6→65.4	36.8→34.6	39.0→45.2	67.5→75.4
	SemEval-2010 T8	89.0→88.9	42.7→50.3	73.9→80.6	51.6→51.4	43.8→49.6	78.1→80.9
	SCOTUS	68.5→68.5	47.7→52.8	78.1→82.5	35.4→41.6	45.6→49.2	79.4→82.0
Source	MedMCQA	40.0→39.9	56.5→60.8	79.8→84.9	45.3→49.6	49.0→52.8	81.5→82.1
	MASSIVE-en	84.2→84.2	46.7→55.1	75.0→82.7	41.1→37.2	42.8→51.0	77.4→81.9
	WikiSQL	85.6→85.5	49.5→54.2	76.6→83.6	52.0→53.9	42.4→49.0	79.4→81.7
	RACE	59.6→59.4	58.8→62.4	78.9→83.9	47.6→48.3	47.0→52.0	81.5→82.6
	Banking77	92.9→92.9	49.2→54.0	77.0→82.3	35.0→34.2	47.0→50.4	78.0→81.3
	CodeXGLUE-JS	15.6→15.6	53.4→57.8	78.8→85.0	62.7→61.5	46.0→52.2	80.6→82.1
	LEDGAR-Top10	98.0→98.0	40.1→52.5	62.6→80.0	28.1→29.9	42.8→50.2	74.0→80.8
	SNIPS-7	98.7→98.7	38.1→49.7	68.3→78.1	23.5→25.8	41.0→47.6	75.9→80.4
	DROP	65.4→65.3	55.2→59.6	76.2→82.1	33.2→36.7	47.2→52.0	79.8→82.1

Role	Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
Held-out	HateXplain	64.4→73.0	66.7→73.0	33.3→33.3	36.6→38.2	26.3→24.3
	SemEval-2010 T8	72.0→78.7	71.6→78.2	34.9→40.2	45.6→45.9	29.5→26.6
	SCOTUS	80.1→87.0	72.2→77.9	34.2→40.9	47.1→46.9	32.0→30.4
Source	MedMCQA	95.7→96.4	73.5→79.2	34.4→44.1	52.5→53.6	29.3→28.6
	MASSIVE-en	84.4→93.9	74.0→78.4	32.3→46.1	42.2→44.7	30.7→28.6
	WikiSQL	96.0→96.5	73.2→80.2	32.6→38.6	48.5→50.8	30.9→30.7
	RACE	96.3→97.2	71.9→78.9	35.0→43.1	52.0→54.3	28.0→28.3
	Banking77	95.5→94.5	73.6→78.5	37.3→47.0	43.9→46.5	29.6→28.4
	CodeXGLUE-JS	95.6→97.2	73.2→79.0	36.8→44.3	49.6→50.8	31.6→33.0
	LEDGAR-Top10	69.0→91.9	71.0→77.4	36.2→42.5	40.7→44.0	31.2→28.4
	SNIPS-7	76.7→89.5	70.7→76.6	33.2→39.2	40.3→44.5	29.3→29.2
	DROP	96.3→96.9	74.5→80.0	32.1→40.7	51.5→52.6	27.5→26.9

Role	Domain	WiC	COPA	MATH	HumanEval	MBPP
Held-out	HateXplain	50.0→50.0	71→88	0.0→1.2	0.0→15.9	0.0→9.4
	SemEval-2010 T8	50.0→58.5	89→86	0.0→7.2	0.6→25.6	0.0→29.2
	SCOTUS	52.2→50.2	94→93	0.0→9.3	28.0→25.6	35.2→29.8
Source	MedMCQA	59.9→60.2	90→93	12.2→14.8	34.8→30.5	43.2→34.0
	MASSIVE-en	50.5→54.7	90→91	0.0→9.7	0.6→21.3	0.0→13.0
	WikiSQL	50.9→55.3	87→92	6.1→9.7	20.1→22.6	28.6→27.6
	RACE	50.2→58.5	91→96	3.3→7.7	34.1→30.5	41.6→37.6
	Banking77	49.8→55.3	91→92	0.0→10.0	25.6→26.2	29.0→27.8
	CodeXGLUE-JS	58.3→63.9	93→94	5.9→14.8	33.5→31.1	42.2→35.6
	LEDGAR-Top10	50.0→56.4	86→94	0.0→4.1	0.0→12.2	0.0→11.2
	SNIPS-7	49.8→54.1	88→90	0.0→3.9	12.2→14.6	1.2→10.4
	DROP	53.1→60.5	92→94	0.0→3.4	13.4→22.0	25.4→33.8

F.3 SPLIT B: QWEN2.5-7B

Table 10: Qwen2.5-7B results on Split B.

Role	Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
Held-out	MedMCQA	38.4→37.2	62.5→72.0	79.3→89.8	69.5→70.3	50.2→55.4	81.4→84.0
	MASSIVE-en	85.7→85.7	56.7→69.5	78.1→88.7	68.9→69.4	47.8→55.2	79.7→83.9
	WikiSQL	86.5→86.1	55.9→65.7	77.3→88.9	66.6→65.4	44.4→50.6	79.4→83.6
Source	HateXplain	68.3→68.6	41.1→64.4	63.0→89.2	67.4→69.7	43.2→52.4	73.3→84.0
	SemEval-2010 T8	87.5→87.5	51.4→67.8	72.9→87.8	63.3→65.0	46.6→55.2	77.3→83.1
	SCOTUS	68.3→68.4	45.7→65.0	77.7→89.5	25.6→62.8	46.8→54.0	80.2→83.9
	RACE	59.5→59.6	60.4→70.3	79.5→89.6	68.9→71.2	48.0→55.6	79.9→84.3
	Banking77	92.7→92.7	53.4→66.4	76.7→89.2	27.3→57.1	45.4→56.8	77.4→83.2
	CodeXGLUE-JS	15.8→15.9	55.7→68.7	79.2→89.8	71.3→71.2	45.6→53.0	78.8→84.1
	LEDGAR-Top10	98.0→98.2	53.8→67.9	73.1→89.2	55.0→65.6	46.0→55.0	79.4→83.7
	SNIPS-7	98.3→98.3	44.8→63.1	70.3→87.6	61.8→66.8	41.8→52.8	75.1→81.7
	DROP	76.2→76.1	59.9→71.3	76.1→88.2	64.8→66.5	46.8→56.4	79.8→83.6

Role	Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
Held-out	MedMCQA	95.7→97.5	72.0→78.5	53.1→62.8	59.3→60.3	35.5→44.4
	MASSIVE-en	96.0→96.7	74.2→78.9	44.7→50.6	50.0→52.1	34.9→43.2
	WikiSQL	96.9→97.7	73.0→80.9	51.9→56.9	53.2→56.4	34.1→36.7
Source	HateXplain	77.6→95.0	65.6→77.4	40.5→55.5	40.5→48.3	31.6→39.9
	SemEval-2010 T8	80.9→95.0	69.4→80.0	50.6→58.2	45.9→52.4	33.8→35.6
	SCOTUS	85.0→92.6	73.6→79.2	40.4→45.9	47.6→50.7	31.8→33.3
	RACE	96.5→97.6	69.1→78.5	50.8→63.8	57.5→61.0	48.2→48.8
	Banking77	95.0→96.3	72.6→80.6	35.0→48.6	47.2→48.4	33.2→36.4
	CodeXGLUE-JS	96.5→97.2	70.7→79.6	52.4→60.7	54.7→58.2	39.3→46.9
	LEDGAR-Top10	88.1→96.0	70.8→78.4	48.5→52.9	49.0→53.5	33.6→34.4
	SNIPS-7	87.3→94.4	68.7→78.4	37.1→46.9	38.0→42.5	31.0→36.9
	DROP	97.3→97.4	75.1→81.2	47.0→62.7	54.9→58.7	44.9→45.3

Role	Domain	WiC	COPA	MATH	HumanEval	MBPP
Held-out	MedMCQA	55.8→67.9	90→95	40.2→44.4	56.1→52.4	62.8→57.8
	MASSIVE-en	64.9→52.7	91→95	43.4→45.3	46.3→50.0	42.0→48.6
	WikiSQL	60.5→62.9	91→94	39.7→41.0	52.4→43.3	59.4→56.4
Source	HateXplain	50.0→56.7	75→93	0.0→40.4	0.0→33.5	0.0→45.8
	SemEval-2010 T8	49.7→58.5	85→93	0.0→41.6	0.0→45.1	6.4→52.4
	SCOTUS	50.3→62.4	89→93	0.0→43.4	22.6→31.7	36.0→27.4
	RACE	51.6→68.7	84→93	34.7→43.3	54.3→46.3	62.4→62.6
	Banking77	56.6→59.1	90→94	15.0→40.1	43.3→44.5	47.6→42.0
	CodeXGLUE-JS	56.0→54.5	89→93	22.0→43.7	53.7→50.0	60.4→58.4
	LEDGAR-Top10	50.0→63.6	88→94	3.6→38.9	22.0→36.6	28.8→42.8
	SNIPS-7	50.3→54.5	85→94	0.0→35.5	1.8→31.7	0.0→31.4
	DROP	65.8→66.1	90→94	33.4→38.9	51.8→46.3	58.8→55.2

F.4 SPLIT B: LLAMA-3.1-8B

Table 11: Llama-3.1-8B results on Split B.

Role	Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
Held-out	MedMCQA	40.0→39.4	56.5→61.2	79.8→86.6	45.3→38.1	49.0→54.6	81.5→82.8
	MASSIVE-en	84.2→84.0	46.7→55.8	75.0→83.7	41.1→40.8	42.8→52.4	77.4→81.3
	WikiSQL	85.6→84.9	49.5→55.9	76.6→85.4	52.0→50.6	42.4→52.0	79.4→81.8
Source	HateXplain	65.6→65.8	35.5→54.5	51.6→82.8	36.8→51.9	39.0→50.4	67.5→81.7
	SemEval-2010 T8	89.0→89.0	42.7→57.6	73.9→85.4	51.6→50.7	43.8→50.6	78.1→82.4
	SCOTUS	68.5→68.6	47.7→59.9	78.1→85.9	35.4→46.2	45.6→54.0	79.4→83.4
	RACE	59.6→59.4	58.8→64.5	78.9→86.1	47.6→50.8	47.0→54.4	81.5→83.4
	Banking77	92.9→92.9	49.2→56.8	77.0→84.0	35.0→38.0	47.0→52.2	78.0→81.2
	CodeXGLUE-JS	15.6→15.6	53.4→59.6	78.8→86.6	62.7→62.2	46.0→54.0	80.6→82.8
	LEDGAR-Top10	98.0→98.0	40.1→54.9	62.6→83.1	28.1→31.5	42.8→51.6	74.0→81.5
	SNIPS-7	98.7→98.7	38.1→53.0	68.3→81.5	23.5→28.3	41.0→51.4	75.9→81.0
	DROP	65.4→65.3	55.2→61.9	76.2→85.0	33.2→39.5	47.2→56.4	79.8→82.6

Role	Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
Held-out	MedMCQA	95.7→96.9	73.5→78.5	34.4→37.7	52.5→53.8	29.3→29.3
	MASSIVE-en	84.4→89.5	74.0→78.0	32.3→46.4	42.2→43.8	30.7→30.1
	WikiSQL	96.0→97.0	73.2→78.9	32.6→36.4	48.5→51.7	30.9→30.6
Source	HateXplain	64.4→90.5	66.7→76.6	33.3→41.5	36.6→41.3	26.3→28.3
	SemEval-2010 T8	72.0→94.9	71.6→78.2	34.9→46.6	45.6→50.0	29.5→30.4
	SCOTUS	80.1→92.9	72.2→77.4	34.2→44.8	47.1→49.6	32.0→30.6
	RACE	96.3→97.7	71.9→79.3	35.0→47.2	52.0→55.6	28.0→26.9
	Banking77	95.5→95.0	73.6→78.5	37.3→47.8	43.9→47.3	29.6→28.7
	CodeXGLUE-JS	95.6→97.4	73.2→77.8	36.8→43.6	49.6→52.8	31.6→32.1
	LEDGAR-Top10	69.0→93.3	71.0→78.2	36.2→45.0	40.7→45.5	31.2→29.5
	SNIPS-7	76.7→90.5	70.7→76.9	33.2→40.9	40.3→44.7	29.3→28.7
	DROP	96.3→97.2	74.5→80.1	32.1→43.1	51.5→54.0	27.5→28.0

Role	Domain	WiC	COPA	MATH	HumanEval	MBPP
Held-out	MedMCQA	59.9→58.6	90→95	12.2→14.7	34.8→29.9	43.2→35.0
	MASSIVE-en	50.5→50.0	90→92	0.0→9.8	0.6→20.1	0.0→16.8
	WikiSQL	50.9→50.0	87→94	6.1→10.1	20.1→19.5	28.6→24.2
Source	HateXplain	50.0→50.2	71→93	0.0→2.6	0.0→14.6	0.0→5.0
	SemEval-2010 T8	50.0→50.9	89→93	0.0→6.8	0.6→26.8	0.0→29.0
	SCOTUS	52.2→53.3	94→94	0.0→10.6	28.0→25.6	35.2→26.8
	RACE	50.2→62.7	91→95	3.3→8.1	34.1→35.4	41.6→35.6
	Banking77	49.8→55.6	91→92	0.0→10.5	25.6→23.8	29.0→29.0
	CodeXGLUE-JS	58.3→50.5	93→95	5.9→14.8	33.5→29.9	42.2→35.2
	LEDGAR-Top10	50.0→50.9	86→93	0.0→6.7	0.0→12.8	0.0→11.8
	SNIPS-7	49.8→52.0	88→93	0.0→4.2	12.2→15.9	1.2→12.4
	DROP	53.1→58.2	92→94	0.0→3.2	13.4→24.4	25.4→34.0

F.5 SPLIT C: QWEN2.5-7B

Table 12: Qwen2.5-7B results on Split C.

Role	Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
Held-out	RACE	59.5→64.0	60.4→70.4	79.5→90.2	68.9→70.6	48.0→57.4	79.9→83.9
	Banking77	92.7→92.7	53.4→64.2	76.7→89.4	27.3→34.6	45.4→54.2	77.4→82.5
	CodeXGLUE-JS	15.8→15.7	55.7→68.1	79.2→90.4	71.3→71.6	45.6→52.2	78.8→84.3
Source	HateXplain	68.3→68.6	41.1→64.9	63.0→89.6	67.4→69.1	43.2→51.4	73.3→83.5
	SemEval-2010 T8	87.5→87.5	51.4→68.0	72.9→89.5	63.3→65.1	46.6→56.0	77.3→84.4
	SCOTUS	68.3→68.4	45.7→65.7	77.7→89.9	25.6→62.5	46.8→54.0	80.2→83.7
	MedMCQA	38.4→38.1	62.5→73.5	79.3→90.4	69.5→70.3	50.2→56.8	81.4→84.7
	MASSIVE-en	85.7→85.7	56.7→70.5	78.1→90.3	68.9→69.3	47.8→57.2	79.7→85.3
	WikiSQL	86.5→86.3	55.9→66.4	77.3→90.6	66.6→67.5	44.4→50.8	79.4→83.8
	LEDGAR-Top10	98.0→98.2	53.8→68.1	73.1→89.3	55.0→62.1	46.0→53.6	79.4→85.1
	SNIPS-7	98.3→98.3	44.8→64.7	70.3→89.5	61.8→66.0	41.8→52.4	75.1→83.2
	DROP	76.2→76.1	59.9→70.9	76.1→89.9	64.8→66.4	46.8→55.6	79.8→85.1

Role	Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
Held-out	RACE	96.5→97.8	69.1→77.7	50.8→63.3	57.5→60.8	48.2→46.4
	Banking77	95.0→95.2	72.6→79.6	35.0→45.5	47.2→49.0	33.2→37.3
	CodeXGLUE-JS	96.5→97.3	70.7→79.2	52.4→50.3	54.7→58.0	39.3→43.2
Source	HateXplain	77.6→94.3	65.6→78.7	40.5→56.4	40.5→48.9	31.6→39.0
	SemEval-2010 T8	80.9→95.0	69.4→79.7	50.6→56.8	45.9→53.2	33.8→34.7
	SCOTUS	85.0→93.2	73.6→80.4	40.4→44.6	47.6→50.8	31.8→32.6
	MedMCQA	95.7→97.6	72.0→79.0	53.1→62.0	59.3→61.6	35.5→39.9
	MASSIVE-en	96.0→97.2	74.2→80.0	44.7→52.3	50.0→54.4	34.9→40.1
	WikiSQL	96.9→97.6	73.0→81.4	51.9→61.9	53.2→57.8	34.1→40.2
	LEDGAR-Top10	88.1→96.1	70.8→78.3	48.5→54.4	49.0→53.9	33.6→33.5
	SNIPS-7	87.3→94.6	68.7→79.2	37.1→46.9	38.0→44.7	31.0→38.4
	DROP	97.3→97.6	75.1→82.1	47.0→64.5	54.9→59.9	44.9→45.2

Role	Domain	WiC	COPA	MATH	HumanEval	MBPP
Held-out	RACE	51.6→64.4	84→92	34.7→44.2	54.3→50.6	62.4→60.4
	Banking77	56.6→56.9	90→92	15.0→38.9	43.3→43.3	47.6→45.2
	CodeXGLUE-JS	56.0→52.8	89→92	22.0→45.4	53.7→50.6	60.4→57.8
Source	HateXplain	50.0→57.4	75→93	0.0→41.0	0.0→34.1	0.0→46.6
	SemEval-2010 T8	49.7→59.1	85→95	0.0→41.0	0.0→47.6	6.4→52.2
	SCOTUS	50.3→62.9	89→94	0.0→43.1	22.6→29.9	36.0→29.2
	MedMCQA	55.8→67.4	90→94	40.2→44.6	56.1→49.4	62.8→61.0
	MASSIVE-en	64.9→60.7	91→96	43.4→45.6	46.3→43.9	42.0→46.4
	WikiSQL	60.5→53.1	91→94	39.7→40.8	52.4→47.0	59.4→57.4
	LEDGAR-Top10	50.0→63.3	88→93	3.6→38.3	22.0→39.6	28.8→44.2
	SNIPS-7	50.3→55.3	85→95	0.0→35.2	1.8→36.0	0.0→30.4
	DROP	65.8→65.0	90→97	33.4→39.3	51.8→48.2	58.8→58.0

F.6 SPLIT C: LLAMA-3.1-8B

Table 13: Llama-3.1-8B results on Split C.

Role	Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
Held-out	RACE	59.6→59.6	58.8→60.1	78.9→84.4	47.6→48.3	47.0→51.4	81.5→81.4
	Banking77	92.9→92.9	49.2→54.9	77.0→84.0	35.0→37.3	47.0→53.0	78.0→81.0
	CodeXGLUE-JS	15.6→15.6	53.4→59.4	78.8→86.2	62.7→62.3	46.0→52.0	80.6→81.8
Source	HateXplain	65.6→65.8	35.5→54.5	51.6→83.5	36.8→54.1	39.0→52.0	67.5→80.7
	SemEval-2010 T8	89.0→89.0	42.7→56.7	73.9→86.3	51.6→53.5	43.8→51.2	78.1→82.9
	SCOTUS	68.5→68.6	47.7→59.5	78.1→86.8	35.4→50.1	45.6→51.6	79.4→82.5
	MedMCQA	40.0→39.9	56.5→61.9	79.8→86.6	45.3→54.2	49.0→53.4	81.5→82.6
	MASSIVE-en	84.2→84.2	46.7→57.4	75.0→85.4	41.1→45.7	42.8→51.6	77.4→81.0
	WikiSQL	85.6→85.5	49.5→55.8	76.6→86.3	52.0→54.9	42.4→51.0	79.4→81.5
	LEDGAR-Top10	98.0→98.0	40.1→55.0	62.6→83.4	28.1→32.8	42.8→51.8	74.0→81.1
	SNIPS-7	98.7→98.7	38.1→51.9	68.3→82.0	23.5→30.0	41.0→50.0	75.9→79.6
	DROP	65.4→65.3	55.2→60.0	76.2→84.8	33.2→40.2	47.2→54.2	79.8→82.4

Role	Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
Held-out	RACE	96.3→97.5	71.9→78.2	35.0→34.7	52.0→55.0	28.0→28.7
	Banking77	95.5→95.8	73.6→79.2	37.3→39.1	43.9→45.9	29.6→29.2
	CodeXGLUE-JS	95.6→97.2	73.2→78.1	36.8→43.9	49.6→51.9	31.6→32.0
Source	HateXplain	64.4→91.7	66.7→76.7	33.3→40.8	36.6→40.9	26.3→29.0
	SemEval-2010 T8	72.0→93.7	71.6→78.3	34.9→43.2	45.6→49.9	29.5→30.7
	SCOTUS	80.1→93.5	72.2→78.0	34.2→43.6	47.1→50.0	32.0→30.4
	MedMCQA	95.7→96.7	73.5→79.0	34.4→41.9	52.5→55.2	29.3→29.3
	MASSIVE-en	84.4→95.0	74.0→78.0	32.3→48.0	42.2→44.1	30.7→28.4
	WikiSQL	96.0→97.1	73.2→79.7	32.6→40.0	48.5→51.4	30.9→30.4
	LEDGAR-Top10	69.0→93.6	71.0→78.1	36.2→46.0	40.7→45.0	31.2→29.0
	SNIPS-7	76.7→91.2	70.7→76.6	33.2→40.3	40.3→44.8	29.3→29.0
	DROP	96.3→97.3	74.5→79.8	32.1→39.9	51.5→53.8	27.5→29.5

Role	Domain	WiC	COPA	MATH	HumanEval	MBPP
Held-out	RACE	50.2→50.8	91→96	3.3→7.1	34.1→34.1	41.6→38.0
	Banking77	49.8→54.2	91→93	0.0→10.8	25.6→25.0	29.0→31.6
	CodeXGLUE-JS	58.3→58.6	93→94	5.9→14.2	33.5→29.9	42.2→36.8
Source	HateXplain	50.0→56.1	71→92	0.0→1.9	0.0→13.4	0.0→6.2
	SemEval-2010 T8	50.0→56.7	89→91	0.0→6.9	0.6→25.0	0.0→26.2
	SCOTUS	52.2→55.2	94→94	0.0→10.7	28.0→28.0	35.2→27.4
	MedMCQA	59.9→61.6	90→94	12.2→14.9	34.8→30.5	43.2→36.0
	MASSIVE-en	50.5→53.4	90→94	0.0→9.6	0.6→17.7	0.0→14.6
	WikiSQL	50.9→50.9	87→94	6.1→10.0	20.1→21.3	28.6→26.0
	LEDGAR-Top10	50.0→58.2	86→95	0.0→6.9	0.0→12.8	0.0→13.4
	SNIPS-7	49.8→54.1	88→93	0.0→4.2	12.2→17.1	1.2→11.0
	DROP	53.1→59.4	92→94	0.0→2.5	13.4→23.2	25.4→34.4

G SPLIT D SOURCE-DOMAIN RESULTS

Source models provide functional supervision, RSN parameter targets, and template initialization.

G.1 QWEN2.5-7B

Table 14: Qwen2.5-7B results on the source domains of Split D.

Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
HateXplain	68.3→68.6	41.1→64.3	63.0→89.2	67.4→69.0	43.2→55.2	73.3→83.2
SemEval-2010 T8	87.5→87.5	51.4→66.8	72.9→88.3	63.3→62.4	46.6→53.4	77.3→83.5
SCOTUS	68.3→68.4	45.7→64.1	77.7→89.6	25.6→63.9	46.8→53.4	80.2→83.0
MedMCQA	38.4→38.1	62.5→72.9	79.3→90.1	69.5→70.3	50.2→56.2	81.4→84.3
MASSIVE-en	85.7→85.7	56.7→70.5	78.1→89.6	68.9→69.2	47.8→57.0	79.7→85.3
WikiSQL	86.5→86.3	55.9→66.0	77.3→90.0	66.6→67.0	44.4→51.4	79.4→83.8
RACE	59.5→59.6	60.4→70.3	79.5→90.1	68.9→70.3	48.0→55.8	79.9→85.0
Banking77	92.7→92.7	53.4→65.1	76.7→89.0	27.3→57.2	45.4→55.8	77.4→83.5
CodeXGLUE-JS	15.8→15.9	55.7→69.6	79.2→90.7	71.3→71.4	45.6→53.4	78.8→84.9

Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
HateXplain	77.6→94.6	65.6→78.1	40.5→54.2	40.5→47.6	31.6→40.4
SemEval-2010 T8	80.9→95.5	69.4→79.3	50.6→58.3	45.9→51.9	33.8→35.0
SCOTUS	85.0→90.9	73.6→79.6	40.4→44.3	47.6→49.9	31.8→34.6
MedMCQA	95.7→97.5	72.0→79.1	53.1→61.9	59.3→60.2	35.5→43.9
MASSIVE-en	96.0→97.1	74.2→79.6	44.7→54.3	50.0→52.8	34.9→44.1
WikiSQL	96.9→97.6	73.0→81.1	51.9→60.6	53.2→57.9	34.1→38.9
RACE	96.5→97.6	69.1→79.3	50.8→65.5	57.5→60.9	48.2→48.5
Banking77	95.0→95.5	72.6→80.7	35.0→46.7	47.2→49.3	33.2→37.2
CodeXGLUE-JS	96.5→97.3	70.7→79.8	52.4→60.6	54.7→57.7	39.3→45.8

Domain	WiC	COPA	MATH	HumanEval	MBPP
HateXplain	50.0→55.8	75→93	0.0→41.3	0.0→32.3	0.0→45.4
SemEval-2010 T8	49.7→60.2	85→94	0.0→40.8	0.0→45.7	6.4→52.2
SCOTUS	50.3→59.6	89→94	0.0→43.4	22.6→28.7	36.0→27.0
MedMCQA	55.8→68.2	90→92	40.2→44.4	56.1→49.4	62.8→59.2
MASSIVE-en	64.9→61.0	91→94	43.4→45.6	46.3→37.8	42.0→46.6
WikiSQL	60.5→62.9	91→92	39.7→40.9	52.4→51.2	59.4→58.8
RACE	51.6→65.8	84→92	34.7→43.4	54.3→49.4	62.4→61.4
Banking77	56.6→55.8	90→93	15.0→39.0	43.3→40.9	47.6→41.6
CodeXGLUE-JS	56.0→64.6	89→91	22.0→45.9	53.7→50.0	60.4→57.6

G.2 LLAMA-3.1-8B

Table 15: Llama-3.1-8B results on the source domains of Split D.

Domain	Domain score	ARC-C	HellaSwag	MMLU	OpenBookQA	PIQA
HateXplain	65.6→65.8	35.5→52.7	51.6→80.3	36.8→51.2	39.0→48.4	67.5→80.8
SemEval-2010 T8	89.0→89.0	42.7→56.7	73.9→84.3	51.6→48.7	43.8→51.6	78.1→83.4
SCOTUS	68.5→68.6	47.7→58.9	78.1→84.8	35.4→46.7	45.6→51.0	79.4→83.4
MedMCQA	40.0→39.9	56.5→62.7	79.8→85.3	45.3→51.9	49.0→54.0	81.5→83.2
MASSIVE-en	84.2→84.2	46.7→56.5	75.0→83.1	41.1→41.3	42.8→51.4	77.4→81.7
WikiSQL	85.6→85.5	49.5→55.4	76.6→84.2	52.0→54.0	42.4→50.2	79.4→82.3
RACE	59.6→59.4	58.8→64.1	78.9→85.1	47.6→50.0	47.0→54.0	81.5→83.4
Banking77	92.9→92.9	49.2→55.0	77.0→82.7	35.0→37.9	47.0→52.6	78.0→80.7
CodeXGLUE-JS	15.6→15.6	53.4→61.2	78.8→85.9	62.7→61.2	46.0→53.0	80.6→83.2

Domain	SciQ	WinoGrande	ANLI-R1	SocialIQA	LogiQA
HateXplain	64.4→90.7	66.7→76.8	33.3→38.3	36.6→40.4	26.3→28.7
SemEval-2010 T8	72.0→94.3	71.6→78.1	34.9→46.2	45.6→49.2	29.5→33.2
SCOTUS	80.1→92.7	72.2→78.5	34.2→44.2	47.1→48.0	32.0→30.1
MedMCQA	95.7→97.1	73.5→79.4	34.4→45.3	52.5→53.4	29.3→27.5
MASSIVE-en	84.4→94.2	74.0→78.4	32.3→44.7	42.2→44.6	30.7→28.9
WikiSQL	96.0→96.7	73.2→79.3	32.6→39.9	48.5→50.1	30.9→31.0
RACE	96.3→97.5	71.9→79.2	35.0→43.8	52.0→54.6	28.0→28.9
Banking77	95.5→95.8	73.6→78.9	37.3→46.1	43.9→46.5	29.6→29.5
CodeXGLUE-JS	95.6→97.8	73.2→78.9	36.8→43.8	49.6→50.4	31.6→33.0

Domain	WiC	COPA	MATH	HumanEval	MBPP
HateXplain	50.0→55.5	71→92	0.0→1.1	0.0→14.6	0.0→4.4
SemEval-2010 T8	50.0→59.7	89→93	0.0→6.6	0.6→22.6	0.0→24.4
SCOTUS	52.2→54.9	94→95	0.0→8.8	28.0→29.3	35.2→25.2
MedMCQA	59.9→60.8	90→95	12.2→14.4	34.8→31.7	43.2→35.0
MASSIVE-en	50.5→56.6	90→92	0.0→8.7	0.6→18.9	0.0→14.4
WikiSQL	50.9→54.7	87→90	6.1→9.7	20.1→22.0	28.6→25.0
RACE	50.2→58.0	91→97	3.3→6.3	34.1→32.3	41.6→38.4
Banking77	49.8→52.5	91→92	0.0→9.5	25.6→26.8	29.0→30.4
CodeXGLUE-JS	58.3→61.3	93→93	5.9→15.5	33.5→28.7	42.2→35.6