PROBING UNDER DISTRIBUTION SHIFT: LEARNING GENERALIZABLE DISTRIBUTED REPRESENTATIONS OF LLM ACTIVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Probing is one of the most promising approaches for monitoring the internal states of large language models: probes read a subset of model activations and predict an external variable of interest. Yet recent work has documented poor out-of-distribution generalization, overfitting. These are standard supervised-learning generalization problems, of which probing is a special case, with the additional challenge that its inputs come from the high-dimensional and highly structured activation space of a pretrained model. Therefore, we propose to focus on generalization and evaluate probes under distribution shift using **leave-one-environment-out evaluation**. This perspective suggests two natural independent methodological directions: distributionally robust learning that explicitly targets generalization across environments, and learning distributed representations of the activation space that best support generalizable predictions. We systematically compare baseline probes with distributed representations trained with and without distributionally robust objectives. Across three tasks and 15 environments, our proposed distributed representations consistently improve OOD generalization despite greater complexity, while none of the existing distributionally robust methods we evaluate provides additional gains. Our proposed distributed representation compresses the full $[L \times N \times d]$ activation tensor via two-stage inducing-point attention with a sparse layer selector, meaning that once it is learned the probe remain resource-efficient. Our results suggest that regularization is better achieved by training across diverse environments than by simply constraining probe complexity. Finally, we discuss implication for future research.

1 INTRODUCTION

Robust monitoring of language models requires inferring not only what they will output, but also what internal state they are in while producing it. For reasoning models, the most direct approach is to monitor their chain-of-thought (CoT) (Baker et al., 2025; Korbak et al., 2025; Guan et al., 2026). However, CoTs are not always faithful descriptions of the model's internal computation or state (Turpin et al., 2023; Lanham et al., 2023; Chen et al., 2025). This is especially problematic when the target is a *model-relative internal state*, such as uncertainty, hallucination risk, evaluation awareness, or whether an input was encountered during pre-training. The same input can induce different states for different models or interaction histories, and direct verbalization of these states remains unreliable (Kumar et al., 2024; Groot & Valdenegro-Toro, 2024; Chen et al., 2025; Lindsey, 2026). A principled alternative would be causal analysis: repeatedly intervening on candidate internal variables and measuring their downstream effects. Methods based on causal mediation, interchange interventions, causal tracing, and activation patching provide systematic tools for such analyses (Vig et al., 2020; Geiger et al., 2025; Meng et al., 2022; Monea et al., 2024; Zhang & Nanda, 2024). This requires multiple counterfactual executions making them too costly in practice.

Probing offers a complementary and scalable alternative. A probe is trained to predict a target property from model activations using labeled examples (Alain & Bengio, 2017; Belinkov, 2022). Since activations mediate the model's computation, they provide a natural substrate for monitoring signals that may not be faithfully verbalized. Probes have been used to detect hallucinations (Han et al., 2025), strategic deception (Goldowsky-Dill et al., 2025), reward hacking (Bergen et al., 2026),

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.

and unsafe behavior (Saglam et al., 2025). Unlike causal interpretability, the goal is not to establish which variable is causally responsible for the behavior, but to extract a reliable signal for downstream decision making. The main goal is therefore to produce a probe that is robust and generalizes.

Yet, several reports document failures of probes: they can predict labels from randomized computation (Hewitt & Liang, 2019; Voita & Titov, 2020; Méloux et al., 2025), decodability alone does not establish computational relevance (Ravichander et al., 2021; Elazar et al., 2021), and risks of overfitting demand to control the capacity Voita & Titov (2020); Zhu & Rudzicz (2020); Pimentel et al. (2020). More importantly, probes often rely on spurious correlations and fail under distribution shift (Hall Maudslay & Cotterell, 2021; Elazar et al., 2021; Stacey et al., 2026; Aswal et al., 2026). For example, uncertainty probes fail to transfer across environments (Stacey et al., 2026), deception probes can collapse under stylistic shifts (Kumar, 2026), and malicious-input probes may rely on superficial trigger patterns rather than semantic harmfulness (Wang et al., 2026).

In this work, we therefore argue that *out-of-distribution generalization* should be the primary criterion for evaluating probes. We collect three probing tasks spanning 15 distinct environments and evaluate them using a leave-one-environment-out (LOO) protocol to measure cross-environment generalization. Once probing is framed as a standard supervised-learning generalization problem, its specific challenge lies in the structure of its input. A language model produces a high-dimensional activation tensor whose features are strongly structured and dependent across layers, token positions, and representation dimensions. This framing suggests two complementary routes toward robust probes. The first is *representation learning* to address the structured, high-dimensional input by learning a compact, distributed representation of the full computation. The second is *distributionally robust learning*, which directly targets generalization by exploiting environment labels through robust or invariant-learning objectives designed to reduce spurious environment-specific associations.

Along the first axis, inspired by inducing-point attention for scalable set representations (Jaegle et al., 2021; Lee et al., 2019) and axial attention (Ho et al., 2019), we introduce FALA which compresses the full activation tensor together with a sparse layer-selection mechanism. This allows the probe to integrate information distributed across the computation while retaining only a sparse set of layers for efficient inference after training. Across our three tasks and 15 environments, FALA consistently improves LOO generalization over probing baselines.

Along the second axis, we benchmark a broad collection of invariant and distributionally robust learning objectives. Surprisingly, none provides a consistent improvement over standard empirical risk minimization (ERM). Our analyses suggest that, when ERM is trained on sufficiently diverse environments, the representations learned by FALA are already substantially less environment-specific, leaving limited room for further gains from explicit invariance objectives. This echoes recurring worries in the field of OOD generalization (Gulrajani & Lopez-Paz, 2021).

Our contributions. (i) We formulate monitoring probes explicitly as an OOD generalization problem and construct a benchmark of three probing tasks spanning 15 environments, evaluated through leave-one-environment-out generalization. (ii) We introduce FALA, a distributed probing architecture that compresses the complete activation tensor through two-stage inducing-point attention and sparse layer selection, and show that it substantially improves OOD generalization over conventional probes while retaining efficient inference. (iii) We conduct a systematic comparison of invariant and distributionally robust training objectives and find that they provide no consistent improvement over ERM on top of the learned distributed representations; our analysis links this negative result to representations that are already comparatively invariant under ERM with diverse data. (iv) From these results, we derive a simple practical recipe for practionners: evaluate probes across held-out environments; learn distributed representations like FALA with sparsity constraints; and prioritize diverse training environments before introducing more elaborate robust-learning objectives.

2 Related Work

2.1 Probing and Activation Readouts

Probing trains a classifier to predict a property of interest from a model's internal representations. Early work decoded linguistic properties from static embeddings and recurrent hidden states (Köhn, 2015; Gupta et al., 2015; Shi et al., 2016); probing is now a standard tool for analyzing modern

models (Alain & Bengio, 2017; Belinkov, 2022; Zhou et al., 2026) and for monitoring behaviors such as hallucination, uncertainty, refusal, and reward hacking (Chen et al., 2024; Ling et al., 2024; Arditi et al., 2024; Bergen et al., 2026).

Most probes read a restricted view of the computation, selecting a single layer and token, or a fixed pooling (Mosbach et al., 2020). Attention-based probes learn how to aggregate token representations (Shabalin & Belrose, 2025; Kantamneni et al., 2025; McKenzie et al., 2026) and more recent work operates over larger portions of the activation tensor; ACT-ViT jointly models activations across layers and tokens and improves hallucination detection (Bar-Shalom et al., 2025). Our approach follows this latter direction, treating the layer–token activation grid as a structured input. Architecturally, our approach builds on ideas from inducing-point and latent attention for compressing large structured inputs (Jaegle et al., 2021; Lee et al., 2019) and axial attention for processing multidimensional tensors (Ho et al., 2019).

Probing nevertheless has important well-documented methodological limitations (Belinkov, 2022): accuracy depends partly on classifier capacity (Hewitt & Liang, 2019) and can stay high for random representations, motivating selectivity controls and information-theoretic measures (Voita & Titov, 2020; Pimentel et al., 2020), and decodability does not imply that the model uses the decoded information (Elazar et al., 2021). For monitoring, the most consequential limitation is generalization: probes that perform well in-distribution can fail to transfer across environments (Stacey et al., 2026). Distributed readouts may help: information spread across layers and token positions can outperform fixed single-location readouts (Zhu et al., 2024; Kim et al., 2025; Binkowski et al., 2025). We therefore ask whether distributed probing strategies generalize to unseen environments, rather than only whether they decode well in-distribution.

2.2 Invariant and Distributionally Robust Learning

Our second axis connects probing to the broader literature on domain generalization and invariant learning (Peters et al., 2015; Zhou et al., 2023). Given multiple training environments, these methods seek predictors that remain reliable when environment-specific correlations change. Invariant Risk Minimization (IRM) searches for representations supporting the same optimal predictor across environments (Arjovsky et al., 2020); V-REx penalizes variation in risk across environments (Krueger et al., 2021); domain-adversarial objectives encourage representations from which environment identity cannot be recovered (Ganin & Lempitsky, 2015); and Group-DRO directly optimizes worst-environment performance (Sagawa* et al., 2020).

These ideas have been applied to NLP for out-of-domain generalization (Peyrard et al., 2022; Zheng et al., 2023; Jia & Zhang, 2023), but their benefits are far from universal. Large-scale comparisons in domain generalization find that carefully tuned empirical risk minimization can be difficult to outperform (Gulrajani & Lopez-Paz, 2021). Existing applications to language representations also typically operate on a preselected representation rather than jointly learning a distributed readout over the full activation tensor. We therefore study the two questions jointly: whether richer activation representations improve OOD probing, and whether explicit invariant or robust objectives provide additional gains once such representations are learned.

3 Methodology

Probing under distribution shift. For an input example x, a language model produces a trajectory of hidden states $\boldsymbol{H}(x) \in \mathbb{R}^{N\times L\times d}$, where N is the number of retained tokens, L the number of layers, and d the hidden dimension. Each example is paired with a prediction target $y \in \mathcal{Y}$, e.g., binary classification for hallucination or recognizing data seen during training, and continuous regression for estimating uncertainty.

Following the out-of-distribution generalization literature (Shen et al., 2021; Chen et al., 2023), we associate each example x with an environment $e \in \mathcal{E}$. The training data for a probe therefore comes from several environments $\mathcal{E}_{\text{tr}}$,

$$\mathcal{D}_{\text{tr}} = \bigcup_{e\in\mathcal{E}_{\text{tr}}} \{(\boldsymbol{H}_i, y_i, e)\}.$$

We ask whether a probe learned on these environments generalizes to an unseen environment $e^\star \notin \mathcal{E}_{\text{tr}}$. Given a fixed set of available environments $\mathcal{E}$, we evaluate this using leave-one-environment-out (LOO): for each $e^\star$, the probe is trained on $\mathcal{E} \setminus \{e^\star\}$ and evaluated only on $e^\star$.

A **probe** can be written as $\hat{y} = g_\theta(r_\phi(\boldsymbol{H}))$, where r_ϕ maps the variable-length, multi-layer activation trajectory to a fixed-size representation and g_θ predicts the target. This formulation exposes two complementary routes to out-of-environment generalization. First, r_ϕ can represent signal distributed across tokens to support generalizable predictions. Second, the probe training can use the environment labels to discourage environment-specific spurious associations.

3.1 Learning Distributed Representations of Activations

Here, we discuss how to learn a representation of the activation tensor $\boldsymbol{H} \in \mathbb{R}^{N\times L\times d}$ to produce a fixed-size representation while learning how information should be combined across layers and tokens. We consider two architectures of increasing expressivity: Weighted Average Projection (WAP) and Factorized Axial Latent Attention (FALA).

3.1.1 Weighted Average Projection

WAP is the simplest learned aggregation we consider. It replaces fixed pooling with learned importance weights along the token and layer axes. For each layer ℓ, a learned query $\boldsymbol{q} \in \mathbb{R}^d$ scores the token activations by dot product and induces normalized token weights

$$\alpha_{n\ell} = \frac{\exp\big(\boldsymbol{q}^\top \boldsymbol{H}_{n\ell}\big)}{\sum_{m=1}^{N} \exp(\boldsymbol{q}^\top \boldsymbol{H}_{m\ell})}, \qquad \boldsymbol{u}_\ell = \sum_{n=1}^{N} \alpha_{n\ell} \boldsymbol{H}_{n\ell}. \tag{1}$$

Thus, rather than selecting a token position a priori, WAP learns which positions are informative for the probing task. The resulting L layer summaries are themselves aggregated using learned layer logits $\boldsymbol{a} \in \mathbb{R}^L$,

$$\beta_\ell = \frac{\exp(a_\ell)}{\sum_{j=1}^{L} \exp(a_j)}, \qquad \boldsymbol{z} = \sum_{\ell=1}^{L} \beta_\ell \boldsymbol{u}_\ell. \tag{2}$$

The resulting fixed-size representation $\boldsymbol{z} = r_\phi(\boldsymbol{H}) \in \mathbb{R}^d$ is passed to the prediction head g_θ.

3.1.2 Factorized Axial Latent Attention (FALA)

FALA provides a higher-capacity alternative to the weighted-average structure of WAP by replacing each aggregation step with inducing-point cross-attention (Lee et al., 2019; Ho et al., 2020). Instead of reducing each axis to a single weighted mean, it retains multiple learned summaries, allowing heterogeneous patterns across tokens and layers to be preserved. Before compression, we add separable sinusoidal encodings for layer and token position (see Appendix D.4).

Layer compression. For each token independently, k learned inducing points $\boldsymbol{I}^L \in \mathbb{R}^{k\times d'}$ attend over that token's L layer states, which are projected from d to d' via a shared key/value matrix $\boldsymbol{W}^{KV} \in \mathbb{R}^{d\times d'}$ (applied layer-slice by layer-slice):

$$\boldsymbol{Z}^N = \text{Attn}\big(\boldsymbol{I}^L,\ \boldsymbol{H}^+\boldsymbol{W}^{KV}\big) \in \mathbb{R}^{N\times k\times d'}, \tag{3}$$

where $\text{Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{softmax}\big(\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d'}\big)\boldsymbol{V}$ with $\boldsymbol{K} = \boldsymbol{V} = \boldsymbol{H}^+\boldsymbol{W}^{KV}$ under the shared projection, followed by an output projection $\boldsymbol{W}^O$ and LayerNorm. The inducing points $\boldsymbol{I}^L$ serve directly as queries: a separate query projection would be redundant, since a learnable matrix times a learnable parameter collapses to a single learnable parameter.

Token compression. A second set of k inducing points $\boldsymbol{I}^N \in \mathbb{R}^{k\times d'}$ — the same grid size k as Stage 1 — then compresses the token dimension independently for each of the k layer summaries:

$$\boldsymbol{Z} = \text{Attn}\big(\boldsymbol{I}^N,\ \boldsymbol{Z}^N\boldsymbol{W}^K,\ \boldsymbol{Z}^N\boldsymbol{W}^V\big) \in \mathbb{R}^{k\times k\times d'}, \tag{4}$$

again followed by $\boldsymbol{W}^O$ and LayerNorm; unlike Stage 1, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$ are independent here, since Stage 2 operates entirely in the small d'-dimensional space where sharing them would save little. The result is a fixed $k\times k\times d'$ latent tensor regardless of the generation length N.

3.1.3 CAPACITY, DIMENSIONALITY REDUCTION, AND SPARSE LAYER SELECTION

WAP and FALA differ primarily in their representational capacity. WAP reduces the activation trajectory through successive weighted averages, whereas FALA retains $k \times k'$ latent summaries and introduces cross-attention along both axes. Both operate on the full $N \times L$ trajectory, but WAP provides a substantially lighter aggregation mechanism at the cost of a more restrictive representation.

For both architectures, we can first project the hidden states from the model dimension d to a smaller probe dimension d' using a shared linear map $\widetilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{W}^P, \quad \boldsymbol{W}^P \in \mathbb{R}^{d \times d'}$. The same WAP or FALA architecture is then applied to $\widetilde{\boldsymbol{H}}$, with d' replacing d throughout. For example, on Llama-3.1-8B-Instruct (L=33, d=4096), WAP has 12,323 trainable parameters and FALA 292,774.

For both architectures, we additionally consider sparse layer selection using a Gumbel-Softmax top-K selector before layer aggregation. During training, the selector uses a straight-through estimator to select discrete subsets of layers while preserving gradient-based optimization; at inference, only the K layers with the highest learned logits are provided to the probe. Layer selection is therefore itself task-dependent while imposing an explicit sparsity constraint on the representation. We use $K = 3$, selected once on `env3_coqa` for S1 from $K \in \{2, 3, 4, 5, 6\}$, and keep it fixed for all subsequent signals and environments without further tuning.

3.2 DISTRIBUTIONALLY ROBUST PROBING

Our second axis concerns how the probe is trained across environments. For each environment $e \in \mathcal{E}$, let $P_e(\boldsymbol{H}, y)$ denote the joint distribution of activation trajectories and probing targets in that environment (induced by a distribution over inputs), and let $\ell(\hat{y}, y)$ denote the task loss used to train the probe. For a probe $f_{\theta,\phi} = g_\theta \circ r_\phi$, its expected risk in environment e is

$$R_e(\theta, \phi) = \mathbb{E}_{(\boldsymbol{H},y)\sim P_e}\left[\ell(f_{\theta,\phi}(\boldsymbol{H}), y)\right].$$

The environment-generalization problem is to learn $f_{\theta,\phi}$ using only the observed training environments $\mathcal{E}_{\text{tr}}$, while achieving low risk on an unseen environment $e^\star \notin \mathcal{E}_{\text{tr}}$. Since $P_{e^\star}$ is unavailable during training, different methods use different surrogate objectives to encourage transfer beyond the observed environments.

ERM minimizes the average empirical risk across the training environments mixed together. We compare it with representative existing domain-generalization methods: **DANN** (Ganin & Lempitsky, 2015) adversarially removes environment-predictive information from the representation; **CORAL** (Sun & Saenko, 2016) aligns representation statistics across environments; **V-REx** (Krueger et al., 2021) penalizes variation in risk across environments; **Group-DRO** (Sagawa* et al., 2020) optimizes against adverse reweightings of the training environments; supervised contrastive learning (**SupCon**) (Khosla et al., 2020) aligns same-label examples across environments; inter-environment **mixup** (Zhang et al., 2018; Verma et al., 2019) augments training with interpolations across environments; and a bi-level meta-learning objective inspired by **HUME** (Gadetsky & Brbic, 2023) explicitly optimizes transfer between subsets of the observed environments. Implementations follow each method's core idea; details in App. E.

4 EXPERIMENTAL SETUP

4.1 SIGNALS

Hallucination (S1). We construct six environments spanning diverse domains and task structures: closed-book factoid QA (TriviaQA) (Joshi et al., 2017), real user queries (Natural Questions) (Kwiatkowski et al., 2019a), grounded reading comprehension (SQuAD v2) (Rajpurkar et al., 2018), and conversational multi-turn dialogue (CoQA) (Reddy et al., 2019a); plus two high-shift settings: three-way inference (MNLI) (Williams et al., 2018) and scientific multiple choice (SciQ) (Welbl et al., 2017). For each input, the target label is binary: an instance is labeled correct only if the model succeeds across all 10 independent generations sampled from `Llama-3.1-8B-Instruct` at temperature 1.0 (top-p 0.9, top-k 50). This produces 10,100 total samples. We report AUROC as the primary evaluation metric alongside the validation–test gap to monitor overfitting. Full details on environment construction, label assignment, and per-environment sizes are in App. A.

Uncertainty Calibration (S2). Using the same model (`Llama-3.1-8B-Instruct`) and the first five environments from S1, S2 shifts from binary correctness to predicting the empirical frequency of correct answers over $n = 10$ resamples at temperature 1.0. S2 isolates ambiguous items by retaining only the non-unanimous samples from S1, excludes SciQ because only 42 of its samples are non-unanimous, leaving 8,321 training and 1,841 test instances. Rather than classification, this is cast as a calibrated regression task via quantile-conditioned quantile regression with the pinball loss. The primary metric is Cov80—the empirical coverage of the predicted 80% interval against the nominal 0.80 target—reported alongside interval width (to guard against artificially inflated coverage via wide bands), with Spearman ρ and pinball loss as secondary metrics. Details in App. B.

Training-Data Familiarity (S3). S3 targets a distinct capability using an open-weights model with fully documented pre-training data: `OLMo-3-1025-7B`, whose Dolma-3 corpus is publicly accessible (Olmo et al., 2026). The probe predicts whether an input was present in pre-training based on internal activations. We curate four web-text domains (Wikipedia, arXiv, Common Crawl, Stack Exchange), each with 1,000 *seen* passages drawn from Dolma-3 and 1,000 *unseen* passages sourced from post-cutoff documents. Performance is evaluated via AUROC and the validation–test gap. App. C details the split construction.

Activation extraction. Hidden states are extracted across all layers. Standard linear probes read from the final response token, whereas distributed probes consume the full $L \times N \times d$ activation grid. For S1 and S2, we restrict sequence length N strictly to generated response tokens, following Stacey et al. (2026), who show that incorporating prompt tokens degrades out-of-distribution generalization by roughly $3\times$. Because S3 measures input familiarity without generative output, it retains all prompt positions up to a 2,048-token ceiling.

4.2 Testing Distributed Probes

Rather than probing an isolated representation slice, we test whether conditioning on the model's entire computational volume improves out-of-distribution transfer. For FALA, we fix $k = 6$ inducing points per stage and projected dimension $d' = 64$, paired with a Gumbel selector budget of $K = 3$; these hyperparameters are tuned once on the hardest S1 validation fold and held constant across all signals. Positional encodings are sinusoidal, while optimization hyperparameters vary by architecture and signal (App. D). We benchmark *FALA* and *WAP*—both with and without Gumbel sparse layer selection—against three baseline representations: (1) *last-token linear*, a linear probe trained on the last token's activation at each layer; (2) *concat-layers*, which concatenates the last token's activation across all layers; and (3) *mean-tokens*, which mean-pools over all tokens within each layer. We add two controls to rule out false effects: *MLP with* d'=71, a one-hidden-layer MLP on the same single-token input with parameter count matched to FALA-G (FALA with Gumbel-sparse selection); and *random-init*, which probes the same model with randomized weights.

4.3 Testing Distributionally Robust Training Methods

Fixing the highest-performing distributed probes (*FALA-G* and WAP-G), we evaluate whether distributionally robust optimization improves generalization relative to standard ERM. We test seven methods, DANN, CORAL, Group-DRO, V-REx, SupCon, HUME, and Mixup, described in §3.2.

To make exhaustive sweeps tractable across this method suite, we evaluate on a fixed seven-fold benchmark subsampled across all three signals: S1 ({CoQA, NQ, MNLI}), S2 ({NQ, MNLI}), and S3 ({arXiv, StackEdu}). Folds were pre-selected using criteria determinable *a priori* without probe training: statistical reliability, which excludes environments whose test sets are too small for a stable estimate, and structural coverage, which keeps the suite from being drawn from a single task format or domain. Suite construction is detailed in App. F.

5 Results

5.1 Distributed Probing Improves Representation Quality

Figure 1 compares the performance of WAP and FALA, both with and without Gumbel-sparse selection (FALA/WAP-G denote the Gumbel-selected variants), against our baselines and controls. Full

per-fold results are in Appendix G. Among the baseline methods, concatenating last-token representations across all 33 layers yields no clear improvement over standard linear probing on the last token of the validation-selected best layer: 0.757 vs. 0.759 on S1, 0.004 vs. 0.020 on S2, and 0.595 vs. 0.612 on S3. In contrast, average-pooling representations across tokens at the validation-selected layer provides modest gains on two signals: 0.777 vs. 0.759 on S1 and 0.676 vs. 0.612 on S3.

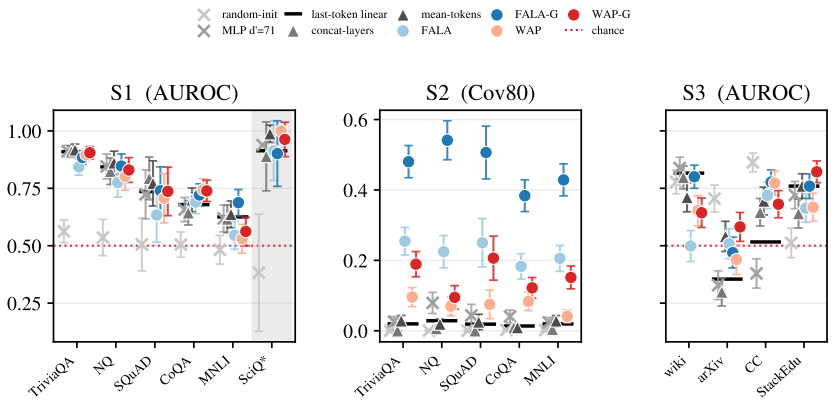

Figure 1: **Distributed probes vs. non-distributed baselines and controls.** Each point is one held-out fold; markers distinguish model classes (△ non-distributed, • distributed, × controls). The black tick marks the *last-token linear* LOO baseline within the non-distributed class. Dotted lines denote chance; error bars are bootstrap confidence intervals over the test split. *SciQ has n_{test}=18 (shaded). Per-fold tables: Tables 12, 13, and 14.

For our proposed distributed architectures, Gumbel-sparse selection provides critical regularization: FALA-G and WAP-G consistently rank as the top two performing probes across signals. By contrast, their dense, full-capacity counterparts perform substantially worse, occasionally falling below non-distributed baselines, most likely due to overfitting.

A parameter-matched capacity control (a single-hidden-layer MLP with the same parameter count as FALA-G) still performs significantly worse than FALA-G. This confirms that probe capacity alone is insufficient; predictive power requires pooling signals from diverse, multi-layer representations. The randomized-model control achieves surprisingly strong performance in S3, showing that probes can often separate seen from unseen inputs purely via random projections of the raw text embeddings. Indeed, the baseline probes fail to outperform the randomized-model one: a classic "dead salmon" artifact (Méloux et al., 2025). Unlike these baselines, FALA-G performs significantly above the randomized-model control on all signals, including S3.

While the performance advantages of FALA-G over the best baseline on S1 and S3 are marginal, S2 displays a large, statistically significant improvement. Furthermore, analysis of the validation–test generalization gap reveals that FALA-G exhibits the smallest performance degradation from validation to test among all evaluated probes. Together, these findings indicate that FALA-G provides a robust distributed probing mechanism that improves out-of-distribution (OOD) generalization while remaining computationally efficient at inference time thanks to sparse activation routing.

5.2 No Evidence of Improvements from Distributionally Robust Training

For the two best-performing distributed probes, FALA-G and WAP-G, we evaluate replacing standard ERM with the suite of distributionally robust optimization objectives introduced in Section 3. Figure 2 shows the average performance gain over ERM across test scenarios. Strikingly, none of the seven distributionally robust objectives yields a consistent improvement: medians all fall within ± 0.01 of zero, narrower than the run-to-run spread of any seeded arm, and fewer than half of the tested configurations yield positive gains. Further hyperparameter tuning and ensembling do not change this conclusion. Furthermore, applying distributionally robust training to the baseline linear probe (when applicable) also provides no benefit (Table 10).

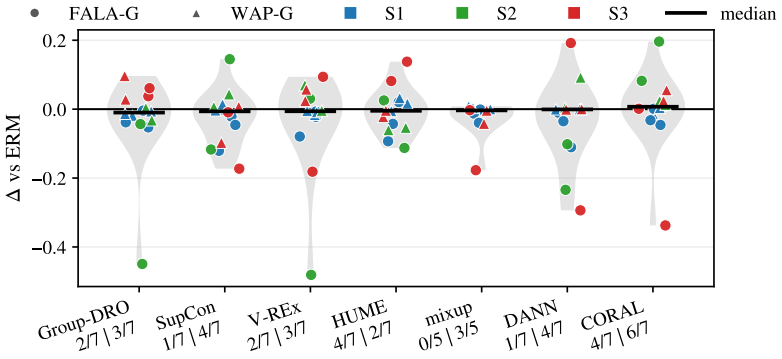

Figure 2: **Distributionally robust training fails to outperform plain ERM.** Each point is one scenario (fold × host) relative to its ERM run, so values above zero are improvements. Colors denote signals, markers host models, bars medians. Annotations beneath each objective give folds beaten out of seven (FALA-G | WAP-G). Per-fold results: Table 11.

5.2.1 Why Does Distributionally Robust Training Fail?

This result is surprising, as distributionally robust training should, in principle, act as a regularizer particularly suited to OOD evaluation and potentially useful to our high-capacity probes. To investigate this, we inspect the latent representation space r_ϕ learned by the probes. Under distributionally robust optimization, this representation is subjected to explicit invariance pressure (e.g., Group-DRO) or environment-independence pressure (e.g., DANN). We hypothesize that ERM trained with data from diverse environments might already naturally enforce such properties. This would also align with some observed empirical limitations of distributionally robust training in other domains (Rosenfeld et al., 2021). To test this, we evaluate two diagnostic quantities on r_ϕ:

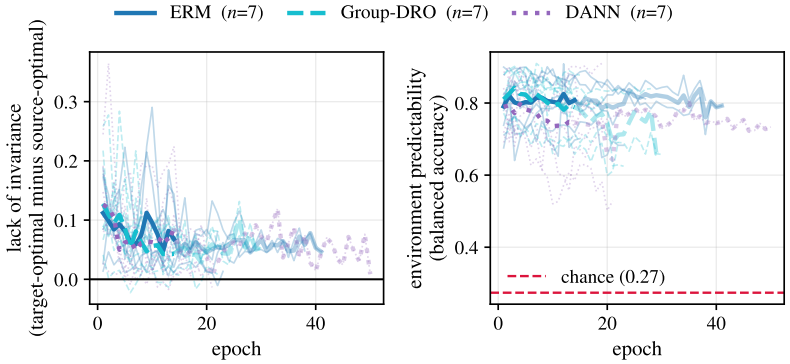

Figure 3: **Representation diagnostics during FALA-G training.** *Left*: lack of invariance, measured by the discrepancy on target data between source- and target-optimal linear heads across training environments; lower is more invariant. *Right*: environment-prediction accuracy of an auxiliary probe on frozen representations (dashed: chance). Both metrics are recomputed each epoch. Faint lines show the seven selection-suite folds; bold lines show their mean. Curves fade once a fold early-stops. Single seed; estimator details in App. H.2.

Environment predictability: We train an auxiliary probe on top of frozen representations r_ϕ to predict the environment index e. If r_ϕ successfully forgets spurious environmental features, classification accuracy on this task should remain near chance.

Lack of invariance: For each environment e, we fit an optimal linear classifier g_e^* atop frozen r_ϕ and evaluate the composite predictor $g_e^* \circ r_\phi$ on every held-out environment e'. We quantify the lack of invariance as the performance discrepancy between using the source-optimal head g_e^* versus the

target-optimal head $g^*_{e'}$ across all pairs (e, e'). If r_ϕ is perfectly invariant across environments, the optimal head for one environment should perform the same as that of any other (Arjovsky et al., 2020).

In Figure 3, we track lack of invariance and environment predictability across training epochs for FALA-G trained with standard ERM, DANN, and Group-DRO. Although DANN explicitly optimizes for environment independence and Group-DRO penalizes worst-group risk (i.e., invariance), neither markedly improve invariance and standard ERM performs as well them. Environment predictability declines slightly during training, but ERM representations paradoxically tend to retain *less* environment information than those trained with distributionally robust objectives, though variance across scenarios remains substantial.

If algorithmic penalties cannot induce beneficial invariance, maybe the benefit of ERM comes from training on data coming from diverse environments. To test this, we scale FALA-G's ERM training data in steps of 100 samples (details in App. I), either by drawing additional samples from the same environments or by sourcing each increment of 100 examples from a previously unseen environment. Across ten evaluation ladders and five held-out folds, adding 100 examples from a novel environment yields an average gain of $+0.037$ AUROC, compared to just $+0.006$ AUROC when adding the same number of samples to existing environments.

6 Discussion and Conclusion

This work starts from the premise that probing is a supervised learning problem, and that its documented failure modes mirror standard supervised-learning pathologies: underdetermination, overfitting, and poor out-of-distribution (OOD) generalization. We therefore argue that robustly monitoring the internal computation of LLMs is a problem of *generalizable* probing, which lets us apply the standard supervised-learning toolkit to evaluate and improve OOD performance. The domain-specific challenge is the input space: activations are high-dimensional and hierarchically structured, with features strongly dependent across layers and token positions, some as the direct computational consequence of others upstream. These dependencies induce representation properties that warrant further study and could be explicitly leveraged to improve generalization.

This methodological reflection again mirrors earlier developments in computational neuroscience. There, the analogue of probing, often called multivariate pattern analysis (MVPA) on functional neuroimaging data (Wager et al., 2013; Kragel et al., 2018), confronted similar challenges, ultimately motivating a shift toward rigorous out-of-sample evaluation across patients and scanner hardware (Gabrieli et al., 2015; Woo et al., 2017).

In this study, we took initial steps toward this challenge for probing by evaluating simple distributed representation learning architectures alongside distributionally robust optimization objectives that exploit multi-environment training data. Our findings show that distributed probing consistently improves OOD generalization over baselines, provided the probe is constrained by sparsity (e.g., via Gumbel-sparse routing) to avoid overfitting. Then, after our best effort implementing a variety of distributionally robust training objectives, we observed no consistent generalization improvements over standard empirical risk minimization (ERM). Our diagnostic analyses suggest that when trained on data coming from diverse environments, standard ERM naturally acquires competitive feature invariance, leaving little room for existing invariant penalties to provide additional benefit.

Recommendations for Practitioners. For practitioners developing internal state monitors and meta-cognitive oversight mechanisms, our findings suggest three guidelines: (i) **Evaluate out-of-distribution:** Benchmark probes on held-out tasks, prompts, or domains rather than relying on in-distribution validation splits; (ii) **Leverage sparse, distributed representations:** Aggregate signals across multiple layers and token positions using explicit sparsity regularizers rather than relying on individual bottleneck layers or dense concatenations; (iii) **Prioritize data diversity over algorithmic invariance:** Allocate data acquisition and labeling budgets toward covering heterogeneous environments rather than engineering complex distributionally robust optimization penalties.

Future Directions. Promising directions for future work include evaluating distributed probes on larger open-weights model families, developing meta-learned distributed routing schemes for fast adaptation to new environment, and adapting distributed probing architectures to streaming, online inference settings where computational efficiency and latency are tightly constrained.

AI USE STATEMENT

In this work, we used generative AI tools to support qualitative and thematic data analysis, assist with writing, and help implement methods. We did not use generative AI tools for any other tasks requiring disclosure, and the remaining disclosure requirements are not applicable to this work.

In addition, we used generative AI tools for tasks with recommended disclosure, including suggesting experimental parameters and brainstorming. All AI-assisted work has been reviewed. We checked LLM-generated text for accuracy and potential hallucinations through re-reading. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available, open-source datasets and does not involve human subjects, personal data, or other sensitive information. Web-crawled environmental data were collected in accordance with the relevant sites' terms of service and contain no personal or identifying information. We declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

The algorithms are described in §3 and the experimental setup in §4. Dataset construction, all datasets, and prompts are in Appendices A, B, and C. Training hyperparameters and implementation details are in App. D. We will open-source our code.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330701. URL https://doi.org/10.1145/3292500.3330701.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2017. URL https://openreview.net/forum?id=ryF7rTqgl.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020. URL https://arxiv.org/abs/1907.02893.

Darpan Aswal, Thomas Palmeira Ferraz, Yongxin Zhou, and Maxime Peyrard. Observable patterns are not explanations: A causal-geometric analysis of latent reasoning models, 2026. URL https://arxiv.org/abs/2606.12689.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation, 2025. URL https://arxiv.org/abs/2503.11926.

Guy Bar-Shalom, Fabrizio Frasca, Yaniv Galron, Yftah Ziser, and Haggai Maron. Beyond token probes: Hallucination detection via activation tensors with ACT-vit. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=aJ7AdfOfij.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Leon Bergen, Usha Bhalla, Andrew Lee, Barak Widawsky, Linas Nasvytis, Connor Watts, Siddharth Boppana, Sidharth Baskaran, Dron Hazra, Michael Byun, Atticus Geiger, Owen Lewis, Matthew Kowal, Vasudev Shyam, Thomas Fel, Thomas McGrath, Ekdeep Singh Lubana, and Jack Merullo. Monitoring and discovering reward hacking with internal representations during llm evaluations, 2026. URL https://arxiv.org/abs/2609.19101.

Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxnZh0ct7.

Jakub Binkowski, Denis Janiak, Albert Sawczyn, Bogdan Gabrys, and Tomasz Jan Kajdanowicz. Hallucination detection in LLMs using spectral features of attention maps. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24354–24385, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1239. URL https://aclanthology.org/2025.emnlp-main.1239/.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Zj12nzlQbz.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think, 2025. URL https://arxiv.org/abs/2505.05410.

Yongqiang Chen, Wei Huang, Kaiwen Zhou, Yatao Bian, Bo Han, and James Cheng. Understanding and improving feature learning for out-of-distribution generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=eozEoAtjG8.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10.

John DE Gabrieli, Satrajit S Ghosh, and Susan Whitfield-Gabrieli. Prediction as a humanitarian and pragmatic contribution from human cognitive neuroscience. *Neuron*, 85(1):11–26, 2015.

Artyom Gadetsky and Maria Brbic. The pursuit of human labeling: A new perspective on unsupervised learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=3GpIeVYw8X.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2015. URL https://arxiv.org/abs/1409.7495.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2025. URL https://arxiv.org/abs/2301.04709.

Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. Detecting strategic deception with linear probes. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=C5Jj3QKQav.

Tobias Groot and Matias Valdenegro-Toro. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. In Anaelia Ovalle, Kai-Wei Chang, Yang Trista Cao, Ninareh Mehrabi, Jieyu Zhao, Aram Galstyan, Jwala Dhamala, Anoop Kumar, and Rahul Gupta (eds.), *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pp. 145–171, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.trustnlp-1.13. URL https://aclanthology.org/2024.trustnlp-1.13/.

Melody Y. Guan, Miles Wang, Micah Carroll, Zehao Dou, Annie Y. Wei, Marcus Williams, Benjamin Arnav, Joost Huizinga, Ian D Kivlichan, Amelia Glaese, Jakub Pachocki, and Bowen Baker. Monitoring monitorability. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=b82fgbMVpz.

Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization. In *International Conference on Learning Representations*, 2021. URL https://mlanthology.org/iclr/2021/gulrajani2021iclr-search/.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 12–21, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1002. URL https://aclanthology.org/D15-1002.

Rowan Hall Maudslay and Ryan Cotterell. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 124–131, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.11. URL https://aclanthology.org/2021.naacl-main.11.

Jiatong Han, Neil Band, Muhammed Razzak, Jannik Kossen, Tim G. J. Rudner, and Yarin Gal. Simple factuality probes detect hallucinations in long-form natural language generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 16209–16226, Suzhou,

China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.880. URL https://aclanthology.org/2025.findings-emnlp.880/.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers, 2019. URL https://arxiv.org/abs/1912.12180.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers, 2020. URL https://openreview.net/forum?id=H1e5GJBtDr.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html.

Chen Jia and Yue Zhang. Memory-based invariance learning for out-of-domain text classification. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1635–1647, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.101. URL https://aclanthology.org/2023.emnlp-main.101/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rNfzT8YkgO.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

Hazel Kim, Tom A. Lamb, Adel Bibi, Philip Torr, and Yarin Gal. Detecting LLM hallucination through layer-wise information deficiency: Analysis of ambiguous prompts and unanswerable questions. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32310–32322, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1644. URL https://aclanthology.org/2025.emnlp-main.1644/.

Arne Köhn. What's in an embedding? analyzing word embeddings through multilingual evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2067–2073, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1246. URL https://aclanthology.org/D15-1246.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, Scott Emmons, Owain Evans, David Farhi,

Ryan Greenblatt, Dan Hendrycks, Marius Hobbhahn, Evan Hubinger, Geoffrey Irving, Erik Jenner, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Madry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, Buck Shlegeris, Martín Soto, Eric Steinberger, Jasmine Wang, Wojciech Zaremba, Bowen Baker, Rohin Shah, and Vlad Mikulik. Chain of thought monitorability: A new and fragile opportunity for ai safety, 2025. URL https://arxiv.org/abs/2507.11473.

Philip A Kragel, Leonie Koban, Lisa Feldman Barrett, and Tor D Wager. Representation, pattern information, and brain signatures: from neurons to neuroimaging. *Neuron*, 99(2):257–273, 2018.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5815–5826. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/krueger21a.html.

Abhishek Kumar, Robert Morabito, Sanzhar Umbet, Jad Kabbara, and Ali Emami. Confidence under the hood: An investigation into the confidence-probability alignment in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 315–334, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.20. URL https://aclanthology.org/2024.acl-long.20/.

Sachin Kumar. Pressure-testing deception probes in LLMs: Scaling, robustness, and the geometry of deceptive representations. In Simon Mille, Sebastian Gehrmann, Patrícia Schmidtová, Ondřej Dušek, Marzieh Fadaee, Kyle Lo, Enrico Santus, and Gabriel Stanovsky (eds.), *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM)*, pp. 472–489, San Diego, California, USA, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-423-1. doi: 10.18653/v1/2026.gem-main.43. URL https://aclanthology.org/2026.gem-main.43/.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 08 2019a. ISSN 2307-387X. doi: 10.1162/tacl_a_00276. URL https://doi.org/10.1162/tacl_a_00276.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019b. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Jack Lindsey. Emergent introspective awareness in large language models, 2026. URL https://arxiv.org/abs/2601.01828.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, Guangji Bai, Liang Zhao, and Haifeng Chen. Uncertainty quantification for in-context learning of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3357–3370, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.184. URL https://aclanthology.org/2024.naacl-long.184/.

Alex McKenzie, Urja Pawar, Phil Blandfort, William Bankes, David Krueger, Ekdeep Singh Lubana, and Dmitrii Krasheninnikov. Detecting high-stakes interactions with activation probes, 2026. URL https://arxiv.org/abs/2506.10805.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022. arXiv:2202.05262.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kiciman, Hamid Palangi, Barun Patra, and Robert West. A glitch in the matrix? locating and detecting language model grounding with fakepedia. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6828–6844, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.369. URL https://aclanthology.org/2024.acl-long.369/.

Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. On the Interplay Between Fine-tuning and Sentence-level Probing for Linguistic Knowledge in Pre-trained Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2502–2516, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.227. URL https://aclanthology.org/2020.findings-emnlp.227.

Maxime Méloux, Giada Dirupo, François Portet, and Maxime Peyrard. The dead salmons of ai interpretability, 2025. URL https://arxiv.org/abs/2512.18792.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals, 2015. URL https://arxiv.org/abs/1501.01332.

Maxime Peyrard, Sarvjeet Ghotra, Martin Josifoski, Vidhan Agarwal, Barun Patra, Dean Carignan, Emre Kiciman, Saurabh Tiwary, and Robert West. Invariant language modeling. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5728–5743, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.387. URL https://aclanthology.org/2022.emnlp-main.387/.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4609–4622, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL https://aclanthology.org/2020.acl-main.420.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3363–3377, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295.

Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 05 2019a. ISSN 2307-387X. doi: 10.1162/tacl_a_00266. URL https://doi.org/10.1162/tacl_a_00266.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019b. doi: 10.1162/tacl_a_00266. URL https://aclanthology.org/Q19-1016.

Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=BbNIbVPJ-42.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Baturay Saglam, Paul Kassianik, Blaine Nelson, Sajana Weerawardhena, Yaron Singer, and Amin Karbasi. Large language models encode semantics and alignment in linearly separable representations. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2282–2303, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. doi: 10.18653/v1/2025.ijcnlp-long.124. URL https://aclanthology.org/2025.ijcnlp-long.124/.

Stepan Shabalin and Nora Belrose. Adding attention to linear probes, 2025. URL https://blog.eleuther.ai/attention-probes/.

Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *ArXiv*, abs/2108.13624, 2021. URL https://api.semanticscholar.org/CorpusID:237364121.

Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1159. URL https://aclanthology.org/D16-1159.

Joe Stacey, Hadas Orgad, Kentaro Inui, Benjamin Heinzerling, and Nafise Sadat Moosavi. Hidden failures in robustness: Why supervised uncertainty quantification needs better evaluation, 2026. URL https://arxiv.org/abs/2604.11662.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV 2016 Workshops*, 2016.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bzs4uPLXvi.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/verma19a.html.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias, 2020. URL https://arxiv.org/abs/2004.12265.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://aclanthology.org/2020.emnlp-main.14.

Tor D Wager, Lauren Y Atlas, Martin A Lindquist, Mathieu Roy, Choong-Wan Woo, and Ethan Kross. An fmri-based neurologic signature of physical pain. *New England Journal of Medicine*, 368(15):1388–1397, 2013.

Cheng Wang, Zeming Wei, Qin Liu, Wenxuan Zhou, and Muhao Chen. False sense of security: Why probing-based malicious input detection fails to generalize. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26100–26113, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1300. URL https://aclanthology.org/2026.findings-acl.1300/.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Choong-Wan Woo, Luke J Chang, Martin A Lindquist, and Tor D Wager. Building better biomarkers: brain models in translational neuroimaging. *Nature neuroscience*, 20(3):365–377, 2017.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods, 2024. URL https://arxiv.org/abs/2309.16042.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Carolina Zheng, Claudia Shi, Keyon Vafa, Amir Feder, and David Blei. An invariant learning characterization of controlled text generation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3186–3206, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.179. URL https://aclanthology.org/2023.acl-long.179/.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain Generalization: A Survey . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(04):4396–4415, April 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3195549. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3195549.

Yongxin Zhou, Changshun Wu, Philippe Mulhem, Didier Schwab, and Maxime Peyrard. What matters to an LLM? behavioral and computational evidences from summarization. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 5712–5737, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. URL https://aclanthology.org/2026.findings-eacl.302/.

Derui Zhu, Dingfan Chen, Qing Li, Zongxiong Chen, Lei Ma, Jens Grossklags, and Mario Fritz. PoLLMgraph: Unraveling hallucinations in large language models via state transition dynamics. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4737–4751, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.294. URL https://aclanthology.org/2024.findings-naacl.294/.

Zining Zhu and Frank Rudzicz. An information theoretic view on selecting linguistic probes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9251–9262, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.744. URL https://aclanthology.org/2020.emnlp-main.744.

A HALLUCINATION (S1)

We construct a six-environment hallucination detection dataset spanning diverse task structures and domains. Following (Chen et al., 2024), we include: **TriviaQA** (Joshi et al., 2017) (9,960 samples, closed-book factoid QA), **Natural Questions** (Kwiatkowski et al., 2019b) (3,610 samples, real user queries), **SQuAD v2** (Rajpurkar et al., 2018) (5,928 answerable samples, reading comprehension with passages), and **CoQA** (Reddy et al., 2019b) (7,983 QA pairs, conversational multi-turn dialogue). We augment with two additional high-shift environments: **MNLI** (Williams et al., 2018) (9,815 premise-hypothesis pairs, 3-way NLI classification) and **SciQ** (Welbl et al., 2017) (1,000 samples, scientific multiple-choice with 88.7% having support passages).

Response Collection and Consistency Labeling. For each question, we generate 10 independent responses from the model M. We employ strict unanimity labeling: questions where all 10 responses are judged correct constitute faithful examples, while questions where all 10 are incorrect constitute hallucination examples.

Prompts. The prompts used for each dataset are summarized in Tables 1–3. Table 1 shows the unified prompt for all QA tasks (TriviaQA, Natural Questions, SQuAD, and CoQA), which differs only in the presence or absence of context. The use of story in CoQA and context in SQuAD follows previous work (Chen et al., 2024). Table 2 presents the prompt for the natural language inference task (MNLI), while Table 3 displays the prompt for multiple-choice science questions (SCIQ) with shuffled options to mitigate position bias.

Judges. To select a reliable and cost-effective judge, we evaluated multiple LLMs on 20 samples per environment and chose `GPT-4o-mini` based on the accuracy–cost tradeoff, matching the choice in Binkowski et al. (2025). We use task-adaptive judging: `GPT-4o-mini` as an LLM judge for QA environments, rule-based label extraction for MNLI (checking whether all ten responses predict the same label matching ground truth), and rule-based letter/text matching for SciQ multiple-choice, with randomly shuffled options.

Environment statistics. Table 4 gives the balanced hallucinated (H) and faithful (F) counts with 70/15/15 splits.

Prompt for the four QA Tasks

Without Context (TriviaQA, NQ-Open):
`Question: {question}`
Respond with the answer only, without any explanation.

With Context (SQuAD, CoQA):
`Context: {context}`
`Question: {question}`
Respond with the answer only, without any explanation.

Table 1: Unified prompt template for QA datasets. Context is included only for SQuAD and CoQA.

Prompt for MNLI (Natural Language Inference)

Premise: {`premise`}

Hypothesis: {`hypothesis`}

Does the hypothesis follow from the premise? Respond with ONLY ONE of these words: entailment, neutral, or contradiction.

Table 2: Prompt for MNLI natural language inference task.

Prompt for SCIQ (Multiple-Choice Science QA)

Question: {`question`}

Options:

`A. {option1}`
`B. {option2}`
`C. {option3}`
`D. {option4}`

Answer with only the letter (A, B, C, or D).

Table 3: Prompt for SCIQ multiple-choice science questions with shuffled options to avoid position bias (without support message).

B Uncertainty Calibration (S2)

B.1 Environment construction

We use the same environments and model as S1, excluding env5_sciq. Rather than binary correctness, each sample gets a continuous label $p_{\text{correct}} = \text{count}(\texttt{true})/10 \in \{0.0, 0.1, \ldots, 1.0\}$, the fraction of correct responses across ten repeated queries—a measure of response reliability and epistemic uncertainty. Rather than predicting a point estimate of p_{correct}, we ask whether hidden states support a *calibrated* estimate: whether a probe's predicted quantiles have the correct empirical coverage, not merely the correct rank order.

S2 reuses five of the six environments (SciQ is dropped: only 42 samples fall in the uncertain middle). The raw p_{correct} distribution is strongly bimodal, 60–95% of samples per environment sit at $p{=}0$ or $p{=}1$, so S2 would otherwise be a relabelled copy of S1. We therefore keep all samples with $p \in [0.2, 0.8]$, up to 1,500 per environment stratified by p-bucket, and add binary endpoints at a 1:1 ratio to anchor the regression. Split sizes are in Tab. 5.

Table 4: S1 hallucination environments: balanced counts and splits.

Environment	**H**	**F**	Train	Val	Test	Total
TriviaQA	1,568	1,568	2,194	470	472	3,136
Natural Questions	671	671	938	200	204	1,342
SQuAD	279	279	390	82	86	558
CoQA	1,370	1,370	1,916	410	414	2,740
MNLI	1,111	1,111	1,554	332	336	2,222
SciQ	51	51	70	14	18	102
Total	5,050	5,050	7,062	1,508	1,530	10,100

All S2 probes are trained identically with quantile-conditioned quantile regression (QCQR, §B.2), so the comparison isolates the representation rather than the training loss. A point-estimate variant (MSE for the linear probes, soft-BCE for FALA) is used only for within-environment upper bounds, not for the cross-method LOO comparison.

Table 5: S2 uncertainty calibration environments: middle-focused splits (70/15/15).

ID	**Dataset**	**Train**	**Val**	**Test**	**Total**
env0	TriviaQA	2,092	445	460	2,997
env1	Natural Questions	1,426	303	316	2,045
env2	SQuAD	696	145	160	1,001
env3	CoQA	2,012	426	443	2,881
env4	MNLI	2,095	443	462	3,000
Total		8,321	1,762	1,841	11,924

Cov80 can be bought by widening: predicting $[0, 1]$ everywhere scores 1.00 with no information. We therefore backfilled interval width and mean pinball loss onto every S2 checkpoint by forward passes alone. Coverage does track width, which is why the check was needed—but **pinball ranks the methods in the same order** (FALA-G 0.104, FALA 0.130, WAP 0.143), and pinball is a proper scoring rule minimized only at the true quantile, so widening cannot improve it. FALA-G places its quantiles better, not merely wider.

T1 and V-REx show the smallest mean gaps ($+0.005$, $+0.002$), but both their validation and test coverage are already collapsed near zero, and the gap between two uninformative numbers is mechanically small. Among the methods where the comparison is meaningful, FALA-G's $+0.010$ is less than half WAP's $+0.022$, while its test coverage is about six times larger.

The target $p_{\text{correct}}{=}k/10$ is a noisy Binomial estimate (sd ≈ 0.16 at $p{=}0.5$), which caps achievable coverage. Selection is on validation ρ while Cov80 is primary, so reported checkpoints optimize ranking rather than calibration; selecting on a calibration-aware criterion is the obvious refinement. And Cov80 on this configuration is high-variance across seeds (sd 0.148 on env3_coqa), so single-seed S2 coverage differences below roughly 0.15 are not results.

B.2 Quantile-Conditioned Quantile Regression (QCQR)

Motivation. A point-estimate probe predicts $\mathbb{E}[p_{\text{correct}} \mid x]$ but cannot express uncertainty about its own prediction. To obtain calibrated prediction intervals we train *one* model that conditions on the target quantile level τ at every forward pass, rather than training K separate models for K fixed levels.

Architecture. Let $\mathbf{z} \in \mathbb{R}^d$ be the probe's pooled feature vector. The quantile level τ is appended as a scalar, $\tilde{\mathbf{z}}(\tau) = [\mathbf{z};\ \tau] \in \mathbb{R}^{d+1}$, and a single linear head operates on it:

$$\hat{Q}_\tau = \sigma\big(\mathbf{w}_z^\top \mathbf{z} + w_\tau\, \tau + b\big). \tag{5}$$

The weight w_τ is the only extra parameter over a standard point estimator.

Training. At each step $\tau \sim \mathrm{Unif}(0,1)$ is sampled, and the prediction is trained with the pinball loss

$$\ell_\tau(y, \hat{Q}) = \max\big((\tau - 1)(y - \hat{Q}),\ \tau(y - \hat{Q})\big). \tag{6}$$

Inference. At test time the probe is queried at $\mathcal{T} = \{0.1, 0.25, 0.5, 0.75, 0.9\}$; the five predictions are sorted before metric computation to prevent quantile crossing.

C Training-Data Familiarity (S3)

Environments. Environment construction is model-specific. For this signal, we use **OLMo-3-1025-7B**, a 7B-parameter base model with L=32 layers and d=4096. Its training corpus (Dolma 3, `allenai/dolma3_mix-6T-1025-7B`) is publicly documented, allowing exact seen/unseen labeling. We prefer the base model over the instruction-tuned variant because SFT/RLHF fine-tuning may redistribute or suppress pre-training familiarity signals in hidden states.

Seen data. Seen records are drawn from the HuggingFace repository `allenai/dolma3_mix-6T-1025-7B`, which contains the full Dolma 3 pre-training corpus. Per-environment subsets are streamed with random shuffling; texts are retained only after passing quality gates (150+ words, ≥45% alphabetic ratio for prose; 10+ non-blank lines for code).

Unseen data. Unseen records are collected from the *same source families* as seen, but restricted to content published strictly after 2025-01-01 (OLMo-3's training cutoff): (i) `wiki`: MediaWiki `logevents` API (type=create, ns=0) filtered by page creation timestamp; (ii) `arxiv`: arXiv API with `published > 2025-01-01`; (iii) `common_crawl`: WET file parsing with WARC date filter; (iv) `stack_edu`: GitHub Search API (`created:`≥`2025-01-01`), fetching source files from default branches. Collecting `stack_edu` unseen requires a GitHub personal access token to exceed the 60 requests/hour anonymous rate limit; without one, the environment yields fewer than ten records.

The different environments for training data familiarity is presented in Table 6. All S3 results use a uniform 300-sample training point per domain.

Table 6: S3 training-data familiarity environments. Each combines a Dolma 3 seen source with a post-cutoff unseen source from the same domain. Splits are 70/15/15, stratified by label.

Env	Domain	Seen source	Unseen source
`wiki`	Encyclopaedic prose	`dolma1_7-wiki-en` (Dolma 3)	Wikipedia/Wikibooks pages created after 2025-01-01
`arxiv`	Scientific preprints	`dolma3-arxiv` (Dolma 3)	arXiv submissions published after 2025-01-01
`common_crawl`	Web text	`dolma3-cc` (Dolma 3)	Common Crawl WET files with WARC date after 2025-01-01
`stack_edu`	Educational code	`dolma3-stack-edu` (Dolma 3)	GitHub repositories created after 2025-01-01

Preprocessing. All texts undergo Unicode NFKC normalisation, HTML entity unescaping, whitespace compression, and removal of boilerplate phrases. Wikipedia pages containing the {{`AI-generated`}} template marker are excluded from the unseen split to avoid distributional artefacts introduced by LLM-generated encyclopaedic content (<2% of collected `wiki` unseen records).

Deduplication and Contamination Control. Three-level deduplication prevents information leakage: (1) *exact dedup*: SHA-256 hash check within each split; (2) *near-dup dedup*: Jaccard similarity on word 5-grams (threshold τ=0.7) within each split; (3) *cross-split contamination filter*: a 13-gram index is built from all seen texts and applied to every unseen candidate; records with >1% n-gram overlap are discarded.

D Training and Implementation Details

D.1 Hyperparameter Optimization

All trainable probes are tuned withOptuna (Akiba et al., 2019) using the Tree-structured Parzen Estimator (TPE) sampler (seed 42) and a SQLite study backend. HPO is run exclusively on the held-out fold `env3_coqa` (lowest last-token linear LOO AUROC, largest validation-to-test gap), and best parameters are transferred to the remaining folds.

D.2 Training Hyperparameters

Table 7 lists the optimization and early-stopping settings that vary by probe family and signal. The corresponding searches follow.

The linear probe is trained full-batch on pooled activations, so its batch column is empty. The S3 FALA family's 3-epoch patience is paired with an 8-epoch floor because that signal's validation is noisy enough early on that patience alone terminated runs before they had trained (App. D.3).

Table 7: **Optimization and early-stopping settings per probe family and signal.** Floor: the minimum-epoch count below which early stopping cannot fire; "–" means none.

Signal	Family	lr	batch	patience	floor
S1	linear	10^{-2}	–	10	–
S1	FALA	2.05×10^{-3}	16	7	–
S1	WAP	2×10^{-4}	32	10	1
S2	linear	10^{-3}	–	15	–
S2	FALA	5×10^{-4}	4	7	–
S2	WAP	2×10^{-4}	8	10	1
S3	linear	10^{-2}	–	10	–
S3	FALA	2.05×10^{-3}	4	3	8
S3	WAP	2×10^{-4}	8	10	1

D.3 Early Stopping and Minimum-Epoch Floors

The early-stopping settings vary by family and signal (Tab. 7). Patience is counted from the best validation score so far. Because validation is noisy early in training, a chance initial peak can otherwise terminate a run before learning begins. This occurred in three initial objective runs: adding a minimum-epoch floor changed S2 NQ Cov80 from 0.092 and 0.060 to 0.506 and 0.513 for GroupDRO and V-REx, and S3 arXiv AUROC from 0.133 to 0.420 for CORAL. We therefore set a floor per comparison family and apply it equally to an objective and its matched ERM reference. Floors were raised until no run terminated exactly at its floor.

D.4 FALA: Additional Implementation Notes

Beyond the architecture summary in §3.1.2, three implementation details matter for reproducing the exact numbers reported in this paper.

Input normalization is applied per layer-slice. Before the Stage-1 $d \to d'$ projection, each layer's activations are LayerNorm'd one layer at a time rather than on the full $[B{\cdot}N, L, d]$ tensor at once.

Padding is masked twice. Padded token positions are zeroed *before* the Stage-1 key/value projection — so that the additive positional encodings cannot leak a non-zero signal into padding slots — and again *after* Stage-1's LayerNorm, because LayerNorm's learned bias term would otherwise re-populate zeroed positions with a constant before they reach Stage 2.

Sharing $\boldsymbol{W}^{KV}$ means $\boldsymbol{K} = \boldsymbol{V}$ exactly, not tied weights. Under Stage 1's shared projection (§3.1.2), keys and values are the *same tensor* at every step, $\boldsymbol{K} = \boldsymbol{V} = \mathrm{LN}(\boldsymbol{H}^{+})\boldsymbol{W}^{KV}$, rather

than two independently parameterized projections. Stage 2's $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$ remain independent in every configuration we run, since that stage operates entirely in the small d'-dimensional space where sharing would save little.

D.5 Gumbel-Softmax Layer Selection

FALA-G and WAP-G share the top-K selector.

Selection. A learnable logit ℓ_j is maintained for every layer $j \in \{1, \ldots, L\}$. During training, Gumbel noise is added before a temperature-scaled softmax,

$$g_j \sim \mathrm{Gumbel}(0,1), \qquad z_j = \frac{\ell_j + g_j}{\tau}, \qquad \mathrm{soft} = \mathrm{softmax}(\boldsymbol{z}), \tag{7}$$

and the K layers with the largest z_j are kept. This is a genuine subset: discarded layers are never read by Stage 1, so compute scales with K, not L. At evaluation no Gumbel noise is added ($z_j = \ell_j/\tau$), so selection is deterministic given the learned logits.

Forward weighting is uniform; the softmax only carries gradient. The K selected layers do *not* enter Stage 1 (or WAP's layer fusion) weighted by soft. A straight-through estimator instead gives every selected layer equal weight $1/K$ in the forward pass, with the gradient routed through the soft distribution restricted to the selected indices and renormalized, $\widetilde{\mathrm{soft}} = \mathrm{soft}_{\mathrm{sel}} / \sum \mathrm{soft}_{\mathrm{sel}}$:

$$w = \frac{1}{K} + \left(\widetilde{\mathrm{soft}} - \mathrm{sg}[\widetilde{\mathrm{soft}}]\right), \tag{8}$$

where $\mathrm{sg}[\cdot]$ is stop-gradient, so w is numerically $1/K$ but carries $\widetilde{\mathrm{soft}}$'s gradient. This differs from the non-Gumbel variant of both hosts, whose final layer weights (FALA's Stage-1 attention scores; WAP's β_ℓ, §3.1.1) are themselves a softmax over learned, often input-dependent quantities, whereas FALA-G and WAP-G's combination step is a plain average over whichever K layers were selected.

Temperature schedule and choice of K. τ anneals linearly from τ_{start}=5.0 (soft, exploratory) to τ_{end}=0.1 (near-hard) over the training-step fraction. K=3 was chosen once, by a validation-AUROC sweep over $K \in \{2, 3, 4, 5, 6\}$ on `env3_coqa` (§3.1.2), and reused for every fold, signal and host without further tuning.

WAP-specific detail. WAP's per-layer content queries (§3.1.1) are indexed by the selected layers, so each of the K layers is scored by its own query rather than one query shared across all L layers — this is what raises WAP-G's parameter count from bare WAP's 12,323 to 143,428 (§3.1.1).

D.6 Patch Augmentation

FALA + Patches. To mitigate FALA's tendency to overfit under LOO, we apply *subset-patching augmentation*: at each training step we generate n=8 random views of every sample by independently sampling k_L=16 layers from the full L and a fraction τ=0.5 of the minimum valid response length in tokens. The n view losses are averaged before the optimiser step, multiplying effective training diversity by n per epoch without additional forward passes through the LLM.

WAP + Patches. The same patch routine is applied identically on the WAP host: n=8 random views per training step (k_L=16 layers, a τ=0.5 fraction of valid tokens each), with losses averaged before the optimizer step. The one host-specific detail is that the patch routine additionally returns the sampled layer indices, which WAP needs to index its per-layer content queries and layer logits correctly for that view; FALA has no such dependency, since its inducing points are layer-agnostic queries rather than per-layer parameters.

E Distributionally-Robust Objectives

E.1 Objective Definitions

ERM minimizes the mean per-sample empirical risk over the pooled training environments, $\mathcal{L}_{\mathrm{ERM}} = \frac{1}{B}\sum_{i=1}^{B} \ell_i$. We augment ERM with seven domain-generalization objectives. All use only the $E-1$ source environments in each LOO fold.

DANN (adversarial invariance) (Ganin & Lempitsky, 2015). A Gradient Reversal Layer (GRL) reverses the gradient of a domain classifier predicting which of the $E-1$ training environments a sample belongs to (the held-out environment is never seen):

$$\min_{\theta_f,\theta_c} \max_{\theta_d} \mathcal{L}_c(\theta_f,\theta_c) - \lambda_{\text{adv}}\,\mathcal{L}_d(\theta_f,\theta_d), \tag{9}$$

where $\theta_f/\theta_c/\theta_d$ are encoder/task/domain parameters. The GRL coefficient is annealed from 0 to α_{end}.

V-REx (variance penalty) (Krueger et al., 2021). For the environments E_B represented in the current minibatch, penalizes the variance of their mean losses:

$$\mathcal{L}_{\text{VREx}} = \bar{\mathcal{L}} + \beta\ \text{Var}_e\big[\mathcal{L}_e\big], \qquad \bar{\mathcal{L}} = \tfrac{1}{|E_B|}\textstyle\sum_{e\in E_B}\mathcal{L}_e. \tag{10}$$

β=0 recovers ERM. If a batch contains one environment, the variance term is zero.

SupCon (cross-environment supervised contrastive learning) (Khosla et al., 2020). For a minibatch of pre-classifier features $\mathbf{z}_i$, we use their ℓ_2-normalized versions $\bar{\mathbf{z}}_i$ and define the positives for anchor i as samples with the same class label from a different environment, $P(i) = \{p \neq i : y_p = y_i,\ e_p \neq e_i\}$. The implemented penalty is

$$\mathcal{L}_{\text{SupCon}} = -\frac{1}{|A|}\sum_{i\in A}\frac{1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(\bar{\mathbf{z}}_i^\top\bar{\mathbf{z}}_p/T)}{\sum_{a\neq i}\exp(\bar{\mathbf{z}}_i^\top\bar{\mathbf{z}}_a/T)}, \qquad A=\{i:|P(i)|>0\}, \tag{11}$$

and the training loss is $\mathcal{L}_{\text{ERM}} + \lambda\mathcal{L}_{\text{SupCon}}$. Anchors without a cross-environment positive are excluded; if none remain, the step is ERM. We use T=0.1.

Group-DRO (worst-environment reweighting) (Sagawa* et al., 2020). Reweights per-environment losses rather than penalizing a dissimilarity:

$$q_e \ \propto\ q_e\exp\big(\eta\,\mathcal{L}_e\big), \qquad \mathcal{L}_{\text{DRO}} \ =\ \textstyle\sum_e q_e\,\mathcal{L}_e. \tag{12}$$

For pooled batches, we update only environments present in the batch and renormalize their q_e values in the loss; absent environments retain their weights. We use η=0.01.

(Mixup) Inter-environment Mixup (Zhang et al., 2018; Verma et al., 2019). We apply Mixup to the fixed-length pre-classifier feature, rather than to the variable-length activation grid. For each sample i, a partner $\pi(i)$ is drawn uniformly from samples in the minibatch from a different environment. We draw $u_i \sim \text{Beta}(\alpha,\alpha)$ independently per sample and set $\lambda_i = \max(u_i, 1-u_i)$, then train on

$$\tilde{\mathbf{z}}_i = \lambda_i\mathbf{z}_i + (1-\lambda_i)\mathbf{z}_{\pi(i)}, \qquad \mathcal{L}_{\text{mix}} = \frac{1}{B}\sum_i\Big[\lambda_i\ell\big(h(\tilde{\mathbf{z}}_i),y_i\big) + (1-\lambda_i)\ell\big(h(\tilde{\mathbf{z}}_i),y_{\pi(i)}\big)\Big]. \tag{13}$$

If no cross-environment partner is available, a sample is paired with itself and λ_i is set to one. The optional intra-label variant restricts $\pi(i)$ to the anchor's class, reducing the target to y_i. We use Mixup only for classification: it is incompatible with quantile regression, patch augmentation, and the other objectives.

CORAL (non-adversarial moment matching) (Sun & Saenko, 2016). Direct penalty on per-environment feature statistics, no discriminator, no min–max. With $\boldsymbol{\mu}_e, \boldsymbol{v}_e$ the mean/per-dimension variance of the pre-classifier features $\mathbf{z}\in\mathbb{R}^D$ within a minibatch environment $e\in E_B$:

$$\mathcal{L}_{\text{CORAL}} = \bar{\mathcal{L}} + \frac{\lambda}{D}\Big(\tfrac{1}{|E_B|}\textstyle\sum_{e\in E_B}\|\boldsymbol{\mu}_e-\bar{\boldsymbol{\mu}}\|_2^2 + \tfrac{1}{|E'|}\sum_{e\in E'}\|\boldsymbol{v}_e-\bar{\boldsymbol{v}}\|_2^2\Big), \tag{14}$$

where E' restricts the second-moment term to environments with at least four samples in the batch. The variance term is omitted unless $|E'|\geq 2$; the penalty is zero if $|E_B|<2$. We align diagonal covariance estimates, rather than full $D\times D$ covariances.

(HUME-inspired) Bi-level meta-learning (Gadetsky & Brbic, 2023). Unlike the preceding batch penalties, HUME makes cross-environment generalization itself the training signal: at each step draw one environment e^* as inner-test, fit a ridge classifier on the remaining environments' features in closed form via R2-D2 (Bertinetto et al., 2019):

$$\boldsymbol{\alpha} = \left(\mathbf{Z}_{tr}\mathbf{Z}_{tr}^{\top} + \lambda_r \mathbf{I}\right)^{-1}\mathbf{y}_{tr}, \qquad \hat{\mathbf{y}}_{te} = \mathbf{Z}_{te}\mathbf{Z}_{tr}^{\top}\boldsymbol{\alpha}, \qquad \mathcal{L} = \bar{\mathcal{L}} + \lambda \tfrac{1}{n_{te}}\|\hat{\mathbf{y}}_{te} - \mathbf{y}_{te}\|_2^2. \tag{15}$$

The outer gradient flows through the dual ridge solve. We use binary $\{-1, +1\}$ labels, one inner-test draw per step, and a separate ridge readout; the deployed classifier is not adapted. Draws with fewer than two inner-test or four inner-train samples are skipped; if all draws are skipped, the step is ERM.

E.2 CORAL at a Tuned Weight

The default λ=1.0 is negative on the S1 suite. A sweep on CoQA over $\lambda \in \{0.1, 1, 10\}$ selected 0.1; we then evaluated that weight with three seeds on all three S1 suite folds. Table 8 shows that it remains below ERM on average. Thus, the closest candidate to a positive objective result does not survive a matched seed comparison.

Table 8: CORAL at λ=0.1 against ERM, three seeds each, S1 suite folds. Δ is quoted in the *treatment* arm's standard deviations.

Fold	**CORAL** λ=0.1	**ERM**	Δ	**in CORAL** σ
CoQA	0.7228 ± 0.0148	0.7152 ± 0.0036	$+0.008$	$+0.5$
NQ	0.8280 ± 0.0094	0.8292 ± 0.0182	-0.001	-0.1
MNLI	0.6337 ± 0.0828	0.6753 ± 0.0247	-0.042	-0.5
Mean	0.7282	0.7399	$\mathbf{-0.012}$	

F Cross-Signal Selection Suite

Suite construction. Folds were fixed using criteria available before running any method: statistical reliability and structural coverage. We exclude S1 SciQ (n_{test}=18). The remaining suite retains CoQA's conversational structure and MNLI's classification target, so it tests task shift as well as domain shift. For S3, we retain arXiv and StackEdu to cover formal academic prose and instructional Q&A.

Selection protocol. The suite ranks configurations only. Every checkpoint is selected by validation performance pooled over source environments; the held-out test fold is never used for selection. Configurations selected by the suite are subsequently re-run on all folds (App. G).

Reading the tables. Table 9 compares six distributed representations under ERM. Table 10 evaluates loss-space objectives on the linear probe; Table 11 evaluates objectives on the two selected distributed hosts. These tables answer different selection questions and should not be read as a single cell-by-cell comparison.

Table 9: **Step 1: six distributed representations under ERM.** Each cell reports test value (validation–test gap): AUROC for S1 and S3, Cov80 for S2. Compare within columns only.

Representation	**S1** hallucination (AUROC)			**S2** uncertainty (Cov80)		**S3** TDF (AUROC)	
	CoQA	NQ	MNLI	NQ	MNLI	arXiv	StackEdu
FALA	0.690 (+0.163)	0.775 (+0.071)	0.545 (+0.319)	0.225 (+0.047)	0.206 (+0.010)	0.509 (+0.484)	0.663 (+0.336)
FALA+patches	0.742 (+0.101)	0.771 (+0.070)	0.607 (+0.258)	0.323 (+0.004)	0.214 (+0.019)	**0.592** (+0.402)	0.766 (+0.232)
FALA-G	0.715 (+0.154)	**0.847** (+0.011)	**0.688** (+0.188)	**0.541** (+0.024)	**0.429** (−0.012)	0.470 (+0.517)	0.759 (+0.237)
WAP	**0.743** (+0.116)	0.802 (+0.062)	0.530 (+0.348)	0.070 (+0.035)	0.041 (+0.015)	0.439 (+0.550)	0.668 (+0.331)
WAP+patches	0.720 (+0.140)	0.749 (+0.111)	0.563 (+0.317)	0.041 (+0.012)	0.108 (−0.007)	0.514 (+0.478)	0.722 (+0.276)
WAP-G	0.739 (+0.133)	0.830 (+0.035)	0.562 (+0.331)	0.095 (+0.016)	0.152 (+0.001)	0.582 (+0.414)	**0.822** (+0.177)

Table 10: **Step 2: loss-space objectives on the last-token linear probe.** Each cell reports test value (validation–test gap); compare objective rows against ERM. Feature-space objectives have zero gradient on this host.

	S1 (AUROC)			**S2** (Cov80)		**S3** (AUROC)	
Objective	CoQA	NQ	MNLI	NQ‡	MNLI‡	arXiv§	Stack§
Host: last-token linear (last token, one layer)							
ERM (reference)	0.679 (+0.207)	0.843 (+0.030)	0.625 (+0.269)	0.028 (−0.008)	0.019 (+0.008)	0.355 (+0.613)	0.760 (+0.226)
+ Group-DRO	0.682 (+0.191)	0.826 (+0.039)	0.588 (+0.296)	0.038 (−0.006)	0.015 (+0.012)	0.678§ (+0.302)	0.758 (+0.235)
+ V-REx	0.679 (+0.196)	0.820 (+0.046)	0.583 (+0.302)	0.041 (−0.013)	0.013 (+0.017)	0.592§ (+0.385)	0.741 (+0.252)
+ IRMv1 (λ=0.01)	0.683 (+0.189)	0.827 (+0.037)	0.591 (+0.293)	0.032 (−0.006)	0.015 (+0.005)	0.619§ (+0.360)	0.758 (+0.235)

‡ collapsed-coverage regime (0.013–0.041 against 0.43–0.54 on the distributed hosts): differences here are between two uninformative numbers. § S3's replicate spread is 0.32, exceeding every difference in these columns — including Group-DRO's apparent +0.32 on arXiv.

Table 11: **Step 2: distributionally-robust objectives on FALA-G and WAP-G.** Each cell reports test value (validation–test gap) on the seven selection-suite folds. Compare an objective only with the ERM run matched on host, fold, and batch-size family.

	S1 (AUROC)			**S2** (Cov80)		**S3** (AUROC)	
Objective	CoQA	NQ	MNLI	NQ	MNLI	arXiv§	Stack§
Host: FALA-G							
ERM (reference)	0.715 (+0.154)	0.847 (+0.011)	0.688 (+0.188)	0.541 (+0.024)	0.429 (−0.012)	0.470 (+0.517)	0.759 (+0.237)
+ DANN	0.710 (+0.103)	0.812 (+0.007)	0.577 (+0.230)	0.307† (+0.015)	0.550 (−0.054)	0.662 (+0.324)	0.520 (+0.462)
+ CORAL	0.675 (+0.192)	0.848 (+0.005)	0.655 (+0.218)	0.500 (+0.002)	0.346 (+0.005)	0.558§ (+0.382)	0.551§ (+0.426)
+ Group-DRO	0.717 (+0.110)	0.809 (+0.008)	0.652 (+0.226)	0.506 (−0.008)	0.385 (−0.006)	0.508 (+0.479)	0.820 (+0.179)
+ V-REx	0.698 (+0.167)	0.840 (+0.012)	0.608 (+0.268)	0.513 (−0.030)	0.459 (+0.006)	0.564 (+0.426)	**0.836** (+0.162)
+ SupCon	0.699 (+0.167)	0.801 (+0.057)	0.567 (+0.309)	0.424 (−0.002)	0.574 (−0.085)	0.461 (+0.511)	0.587 (+0.409)
+ HUME	0.741 (+0.133)	0.804 (+0.052)	0.594 (+0.286)	0.497 (−0.014)	0.381 (−0.010)	0.661§ (+0.331)	0.822§† (+0.174)
+ mixup	0.720 (+0.141)	0.834 (+0.017)	0.648 (+0.225)	*n/a*	*n/a*	0.467 (+0.522)	0.582 (+0.412)
Host: WAP-G							
ERM (reference)	0.739 (+0.133)	0.830 (+0.035)	0.562 (+0.331)	0.095 (+0.016)	0.152 (+0.001)	0.582 (+0.414)	0.822 (+0.177)
+ CORAL	0.726 (+0.147)	0.840 (+0.023)	0.566 (+0.327)	0.114§ (+0.001)	0.141§ (−0.044)	0.637 (+0.360)	0.846 (+0.154)
+ Group-DRO	0.720 (+0.143)	0.816 (+0.034)	0.556 (+0.326)	0.098 (−0.005)	0.119 (+0.011)	**0.678** (+0.318)	**0.850** (+0.149)
+ V-REx	0.722 (+0.138)	0.825 (+0.033)	0.554 (+0.333)	0.165 (+0.014)	0.147 (−0.006)	0.639 (+0.357)	0.846 (+0.153)
+ SupCon	0.735 (+0.136)	0.827 (+0.039)	0.576 (+0.316)	0.101 (+0.043)	0.195 (−0.038)	0.485 (+0.511)	0.829 (+0.171)
+ HUME	0.734 (+0.155)	0.846 (+0.027)	0.594 (+0.294)	0.047§ (+0.014)	0.056§ (+0.013)	0.559 (+0.437)	0.818 (+0.181)
+ DANN	0.739 (+0.133)	0.829 (+0.035)	0.561 (+0.331)	0.187 (+0.266)	0.156 (+0.359)	0.582 (+0.414)	0.821 (+0.178)
+ mixup	0.739 (+0.131)	0.838 (+0.022)	0.565 (+0.321)	*n/a*	*n/a*	0.540 (+0.456)	0.818 (+0.182)

† selected before learning; exclude from objective comparisons. § not one comparison family; read the cell against its matched ERM reference at the same batch size and stopping floor, not down the column.

G Full Per-Signal Results

Tables 12, 13, and 14 give S1, S2, and S3. Each has four blocks: single environment (trained and tested on the same environment), controls, non-distributed LOO, and distributed LOO. Fig. 1 draws the latter three; the single-environment condition is a different experiment and appears only in the tables.

Every cell is the test value of the checkpoint selected by pooled source- environment validation, never by test: the val-best layer for per-layer probes and the val-best epoch for trained probes. S2 selects on validation Spearman ρ and reports Cov80, so its checkpoint optimises ranking rather than calibration. Parenthesised gaps are validation minus test; positive values indicate overfitting to the source-environment mixture. Where shown, $\pm$ is half a 95% bootstrap interval over the test set and captures sampling, not training, variation.

S1 SciQ (n_{test}=18) is reported but excluded from its mean. The single-environment block is not an upper bound: it uses one environment's data, whereas LOO pools $N-1$. On SQuAD, for example, this is 390 against 6,672 training samples; FALA-G obtains 0.765 in the former setting and 0.776 under LOO. Among the non-distributed rows, *concat-layers* adds all 33 layers to the *last-token linear* baseline, while *mean-tokens* averages tokens at its selected layer. Distributed rows learn both aggregations.

Table 12: **S1 hallucination.** Test AUROC (validation–test gap; bootstrap half-width).

Method	TriviaQA	NQ	SQuAD	CoQA	MNLI	SciQ‡	Mean
Single environment — no shift, but one environment's worth of data							
last-token linear	0.939 -0.00 ±0.021	0.838 +0.05 ±0.055	0.803 +0.04 ±0.094	0.820 +0.04 ±0.040	0.829 +0.02 ±0.044	0.975 -0.08 ±0.056	0.846 +0.03
FALA	0.896 +0.00 ±0.029	0.799 +0.03 ±0.061	0.749 -0.03 ±0.103	0.802 +0.00 ±0.042	0.826 +0.01 ±0.044	0.716 +0.06 ±0.255	0.814 +0.00
FALA-G	0.894 +0.00 ±0.029	0.831 +0.01 ±0.057	0.599 +0.11 ±0.121	0.817 +0.01 ±0.040	0.686 +0.04 ±0.058	0.778 -0.00 ±0.237	0.765 +0.04
WAP	0.918 -0.01 ±0.024	0.835 +0.04 ±0.056	0.696 +0.06 ±0.113	0.805 +0.02 ±0.042	0.811 +0.02 ±0.046	0.790 -0.12 ±0.223	0.813 +0.03
WAP-G	0.923 -0.00 ±0.025	0.837 +0.02 ±0.055	0.772 -0.00 ±0.101	0.819 +0.04 ±0.040	0.779 +0.03 ±0.049	0.753 -0.10 ±0.234	0.826 +0.02
Controls — both LOO, each differing from the last-token linear row below in one way							
MLP d'=71	0.912 -0.07 ±0.025	0.845 +0.03 ±0.054	0.724 +0.16 ±0.107	0.658 +0.23 ±0.053	0.619 +0.28 ±0.060	0.938 -0.06 ±0.100	0.751 +0.13
random-init	0.562 +0.01 ±0.050	0.536 +0.05 ±0.078	0.505 +0.07 ±0.115	0.505 +0.09 ±0.056	0.482 +0.11 ±0.062	0.383 +0.21 ±0.259	0.520 +0.07
Non-distributed, leave-one-out							
last-token linear	0.909 -0.07 ±0.026	0.843 +0.03 ±0.053	0.736 +0.14 ±0.104	0.679 +0.21 ±0.052	0.625 +0.27 ±0.061	0.914 -0.04 ±0.130	0.759 +0.12
concat-layers	0.909 -0.09 ±0.026	0.824 +0.04 ±0.059	0.793 +0.07 ±0.096	0.643 +0.25 ±0.052	0.616 +0.27 ±0.059	0.889 -0.02 ±0.156	0.757 +0.11
mean-tokens	0.917 -0.07 ±0.025	0.861 +0.02 ±0.052	0.771 +0.12 ±0.098	0.700 +0.19 ±0.051	0.637 +0.27 ±0.059	0.988 -0.10 ±0.037	0.777 +0.11
Distributed, leave-one-out							
FALA	0.843 -0.05 ±0.036	0.775 +0.07 ±0.065	0.634 +0.21 ±0.118	0.690 +0.16 ±0.051	0.545 +0.32 ±0.062	0.914 -0.07 ±0.132	0.697 +0.14
FALA-G	0.884 -0.11 ±0.030	0.847 +0.01 ±0.053	0.739 +0.12 ±0.103	0.721 +0.15 ±0.050	0.688 +0.19 ±0.056	0.901 -0.05 ±0.138	0.776 +0.07
WAP	0.896 -0.09 ±0.028	0.802 +0.06 ±0.061	0.707 +0.15 ±0.111	0.743 +0.12 ±0.046	0.530 +0.35 ±0.062	1.000 -0.14 ±0.000	0.736 +0.12
WAP-G	0.905 -0.08 ±0.027	0.830 +0.03 ±0.056	0.737 +0.14 ±0.106	0.739 +0.13 ±0.048	0.562 +0.33 ±0.062	0.963 -0.09 ±0.075	0.754 +0.11

‡ n_{test}=18; reported but excluded from the mean.

Table 13: **S2 uncertainty.** Cov80 (target 0.80; validation–test gap; bootstrap half-width).

Method	TriviaQA	NQ	SQuAD	CoQA	MNLI	Mean
Single environment — no shift, but one environment's worth of data						
last-token linear	0.017 +0.01 ±0.012	0.016 -0.01 ±0.014	0.000 +0.00 ±0.000	0.018 +0.01 ±0.012	0.024 +0.01 ±0.014	0.015 +0.00
FALA	0.000‡	0.000‡	0.000‡	0.000‡	0.186 +0.00 ±0.036	0.037 +0.00
FALA-G	0.293 -0.00 ±0.042	0.177 +0.02 ±0.043	0.044 +0.00 ±0.031	0.061 +0.01 ±0.021	0.147 -0.00 ±0.032	0.145 +0.01
WAP	0.048 -0.02 ±0.018	0.032 -0.02 ±0.021	0.006 +0.01 ±0.009	0.000 +0.01 ±0.000	0.173 +0.00 ±0.034	0.052 -0.00
WAP-G	0.050 -0.00 ±0.021	0.013 +0.01 ±0.011	0.013 -0.01 ±0.016	0.056 +0.02 ±0.021	0.050 -0.00 ±0.019	0.036 +0.00
Controls — both LOO, each differing from the last-token linear row below in one way						
MLP d'=71	0.026 +0.01 ±0.014	0.079 +0.02 ±0.030	0.044 -0.01 ±0.028	0.041 +0.01 ±0.019	0.024 +0.00 ±0.014	0.043 +0.01
random-init	0.000 +0.00 ±0.000	0.000 +0.00 ±0.000	0.000 +0.00 ±0.000	0.002 -0.00 ±0.003	0.002 +0.00 ±0.003	0.001 +0.00
Non-distributed, leave-one-out						
last-token linear	0.020 +0.01 ±0.012	0.028 -0.01 ±0.017	0.019 +0.00 ±0.022	0.014 +0.01 ±0.010	0.019 +0.01 ±0.012	0.020 +0.00
concat-layers	0.000 +0.01 ±0.000	0.006 -0.00 ±0.008	0.000 +0.01 ±0.000	0.009 -0.01 ±0.008	0.004 +0.00 ±0.005	0.004 +0.00
mean-tokens	0.028 +0.00 ±0.015	0.019 +0.01 ±0.014	0.025 +0.01 ±0.022	0.009 +0.02 ±0.008	0.028 -0.01 ±0.014	0.022 +0.01
Distributed, leave-one-out						
FALA	0.254 +0.02 ±0.039	0.225 +0.05 ±0.046	0.250 +0.05 ±0.069	0.183 +0.01 ±0.036	0.206 +0.01 ±0.037	0.224 +0.03
FALA-G	0.480 +0.01 ±0.046	0.541 +0.02 ±0.055	0.506 -0.01 ±0.075	0.384 +0.03 ±0.045	0.429 -0.01 ±0.045	0.468 +0.01
WAP	0.096 +0.00 ±0.026	0.070 +0.04 ±0.027	0.075 +0.03 ±0.041	0.084 +0.03 ±0.026	0.041 +0.01 ±0.018	0.073 +0.02
WAP-G	0.189 -0.00 ±0.036	0.095 +0.02 ±0.033	0.206 +0.04 ±0.062	0.122 +0.02 ±0.030	0.152 +0.00 ±0.031	0.153 +0.01

‡ Degenerate interval: $\text{Cov}_{80} = 0$ because $\text{width}_{80} = 6 \times 10^{-12}$–$2 \times 10^{-9}$. Read these cells as "no interval", not zero coverage.

H Representation Diagnostics

H.1 Static Representation Diagnostics

To test whether the representations each classifier reads are already environment-independent, we run three diagnostics over all seven suite folds. These are forward passes over existing checkpoints; the trace retrains, for the reason given below. The top block reports environment predictability at native width and after projection to a common width, so the comparison does not reward the wider feature. The bottom block reports the cross-environment transfer gap: one linear head per environment, every head evaluated on every environment, both diagonal and off-diagonal read on held-out halves. Figures 4, 5, and 6 give the same three diagnostics fold by fold.

Table 14: **S3 training-data familiarity** (OLMo-3-1025-7B), with 300 source training samples per domain. Test AUROC (validation–test gap; bootstrap half-width).

Method	wiki	arXiv	CommonCrawl	StackEdu	Mean
Single environment — no shift, but one environment's worth of data					
last-token linear	0.996 +0.00 ±0.004	1.000 -0.00 ±0.000	0.974 +0.01 ±0.015	0.926 -0.01 ±0.028	0.974 +0.00
FALA	1.000 +0.00 ±0.000	0.999 +0.00 ±0.002	0.998 +0.00 ±0.002	0.963 +0.01 ±0.017	0.990 +0.00
FALA-G	1.000 -0.00 ±0.001	0.984 +0.01 ±0.011	0.996† +0.00 ±0.004	0.956 +0.00 ±0.020	0.984 +0.00
WAP	1.000 -0.00 ±0.000	1.000 +0.00 ±0.001	1.000 +0.00 ±0.001	0.960 +0.01 ±0.018	0.990 +0.00
WAP-G	1.000 +0.00 ±0.000	1.000 +0.00 ±0.001	1.000 +0.00 ±0.000	0.980 -0.00 ±0.012	0.995 +0.00
Controls — both LOO, each differing from the last-token linear row below in one way					
MLP d'=71	0.838 +0.13 ±0.048	0.328 +0.64 ±0.062	0.379 +0.60 ±0.064	0.721 +0.27 ±0.058	0.567 +0.41
random-init¶	0.778 +0.16 ±0.053	0.705 +0.23 ±0.060	0.862 +0.07 ±0.041	0.513 +0.48 ±0.065	0.714 +0.24
Non-distributed, leave-one-out					
last-token linear	0.816 +0.15 ±0.049	0.355 +0.61 ±0.064	0.516 +0.46 ±0.066	0.760 +0.23 ±0.055	0.612 +0.36
concat-layers	0.797 +0.17 ±0.051	0.298 +0.66 ±0.060	0.646 +0.33 ±0.063	0.640 +0.34 ±0.063	0.595 +0.37
mean-tokens	0.710 +0.28 ±0.062	0.541 +0.45 ±0.066	0.695 +0.30 ±0.059	0.760 +0.24 ±0.054	0.676 +0.32
Distributed, leave-one-out					
FALA	0.499 +0.49 ±0.068	0.509 +0.48 ±0.064	0.719 +0.28 ±0.057	0.663† +0.34 ±0.061	0.598 +0.40
FALA-G	0.800 +0.17 ±0.051	0.470 +0.52 ±0.067	0.777 +0.22 ±0.053	0.759 +0.24 ±0.055	0.702 +0.28
WAP	0.653 +0.34 ±0.065	0.439 +0.55 ±0.064	0.773 +0.22 ±0.055	0.668 +0.33 ±0.062	0.633 +0.36
WAP-G	0.644 +0.35 ±0.064	0.582 +0.41 ±0.065	0.681 +0.32 ±0.060	0.822 +0.18 ±0.048	0.682 +0.32

¶ Surface-form control, not a chance baseline: S3 labels describe the text, so a random network can separate seen from unseen corpora. † The minimum-epoch floor extended this run from its natural stop at epoch 7 to epoch 14; its selected checkpoint was unchanged.

H.2 Training Trajectories

The trajectory diagnostic (Fig. 3) retrains ERM, Group-DRO, and DANN with `diag_every= 1` on seven suite folds and two distributed hosts (42 runs). At every epoch, it freezes the current representation and recomputes the two diagnostics on validation data from the source environments. Environment predictability is the accuracy of an auxiliary environment probe. For lack of invariance, each source environment is split in half: a linear head is fit on one half and evaluated on every environment's other half; the reported quantity is the mean source–target optimal-head discrepancy over ordered environment pairs.

I Diversity Versus Quantity

Here we ask whether a *marginal* budget is better spent on more labels from an existing environment or on a new environment. From a starting environment A at 100 labels, step t trains on $t \times 100$ labels in one of two ways: **quantity**, A alone with more of it, or **diversity**, t environments at 100 labels each. Both arms hold the total equal at every step, so they differ only in where the marginal labels came from. We run five held-out folds $\times$ two starting environments (env3_coqa and env4_mnli), four steps each, on FALA-G; Figure 7 shows both arms.

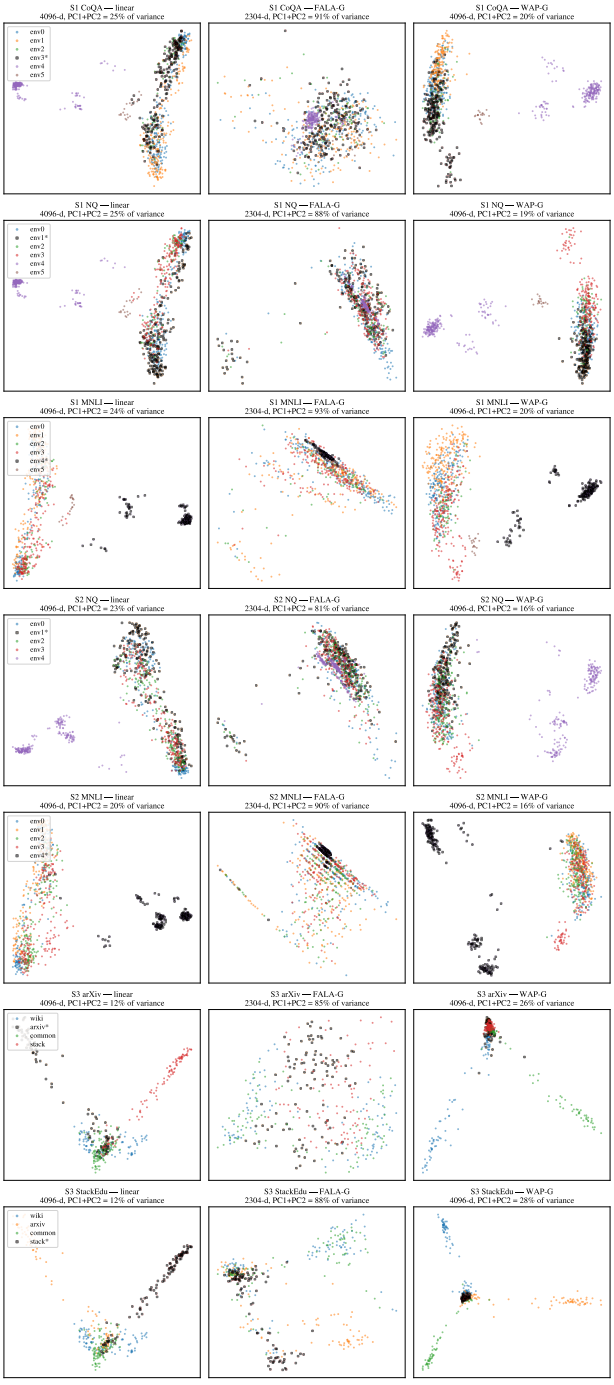

Figure 4: **D1: the representation each classifier reads, projected to two dimensions and colored by environment** (X markers: the held-out environment). Rows are the seven suite folds, columns the three representations. The middle column (FALA-G) mixes environments more visibly than the other two. Each panel prints its explained variance; a 2D projection can only *lower-bound* separability, so the classifier in D2 is the load-bearing measurement.

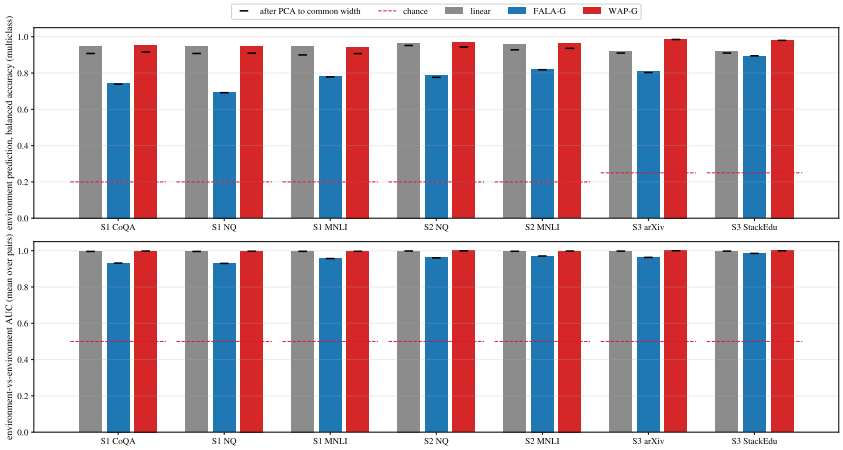

Figure 5: **D2: is environment information present in the representation?** Across all seven suite folds, bars are native width and black ticks the common-width control; a conclusion needs both.

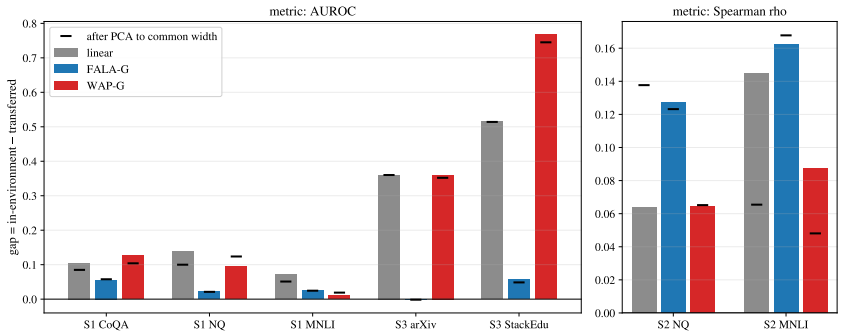

Figure 6: **D3: is the optimal classifier the same in every environment?** Across all seven suite folds, smaller is more invariant. The two panels use different metrics and are never averaged.

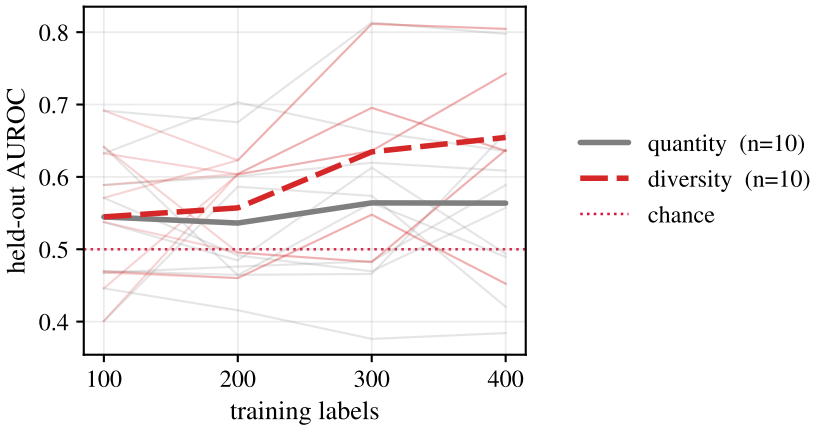

Figure 7: **Diversity versus quantity at equal budget.** Held-out AUROC against the number of training labels; faint lines are the ten (fold, start) ladders, bold their mean. Both arms spend the same budget at every step and are the same run at $t{=}1$.