GNOSIS: CONSISTENCY-AWARE PROBES FOR SELECTIVE PREDICTION OVER INTERNAL STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Selective prediction determines whether an LLM's output should be trusted or deferred, which is critical for building reliable LLM systems. Existing confidence estimators fall into three groups: multi-sample methods requiring repeated generation, post-generation scoring over fully decoded outputs, and pre-generation hidden-state probes. While pre-generation probes yield confidence estimation before token generation and avoid decoding overhead, existing approaches rely on a narrow readout paradigm: they typically preselect a single readout layer and optimize against only one training objective, limiting their overall performance. We present GNOSIS, a suite of lightweight single-pass pre-generation probes which operates on the prompt-final hidden state of a frozen LLM. It comprises three core designs: (1) The Direct Correctness Probe (DCP) infers answer correctness and adaptively selects the optimal readout layer via cross-validation instead of manual specification. (2) Anchor-Regression Distillation (ARD) distills multi-dimensional cross-model consistency signals from a stronger teacher model into student hidden states, where the teacher is only used for target construction and is never invoked during inference. (3) A Fusion module jointly optimizes both probe heads on a shared trunk to exploit complementary error patterns. Across 11 LongBench tasks spanning eight base models (3B-72B), GNOSIS achieves competitive or state-of-the-art AUROC and PRR. When deployed as an out-of-distribution HotpotQA cascade router, GNOSIS allows escalated queries to fully bypass student-model generation, substantially lowering latency, token consumption, and API cost. Ablations confirm that DCP and ARD capture distinct failure modes and the joint multi-objective Fusion outperforms naive post-hoc score averaging. Our results demonstrate that adaptive readout-layer selection together with multi-faceted training targets is essential for high-performance pre-generation selective prediction.

1 INTRODUCTION

Large language models (LLMs) have served as general-purpose assistants, yet their outputs remain unevenly reliable: fluent responses often contain unsubstantiated or factually incorrect claims. This makes *selective prediction*, the task of estimating whether a generated answer should be trusted, a central primitive for trustworthy LLM deployment. In single-model settings, a correctness predictor supports hallucination detection and abstention, improving reliability at a target coverage (Xin et al., 2021; Varshney et al., 2022; Wen et al., 2025). In *cascade* settings, the predictor acts a routing gate: confident queries can be handled by a cheaper student model, while uncertain queries are forwarded to a stronger teacher model, balancing computational cost and task performance (Geifman & El-Yaniv, 2017; Madras et al., 2018; Mozannar & Sontag, 2020; Kamath et al., 2020).

These deployment needs have motivated various methods for estimating LLM reliability, which generally fall into two families based on inference cost. *Multi-sample* methods such as SelfCheckGPT (Manakul et al., 2023), semantic entropy (Kuhn et al., 2023), and lexical-similarity scores (Lin et al., 2023) rely on multiple stochastic generations to measure prediction disagreement. While effective, they often incur heavy computational cost. *Single-pass* methods require only one forward pass but differ in when the signal becomes available. Post-generation approaches score completed answers using output probabilities (Hendrycks & Gimpel, 2016) or self-evaluation, but they must run full decoding before routing and collapse internal model uncertainty into hand-engineered statistics.

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.

In contrast, *pre-generation* probes including P(IK) (Kadavath et al., 2022), SAPLMA (Azaria & Mitchell, 2023), and SEPs (Kossen et al., 2024) extract signals from hidden states before decoding, supporting zero-overhead early deferral (Slobodkin et al., 2023; Gottesman & Geva, 2024; Orgad et al., 2025). Despite their appeal, existing pre-generation probes are limited by a restrictive readout design: they fix readout layers manually and optimize only a single correctness objective. This design overlooks a critical failure mode in cascade routing: cases where the student model produces stable yet incorrect answers. Such samples are hard to identify using only the model's own self-knowledge. To address this gap, we ask: **How can we adaptively extract both model self-knowledge and teacher-aligned consistency from pre-generation hidden states, improving selective prediction without adding inference-time overhead?**

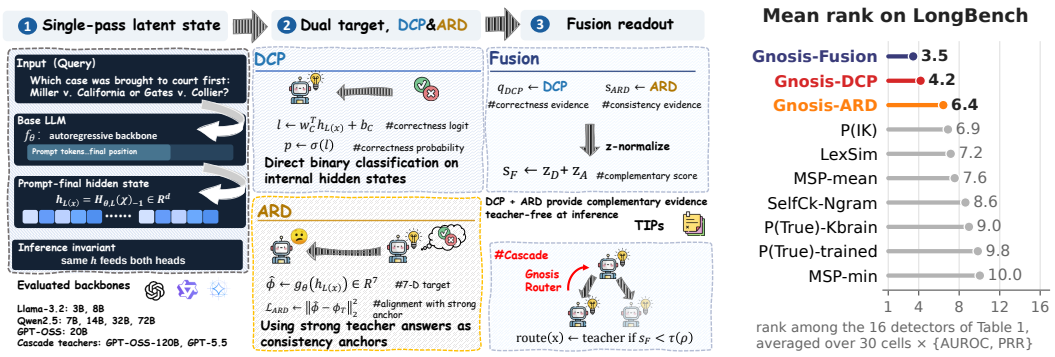
In this paper, We present GNOSIS, a set of lightweight single-pass probes that leverage the final prompt hidden state as a substrate encoding both intended answers and latent uncertainty. As summarized in Figure 1(a), GNOSIS comprises three core designs. First, the **Direct Correctness Probe (DCP)** predicts answer correctness directly from hidden states, adaptively selecting the optimal readout layer via cross-validation. Second, instead of performing conventional feature distillation, **Anchor-Regression Distillation (ARD)** regresses multi-dimensional consistency targets that capture semantic cluster support and agreement with a stronger teacher model. Crucially, the teacher model is only used to construct training targets and is never invoked at inference time. Finally, the **Fusion** module combines correctness-focused and consistency-focused readouts via a shared dual-head trunk, exploiting their complementary error patterns to produce a unified routing score.

We evaluate GNOSIS in two rigorous settings: threshold-free selective prediction across 11 LongBench tasks with eight base models (3B to 72B), and out-of-distribution cascade routing on HotpotQA (Qwen2.5-7B to GPT-OSS-120B/GPT-5.5). GNOSIS achieves competitive or superior performance compared to strong baselines (Figure 1(b); Tables 1–3). Since GNOSIS scores are available before decoding, escalated queries entirely skip the student's generation phase, yielding meaningful reductions in latency, token consumption and API cost relative to post-generation approaches (Figure 2). Our contributions are summarized as follows:

- **Representation insight for pre-generation selective prediction.** We reveal that the final prompt hidden states of frozen LLMs encode both self-knowledge of answer correctness and latent alignment with stronger teacher models. Unlike prior work that only estimates self-correctness, we demonstrate that both signals can be jointly decoded from this single pre-decoding representation, eliminating the need to query the teacher during inference.
- **GNOSIS: lightweight pre-generation probing framework.** We propose GNOSIS, a single-pass probe suite consisting of adaptive DCP for correctness prediction, teacher-anchored ARD for multi-dimensional consistency regression, and a shared dual-head fusion trunk. GNOSIS unifies correctness and consistency estimation without introducing any additional compute overhead at inference time.
- **Empirical insight for cascade routing.** Through evaluations on LongBench and out-of-distribution HotpotQA cascade routing, we verify that DCP and ARD exhibit complementary failure modes. ARD can detect stable yet incorrect outputs that are missed by conventional self-correctness probes, and our joint fusion strategy consistently outperforms ensembles of independently trained scores.

2 Related work

Uncertainty and self-evaluation for LLM factuality. A natural approach to hallucination detection is to treat the model's own confidence as a proxy for factuality (Huang et al., 2025). Kadavath et al. (2022) introduce *P(IK)*, a calibrated scalar trained on top of a language model's hidden states to predict whether the model will answer a question correctly, and show that the model's self-evaluation of an answer, *P(True)*, is also a useful signal on multi-choice tasks. Building on the observation that hallucinations often co-occur with inconsistent answers across samples, Manakul et al. (2023) propose *SelfCheckGPT*, which measures sample-sample agreement using a suite of lexical, NLI, and question-answering scorers. Kuhn et al. (2023) instead cluster semantically equivalent samples and compute the entropy of the resulting distribution, and Farquhar et al. (2024) show that this *semantic entropy* improves detection of confabulations across several model families. A key practical limitation shared by SelfCheckGPT and semantic entropy is their requirement for multiple stochastic



(a) (b)

Figure 1: **GNOSIS overview and LongBench ranking.** (a) A frozen LLM produces one prompt-final hidden state, which feeds two probe heads: DCP predicts correctness, and ARD regresses a seven-dimensional teacher-anchor representation. Fusion trains both heads on a shared trunk and adds their standardized scores for ranking and cascade routing. Teacher anchors serve only as a training target, so inference requires neither a teacher call nor additional sampling. (b) Mean rank on LongBench (lower is better). Gnosis-DCP, Gnosis-ARD, and Gnosis-Fusion correspond to DCP-MLP, ARD-MLP, and Fusion-combo in Table 1; DCP-LR, ARD-Ridge, and the four lowest-ranked baselines are not shown.

generations per query, which hinders deployment under strict latency constraints. Related work also examines whether LLMs can express their uncertainty directly (Xiong et al., 2024).

Hidden-state probes for hallucination. A complementary line of work reads the model's internal activations directly (Slobodkin et al., 2023; Gottesman & Geva, 2024; Orgad et al., 2025; Ferrando et al., 2025). Burns et al. (2023) show that, for binary truthfulness probes, a linear or MLP head over hidden states can recover a model's latent knowledge without any labels (*CCS*), establishing that factual beliefs are linearly represented (Marks & Tegmark, 2023). Chen et al. (2024) demonstrate that the prompt-final hidden state alone is competitive with multi-sample baselines on short-form QA, and propose *EigenScore*, a spectral consistency signal computed from the hidden states of multiple samples. Liu et al. (2024) show that pre-generation hidden states indicate hallucination risk on 15 NLG tasks and 700+ datasets, and that a small MLP head on the last-token state can be transferred across tasks. Su et al. (2024) use the internal states of LLMs for unsupervised, *real-time* hallucination detection. Most recently, Zhang et al. (2025) track cross-layer *Information Contribution to the Residual stream* (ICR) and Han et al. (2025) show that even a single linear feature on the residual stream can be competitive with expensive multi-sample detectors.

Selective prediction and cost-aware LLM cascades. Selective prediction formalizes the idea of abstaining on low-confidence inputs (Madras et al., 2018; Mozannar & Sontag, 2020). El-Yaniv & Wiener (2010) characterize the risk–coverage trade-off of selective classification in the noise-free setting, and Geifman & El-Yaniv (2017) extend selective classification to deep networks, rejecting inputs so that a target risk is met with high probability. The prediction–rejection ratio (Malinin & Gales, 2020) summarizes such rejection curves in a single score. For LLMs, these ideas have been instantiated as *cascades* that route a query from a cheap student to an expensive teacher when the student is unlikely to answer correctly. Chen et al. (2023) propose a learned router over a cascade of APIs; Aggarwal et al. (2024) automatically mix candidate models; Yue et al. (2024) study cascades with mixture-of-thoughts representations; Ong et al. (2025) learn routing from preference data; and Trapeznikov & Wang (2024) emphasize *calibration* of the router's confidence to obtain cost-optimal thresholds.

3 METHOD

Setup. Given an input prompt x, let $\mathbf{h}_\ell(x) \in \mathbb{R}^d$ denote the prompt-final residual hidden state at layer ℓ of a frozen LLM, extracted before any answer tokens are decoded. Let $y \in \{0, 1\}$ denote whether the model's final answer is correct under the task metric. GNOSIS trains lightweight probes

on $\mathbf{h}_\ell(x)$ to produce a reliability score $s(x)$, where larger values indicate higher confidence. All GNOSIS variants are evaluated under the same layer-selection protocol: for each candidate layer $\ell \in \mathcal{L}$, we train the probe with question-level cross-fitting and select the optimal readout layer maximizing out-of-fold AUROC ($\mathrm{AUROC}_{\mathrm{OOF}}$):

$$\ell^\star = \arg\max_{\ell \in \mathcal{L}} \mathrm{AUROC}_{\mathrm{OOF}} \left(\{s(x_i; \ell)\}_{i=1}^n, \{y_i\}_{i=1}^n \right). \tag{1}$$

The final score is $s(x) = s(x; \ell^\star)$. This protocol keeps the base LLM frozen, prevents question leakage across folds, and isolates the effect of the readout objective.

3.1 Correctness and Consistency Readouts

Direct Correctness Probe (DCP). We first establish a correctness-only baseline to assess how well the prompt-final hidden state predicts the binary outcome. DCP is defined as:

$$s_{\mathrm{DCP}}(x; \ell) = p_\theta(y = 1 \mid \mathbf{h}_\ell(x)) = \sigma\Big(f_\theta(\widetilde{\mathbf{h}}_\ell(x)) \Big), \tag{2}$$

where $\widetilde{\mathbf{h}}_\ell(x)$ is the standardized hidden state and f_θ is a lightweight head (logistic regression or a small MLP) trained with binary cross-entropy on correctness labels.

DCP provides a reference for testing whether the teacher-derived targets introduced next improve on correctness-only supervision.

Anchor-Regression Distillation (ARD). Binary correctness labels only record final outcome, and cannot capture risky failure patterns: answers internally stable for the student yet unsupported by stronger teacher models. These distinctions may help identify errors that appear reliable under student self-consistency alone. ARD therefore trains a probe to predict a teacher-derived target describing agreement between the student's answer and sampled teacher answers. Teacher answers and student generations are used to construct training targets; at inference, the probe predicts this target from the student's prompt-final hidden state.

Specifically, for each training prompt x_i, we draw K stochastic answers from a stronger teacher, $\mathcal{A}_i^T = \{a_{i,1}^T, \ldots, a_{i,K}^T\}$. We group these answers into *teacher answer basins*, i.e., equivalence classes induced by a canonical answer form $c(\cdot)$:

$$a \sim_c a' \iff c(a) = c(a'), \qquad \mathcal{B}_i^T = \mathcal{A}_i^T / \sim_c = \{B_{i,1}^T, \ldots, B_{i,M_i}^T\}, \qquad m_{i,j}^T = \frac{|B_{i,j}^T|}{K}. \tag{3}$$

Here $m_{i,j}^T$ is the empirical mass of teacher basin $B_{i,j}^T$. In our implementation, basins are obtained by canonical answer grouping, and student-teacher alignment is computed by answer similarity rather than by an additional NLI model.

Let $\hat{a}_i^S$ be the student's generated answer during training, and let $r_{i,j}^T$ be a representative answer from teacher basin $B_{i,j}^T$. ARD computes teacher support for the student answer as

$$s_i^{\max} = \max_j \mathrm{sim}\big(\hat{a}_i^S, r_{i,j}^T\big), \qquad m_i^{\sup} = \sum_{j=1}^{M_i} m_{i,j}^T \mathbf{1}\big[\mathrm{sim}\big(\hat{a}_i^S, r_{i,j}^T\big) \geq \tau_{\mathrm{align}}\big]. \tag{4}$$

Together with the mass and correctness of the best-matched teacher basin, the teacher-correct support mass, a student-stability penalty, and a lightweight confidence proxy, these quantities constitute our proposed seven-dimensional teacher-anchor target, which summarizes cross-model consistency:

$$\Phi_T(x_i) = \big[s_i^{\max}, m_i^{\max}, c_i^{\max}, m_i^{\sup}, m_i^{\mathrm{corr}}, m_i^{S\setminus T}, a_i\big] \in \mathbb{R}^7. \tag{5}$$

The final coordinate a_i is the scalar anchor score used for ranking. The detailed definitions of these seven dimensions are provided in Appendix A.

ARD trains a probe to amortize this teacher-anchor target into the frozen student's hidden state:

$$g_\theta^{\mathrm{ARD}}(\mathbf{h}_\ell(x_i)) = \widehat{\Phi}_T(x_i), \qquad \theta^\star = \arg\min_\theta \frac{1}{n} \sum_{i=1}^n \big\| g_\theta^{\mathrm{ARD}}(\mathbf{h}_\ell(x_i)) - \Phi_T(x_i) \big\|_2^2. \tag{6}$$

We instantiate g_θ^{ARD} as either multi-output ridge regression (ARD-Ridge) or an MLP regression head (ARD-MLP). The ARD confidence score is the predicted anchor-score coordinate,

$$s_{\mathrm{ARD}}(x;\ell) = \left[g_{\theta^\star}^{\mathrm{ARD}}(\mathbf{h}_\ell(x))\right]_a . \tag{7}$$

ARD is distillation only in a restricted sense. It amortizes a teacher-derived reliability target into a lightweight probe, but it does not update the student LLM or imitate teacher logits or text, and the teacher is never called at test time.

3.2 Fusion Probe

DCP and ARD expose complementary views of the same prompt-conditioned state. DCP provides a decision-aligned correctness readout, while ARD provides a consistency-aligned readout of teacher support. The complementarity is also observed empirically. In our fusion ablation, combining the correctness and anchor readouts improves ranking over either readout alone, which suggests that the anchor target carries information beyond the correctness label. Fusion therefore combines the two objectives in a single multi-task probe.

Given $\mathbf{h}_\ell(x)$, Fusion maps the standardized hidden state through a shared MLP trunk and attaches two heads:

$$\mathbf{z}_\psi(x;\ell) = q_\psi\left(\widetilde{\mathbf{h}}_\ell(x)\right), \qquad s_{\mathrm{cls}}(x;\ell) = \sigma(\mathbf{w}_c^\top \mathbf{z}_\psi(x;\ell) + b_c), \qquad \widehat{\Phi}_T^{\mathrm{fus}}(x;\ell) = W_a \mathbf{z}_\psi(x;\ell) + \mathbf{b}_a . \tag{8}$$

The shared trunk is trained with a joint objective:

$$\mathcal{L}_{\mathrm{Fusion}} = \mathcal{L}_{\mathrm{BCE}}(s_{\mathrm{cls}}(x;\ell), y) + \lambda \left\| \widehat{\Phi}_T^{\mathrm{fus}}(x;\ell) - \Phi_T(x) \right\|_2^2 . \tag{9}$$

where $\lambda = 1$ in our implementation, and the anchor target is standardized within each fold.

At inference, Fusion combines the correctness head and the predicted anchor-score coordinate as ranking evidence. Let $\mathcal{Z}(\cdot)$ denote score standardization to zero mean and unit variance on the evaluated split. The fused score is

$$s_{\mathrm{Fusion}}(x;\ell) = \mathcal{Z}(s_{\mathrm{cls}}(x;\ell)) + \mathcal{Z}\left(\left[\widehat{\Phi}_T^{\mathrm{fus}}(x;\ell)\right]_a\right). \tag{10}$$

Unlike a post-hoc ensemble of separately trained probes, Fusion learns discriminative correctness evidence and teacher-anchored consistency jointly in a shared representation, and it retains the single-pass, pre-generation inference profile of DCP and ARD.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate GNOSIS in two settings: selective prediction on LongBench and out-of-distribution cascade routing on HotpotQA. LongBench (Bai et al., 2024) is a bilingual long-context benchmark with 11 tasks spanning QA, summarization, and few-shot classification. The panel covers eight base models from 3B to 72B parameters (Llama-3.2-3B, Llama-3.1-8B, Gemma-2-27B, Qwen2.5-7B, -14B, -32B, and -72B, and GPT-OSS-20B) and 16 detectors. Within each $(\mathrm{model}, \mathrm{task})$ cell, all detectors use the same 200 questions, generated answers, and official metric; cells with no correct answers are excluded. Table 1 and Appendix E report per-task results for all eight models, and Appendix B reports task-native utility for Llama-3.1-8B, Qwen2.5-7B, and Qwen2.5-72B. For cascade deployment, gates are trained on TriviaQA (Joshi et al., 2017) and evaluated without adaptation on HotpotQA (Yang et al., 2018) ($n = 500$), where a Qwen2.5-7B student routes uncertain queries to GPT-OSS-120B or GPT-5.5.

Metrics. Let $y \in \{0, 1\}$ denote the binary correctness label obtained by thresholding the benchmark's official score per task, $y_i = \mathbf{1}[M_t(\hat{a}_i, a_i^\star) \geq \tau_t]$; for example, QA tasks use token-level F1 ≥ 0.5 and summarization tasks use ROUGE-L ≥ 0.3. For a detector score $s(x)$, with larger values indicating higher confidence, our selective-prediction metric is **Lift@c**, the gain in the official task score when only the top-$c\%$ most confident predictions are retained:

$$\mathrm{Lift@}c \;=\; M(\{x : s(x) \text{ ranks in the top } c\%\}) - M(\mathrm{all}), \tag{11}$$

reported at $c \in \{25, 50, 75\}$. We complement Lift with **AUROC**, which measures how well s separates correct from incorrect predictions, and **PRR** (Malinin & Gales, 2020), which summarizes the prediction–rejection curve and is normalized so that 0 matches random and 1 matches oracle ranking. For cascades, $A_2(e)$ is the system accuracy when the lowest-confidence fraction e of queries is escalated in the two-tier setting, and $A_3(e)$ is the corresponding accuracy in a three-tier setting, in which non-escalated queries in the middle tercile of the gate score are answered by the student under contrastive activation addition (Rimsky et al., 2024), detailed in Appendix D. We summarize these curves by $\mathbf{AUC}_2$ and $\mathbf{AUC}_3$; **route AUROC** measures how well the gate score orders queries for escalation, and $\mathbf{steer}\Delta = \mathrm{AUC}_3 - \mathrm{AUC}_2$ is the gain from this steering tier. **vs-Rand** subtracts the AUC of random routing with the same student/teacher marginals, and **Pareto-front%** is the fraction of escalation budgets at which no other gate has both higher accuracy and lower cost. We also report savings in wall-clock time, generated tokens, and API cost (**save-time%**, **save-token%**, and **save-$%**) relative to always calling the teacher, measured at the teacher-level accuracy point and counted end to end over the student and teacher passes, so that a pre-generation gate is credited for skipping the student decode on escalated queries.

Implementation Details. All base LLMs are kept frozen and run in bf16. For each hidden-state method we extract the prompt-final residual state at 7 to 10 candidate layers spread over depth and select the layer by out-of-fold AUROC, following the protocol of Section 3. All probes use five-fold cross-fitting with question-level groups. DCP-MLP and ARD-MLP are two-layer MLPs, the latter with a seven-dimensional output, and ARD-Ridge fits one ridge regressor per anchor dimension; both ARD variants score a query by the predicted anchor coordinate as in Eq. (7). Fusion uses the same MLP trunk with a correctness head and an anchor head. We group the baselines as in Section 1. The pre-generation probes P(IK), SAPLMA, and SEPs read the prompt-final hidden state, as GNOSIS does. The post-generation scores need the decoded answer: MSP, P(True) on the greedy answer, and HaMI (Niu et al., 2026), which reads the hidden states of the generated tokens. The multi-sample methods need K sampled answers: lexical similarity, SelfCheckGPT, semantic entropy, P(True)-Kbrain, which shows the samples to the model before self-evaluation, and P(True)-trained, a logistic regression over P(True) and sampling-based scores. We use $K{=}4$ on LongBench, the same budget used to construct the anchor targets, and $K{=}8$ student samples in the cascade experiments, where escalation is rank based and each budget routes the lowest-confidence fraction of queries to the teacher. The cascade ARD gates differ from the LongBench probes in one respect: they read the correctness score from the predicted seven-dimensional anchor vector with a logistic regression fitted on the TriviaQA training set, rather than from the anchor coordinate alone.

4.2 Comparison with SOTA

We compare GNOSIS against pre-generation hidden-state probes, including P(IK), SAPLMA, and SEPs (Kadavath et al., 2022; Azaria & Mitchell, 2023; Kossen et al., 2024); multi-sample uncertainty estimators, including SelfCheckGPT and semantic entropy (Manakul et al., 2023; Kuhn et al., 2023; Farquhar et al., 2024); and post-generation scores, including the HaMI detector (Niu et al., 2026). The comparison asks whether a score ranks reliable generations well offline and whether it also yields favorable end-to-end behavior as a cascade gate.

LongBench selective prediction. Table 1 reports AUROC and PRR, which assess ranking quality without committing to an operating point. The GNOSIS probes account for roughly two thirds of the best and second-best entries across the three model families, but no single head dominates: DCP-MLP and DCP-LR contribute most of the GNOSIS wins, whereas ARD and Fusion add gains on tasks where the consistency target changes the ordering of predictions. DCP-LR differs from P(IK) only in its selected readout layer and matches or exceeds it in every AUROC cell, so layer selection alone already helps. To relate ranking quality to what a user gains in practice, we also compute Lift@c for $c \in \{25, 50, 75\}$, the change in each task's official LongBench score when only the $c\%$ most confident predictions are answered; the per-task results for Llama-3.1-8B, Qwen2.5-7B, and Qwen2.5-72B are given in Appendix B (Figure 4). Counting best and runner-up placements over all model–task–coverage cells, Fusion ranks first (35), ahead of DCP-LR (27) and DCP-MLP (26), as one would expect if the correctness and consistency readouts carry complementary ranking information. The advantage is not uniform, however: the lexical-similarity and MSP baselines remain competitive on several summarization and short-answer tasks.

Table 1: **Selective prediction on LongBench.** Each cell reports AUROC / $\text{PRR}_{\leq 50\%}$; higher is better. Bold and underline indicate the best and second-best results within each model–task–metric combination. ⋆ denotes GNOSIS variants; "–" indicates unavailable or undefined results.

		single-doc QA			multi-doc QA			summary		classification	short-QA	
	Method	NarrQA	Qasper	MultiFQA	HotpQA	2WikiM	MuSiQue	GovRep	MultiN	TREC	TrivQA	SAMSum
Llama-8B												
	PTrue-trained	0.67/0.27	0.55/0.09	0.49/0.11	0.63/0.18	0.59/0.21	0.60/0.25	0.71/0.51	0.65/0.28	0.67/0.47	0.80/0.45	0.71/0.45
	PTrue-Kbrain	0.69/0.23	0.58/0.18	0.63/0.22	0.68/0.29	0.59/0.15	0.62/0.25	0.55/0.08	0.66/0.34	0.66/0.38	0.81/0.51	0.73/0.35
Multi-sample	LexSim	0.75/0.40	0.71/0.36	0.71/**0.41**	0.69/0.39	0.64/0.27	0.62/0.36	0.71/0.43	**0.70/0.46**	0.58/0.21	0.79/0.52	0.75/0.49
	SelfCk-Ngram1-avg	0.79/0.36	0.71/0.39	0.66/0.37	0.68/0.32	0.57/0.13	0.65/0.44	0.60/0.26	0.61/0.30	0.70/**0.55**	0.71/0.43	0.84/0.69
	SelfCk-NLI-avg	0.59/0.24	0.61/0.27	0.57/0.14	0.59/0.22	0.61/0.22	0.54/0.16	0.45/-0.02	0.63/0.18	0.67/0.42	0.72/0.38	0.77/0.44
	SemanticEntropy	0.50/0.08	0.50/0.04	0.49/-0.03	0.56/0.10	0.56/0.18	0.54/0.13	0.52/-0.06	0.54/0.08	0.58/0.31	0.52/-0.03	0.52/-0.03
Post-gen.	MSP-mean	0.71/0.30	0.62/0.15	0.63/0.23	0.65/0.36	0.61/0.25	0.63/0.33	0.70/0.40	0.64/0.30	0.62/0.25	0.86/**0.64**	0.46/-0.04
	MSP-min	0.70/0.25	0.61/0.13	0.62/0.17	0.63/0.30	0.57/0.15	0.61/0.23	0.65/0.37	0.64/0.28	0.61/0.20	0.83/0.61	0.47/-0.02
Pre-gen.	P(IK)	0.77/0.35	0.76/0.40	0.64/0.28	0.68/0.33	0.67/0.35	0.76/0.60	0.74/0.39	0.58/0.30	0.57/0.18	0.83/0.49	0.91/0.72
	SAPLMA	0.67/0.23	0.52/0.00	0.57/0.20	0.65/0.26	0.54/0.12	0.61/0.31	0.48/0.06	0.52/-0.01	0.52/0.10	0.76/0.50	0.69/0.35
	SEPs-Ridge	0.60/0.12	0.63/0.17	0.56/0.13	0.66/0.32	0.64/0.28	0.57/0.12	0.51/0.03	0.53/-0.00	0.53/0.13	0.56/0.17	0.57/0.09
	DCP-MLP ⋆	0.83/0.41	0.76/**0.41**	0.69/0.37	0.68/0.34	**0.70/0.42**	**0.79/0.67**	0.81/0.51	0.69/0.45	0.67/0.34	0.87/0.53	**0.92/0.73**
	DCP-LR ⋆	**0.84/0.43**	0.76/0.40	0.69/0.35	0.68/0.33	0.68/0.41	0.76/0.60	**0.82/0.51**	0.69/0.44	**0.73**/0.50	0.85/0.57	0.92/0.72
	ARD-MLP ⋆	0.72/0.29	0.72/0.33	0.70/0.33	0.70/0.34	0.62/0.29	0.64/0.37	0.72/0.39	0.68/0.31	0.57/0.17	**0.87**/0.56	0.77/0.58
	ARD-Ridge ⋆	0.71/0.31	0.71/0.31	0.68/0.28	0.72/0.38	0.60/0.24	0.65/0.32	0.72/0.28	0.64/0.29	0.54/0.15	0.85/0.56	0.74/0.41
	Fusion-combo ⋆	0.80/0.32	**0.77**/0.37	**0.74**/0.37	**0.74/0.41**	0.67/0.37	0.72/0.48	0.81/0.49	0.67/0.40	0.66/0.35	0.86/0.60	0.87/0.64
Gemma-2-27B												
	PTrue-trained	0.65/0.34	0.67/0.31	0.58/0.29	0.74/**0.48**	0.57/0.25	0.57/0.36	–	–	0.73/0.46	0.39/0.01	0.58/0.31
	PTrue-Kbrain	0.72/0.42	0.66/0.16	0.60/0.24	0.71/0.40	0.57/0.13	0.61/0.28	–	–	0.60/0.30	0.77/0.50	0.53/0.17
Multi-sample	LexSim	0.72/**0.54**	0.66/0.29	0.57/0.24	0.61/0.23	0.56/0.15	0.62/0.46	–	–	0.79/0.56	0.78/0.39	0.64/0.39
	SelfCk-Ngram1-avg	0.72/0.46	0.58/0.13	0.49/0.16	0.57/0.11	0.48/-0.03	0.52/0.21	–	–	0.79/0.51	0.74/**0.54**	**0.66/0.47**
	SelfCk-NLI-avg	–	–	–	–	–	–	–	–	–	–	–
	SemanticEntropy	0.61/0.40	0.56/0.12	0.51/0.12	0.60/0.21	0.52/0.08	0.60/0.31	–	–	0.56/0.09	0.47/-0.12	0.50/0.04
Post-gen.	MSP-mean	0.71/0.46	0.73/0.42	0.64/0.30	0.67/0.32	0.57/0.13	0.63/0.44	–	–	0.70/0.33	0.63/0.22	0.61/0.33
	MSP-min	0.69/0.36	0.69/0.29	0.63/0.25	0.66/0.31	0.57/0.11	0.62/0.38	–	–	0.59/0.07	0.62/0.20	0.62/0.31
Pre-gen.	P(IK)	0.73/0.34	0.72/0.40	0.63/0.24	0.70/0.43	0.66/0.35	0.67/0.41	–	–	0.87/0.66	0.75/0.32	0.52/0.03
	SAPLMA	0.63/0.22	0.69/0.36	0.61/0.31	0.67/0.28	0.67/0.36	0.64/0.33	–	–	0.62/0.17	0.77/0.38	0.59/0.32
	SEPs-Ridge	0.67/0.30	0.60/0.12	0.59/0.14	0.71/0.36	0.69/0.36	0.62/0.38	–	–	0.62/0.22	0.65/0.21	0.55/-0.01
	DCP-MLP ⋆	0.75/0.38	0.73/0.40	**0.66/0.32**	0.74/0.46	0.72/0.47	0.70/**0.48**	–	–	0.86/0.67	0.63/0.18	0.60/0.36
	DCP-LR ⋆	**0.75**/0.39	**0.74/0.43**	0.64/0.24	0.73/0.44	**0.73/0.50**	0.70/0.47	–	–	0.87/0.66	0.75/0.32	0.62/0.29
	ARD-MLP ⋆	0.71/0.42	0.65/0.28	0.62/0.26	0.73/0.38	0.67/0.39	0.65/0.39	–	–	0.88/0.66	0.87/0.49	0.60/0.34
	ARD-Ridge ⋆	0.71/0.37	0.65/0.24	0.63/0.26	0.72/0.39	0.67/0.40	0.67/0.39	–	–	0.87/0.62	**0.88**/0.54	0.63/0.29
	Fusion-combo ⋆	**0.75**/0.48	0.69/0.31	0.64/0.27	**0.75**/0.46	0.72/0.41	**0.71**/0.39	–	–	**0.89/0.71**	0.86/0.53	0.62/0.34
Qwen-72B												
	PTrue-trained	0.75/0.44	0.56/0.11	0.62/0.32	0.68/0.37	0.63/0.28	0.70/0.35	–	0.68/0.32	0.63/0.28	0.57/0.21	0.70/0.40
	PTrue-Kbrain	0.76/0.44	0.64/0.17	**0.72/0.48**	0.71/0.29	0.59/0.26	**0.75/0.47**	–	0.60/0.14	0.64/0.30	**0.88**/0.57	0.51/0.03
Multi-sample	LexSim	0.75/**0.56**	0.63/0.23	0.66/0.38	0.64/0.28	0.57/0.08	0.66/0.39	–	0.71/0.38	0.66/0.28	0.72/0.39	0.67/**0.43**
	SelfCk-Ngram1-avg	0.79/0.54	0.59/0.19	0.67/0.42	0.64/0.21	0.54/0.12	0.59/0.28	–	0.72/0.38	0.71/0.37	0.70/0.53	0.64/0.41
	SelfCk-NLI-avg	0.53/0.12	0.54/0.12	0.55/0.07	0.56/0.16	0.60/0.20	0.61/0.34	–	0.49/-0.04	0.57/0.18	0.47/0.02	0.55/0.17
	SemanticEntropy	0.58/0.21	0.50/0.06	0.48/0.03	0.57/0.13	0.56/0.04	0.59/0.21	–	0.50/-0.11	0.52/-0.10	0.46/-0.07	0.48/-0.02
Post-gen.	MSP-mean	0.75/0.46	0.61/0.21	0.69/0.40	0.68/0.37	0.57/0.10	0.69/0.43	–	**0.73**/0.43	0.67/0.32	0.73/0.38	**0.71**/0.43
	MSP-min	0.73/0.41	0.60/0.19	0.68/0.38	0.68/0.37	0.57/0.10	0.68/0.41	–	0.67/0.35	0.66/0.30	0.72/0.37	0.66/0.30
Pre-gen.	P(IK)	0.77/0.47	0.71/0.39	0.66/0.33	0.67/0.28	0.70/0.35	0.72/0.44	–	0.64/0.43	0.83/0.57	0.65/0.24	0.58/0.15
	SAPLMA	0.74/0.33	0.64/0.18	0.69/0.32	0.71/0.26	0.65/0.26	0.58/0.22	–	0.44/-0.16	0.62/0.20	0.78/**0.59**	0.65/0.25
	SEPs-Ridge	0.61/0.27	0.65/0.28	0.49/0.06	0.59/0.20	0.61/0.24	0.61/0.36	–	0.58/0.07	0.63/0.24	0.62/0.12	0.59/0.03
	DCP-MLP ⋆	**0.81**/0.52	**0.72/0.40**	0.66/0.28	0.69/0.28	0.72/**0.41**	0.73/0.40	–	0.68/0.33	**0.86/0.66**	0.67/0.29	0.66/0.25
	DCP-LR ⋆	0.78/0.47	0.71/0.39	0.68/0.28	0.69/0.31	0.72/0.35	0.73/0.41	–	0.66/0.45	0.83/0.57	0.73/0.44	0.65/0.31
	ARD-MLP ⋆	0.73/0.37	0.67/0.25	0.71/0.43	**0.72**/0.36	0.71/0.39	0.65/0.32	–	0.60/0.05	0.83/0.59	0.79/0.40	0.63/0.28
	ARD-Ridge ⋆	0.75/0.36	0.64/0.20	0.69/0.38	0.71/**0.38**	0.67/0.29	0.66/0.35	–	0.70/0.36	0.77/0.48	0.76/0.37	0.55/0.13
	Fusion-combo ⋆	0.80/0.48	0.69/0.31	0.71/0.37	0.72/0.38	**0.73**/0.39	0.72/0.43	–	0.72/**0.52**	0.85/0.63	0.76/0.44	0.64/0.31

Cascade deployment. Tables 2 and 3 evaluate the same scores as routers from Qwen2.5-7B to stronger teachers on HotpotQA. DCP-MLP gives the strongest route AUROC in both settings (0.721), and with GPT-5.5 it also attains the best two-tier AUC, best improvement over random routing, largest Pareto-front coverage, and best end-to-end savings. With GPT-OSS-120B, HaMI slightly leads the raw AUC columns and ARD-MLP is the closest GNOSIS gate in two-tier AUC, but DCP-MLP remains within a narrow accuracy range while giving the best Pareto-front coverage and the largest time and token savings. These results show why deployment cost has to be measured end to end. Post-generation gates such as HaMI can be accurate routers, but they require the student answer before scoring; at the GPT-OSS-120B operating point, HaMI improves task AUC yet incurs negative generated-token savings. In contrast, GNOSIS scores are available before decoding, so escalated queries can bypass student generation. Figure 2 visualizes this distinction: GNOSIS gates occupy the favorable cost–accuracy region and remain token-efficient at teacher-level accuracy, whereas several sampling or post-generation gates lose their apparent advantage once decoding cost is included. Activation steering provides additional gains for some gates, but the primary system-level advantage comes from combining competitive routing with pre-generation availability.

4.3 Ablation and Mechanistic Analysis

We first isolate the contribution of Fusion's joint objective and then test whether DCP and ARD provide non-redundant evidence for that design.

Table 2: **HotpotQA cascade routing from Qwen2.5-7B to GPT-OSS-120B ($n = 500$).** Savings are measured relative to always using the teacher at a target accuracy of 45%. Higher is better for all metrics; formatting follows Table 1.

	Gate	Type	route AUROC ↑	AUC_2 ↑	vs-Rand ↑	Pareto-front% ↑	AUC_3 ↑	steer Δ ↑	save-time% ↑	save-token% ↑
Multi-sample	SelfCheck-NLI	unsup	0.570	40.24	+1.0	1.4	41.58	**+1.3**	-2.9	-31.6
	P(True)-trained	sup	0.597	40.02	+0.8	1.5	40.89	+0.9	+6.8	-21.7
	SemanticEntropy	unsup	0.498	39.53	+0.3	0.1	39.53	+0.0	-1.9	-30.6
	SelfCheck-Ngram	unsup	0.582	39.53	+0.3	0.2	40.19	+0.7	+0.1	-28.6
	LexSim	unsup	0.448	38.67	-0.5	1.2	39.49	+0.8	+1.1	-27.5
	P(True)-Kbrain	unsup	0.430	37.82	-1.4	0.1	38.69	+0.9	-0.4	-29.1
Post-gen.	HaMI	sup	0.679	**41.05**	**+1.8**	18.9	**41.70**	+0.6	+14.8	-14.0
	P(True)-greedy	unsup	0.574	40.25	+1.0	0.5	41.47	+1.2	+2.4	-26.2
	MSP	unsup	0.463	38.72	-0.5	0.1	39.63	+0.9	+3.9	-24.8
Pre-gen.	P(IK)	sup	0.617	40.81	+1.6	13.4	41.63	+0.8	+6.1	+3.7
	SEPs-LR	sup	0.550	39.89	+0.7	0.2	40.72	+0.8	+11.7	+7.5
	SEPs-Ridge	sup	0.558	39.86	+0.7	12.8	40.31	+0.5	+2.8	+1.7
	ARD-MLP ⋆	ours	0.670	41.00	**+1.8**	18.2	41.38	+0.4	+15.4	+10.3
	DCP-MLP ⋆	ours	**0.721**	40.97	**+1.8**	**19.3**	41.29	+0.3	**+20.3**	**+13.1**
	ARD-Ridge ⋆	ours	0.564	40.36	+1.2	4.4	41.21	+0.9	+16.8	+11.0
	Fusion-combo ⋆	ours	0.627	40.34	+1.1	7.5	40.96	+0.6	+15.9	+10.0

Table 3: **HotpotQA cascade routing from Qwen2.5-7B to GPT-5.5 (high reasoning, $n = 500$).** Savings are measured at a target accuracy of 61%; save-$% reports teacher API cost savings. Other conventions follow Table 2.

	Gate	Type	route AUROC ↑	AUC_2 ↑	vs-Rand ↑	Pareto-front% ↑	AUC_3 ↑	steer Δ ↑	save-token% ↑	save-$% ↑
Multi-sample	P(True)-trained	sup	0.597	49.59	+1.5	0.1	50.45	+0.9	-17.4	+13.0
	SelfCheck-NLI	unsup	0.570	49.54	+1.4	1.6	50.88	**+1.3**	-22.3	+7.1
	SelfCheck-Ngram	unsup	0.582	49.10	+1.0	0.1	49.77	+0.7	-21.7	+8.9
	SemanticEntropy	unsup	0.498	48.69	+0.6	0.1	48.69	+0.0	-23.5	+6.4
	LexSim	unsup	0.448	47.47	-0.6	0.1	48.29	+0.8	-24.5	+4.9
	P(True)-Kbrain	unsup	0.430	46.71	-1.4	0.1	47.58	+0.9	-22.4	+7.0
Post-gen.	HaMI	sup	0.679	50.84	+2.7	19.9	**51.49**	+0.6	-18.6	+15.0
	P(True)-greedy	unsup	0.574	49.39	+1.3	3.9	50.62	+1.2	-19.8	+10.8
	MSP	unsup	0.463	47.43	-0.7	0.1	48.34	+0.9	-22.3	+7.1
Pre-gen.	P(IK)	sup	0.617	50.00	+1.9	5.9	50.83	+0.8	+4.0	+9.6
	SEPs-LR	sup	0.550	48.56	+0.5	0.5	49.39	+0.8	+4.1	+9.6
	SEPs-Ridge	sup	0.558	48.52	+0.4	1.1	48.97	+0.5	+2.5	+5.7
	DCP-MLP ⋆	ours	**0.721**	**51.00**	**+2.9**	**46.0**	51.32	+0.3	**+7.0**	**+17.1**
	ARD-MLP ⋆	ours	0.670	50.69	+2.6	15.8	51.07	+0.4	+6.3	+14.9
	Fusion-combo ⋆	ours	0.627	49.81	+1.7	4.4	50.42	+0.6	+5.0	+13.8
	ARD-Ridge ⋆	ours	0.564	48.34	+0.2	0.1	49.19	+0.9	+2.4	+8.0

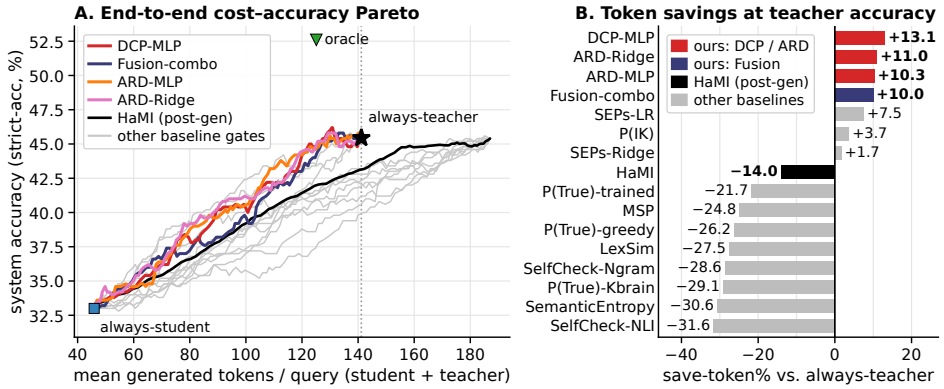
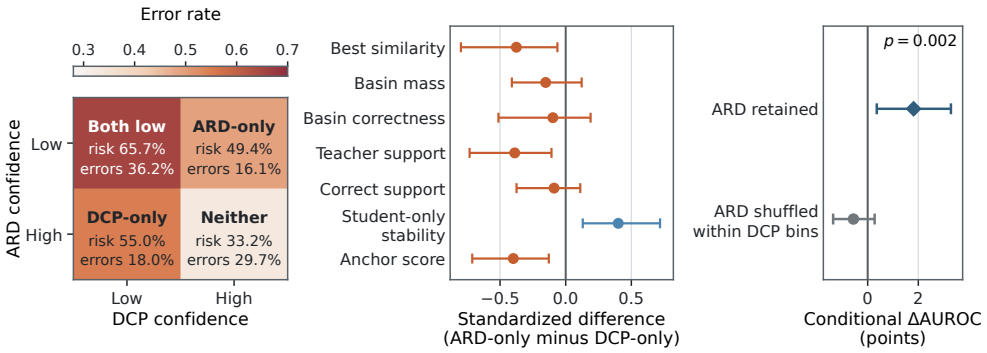


Figure 2: **HotpotQA-OOD 2-tier cascade** (Qwen2.5-7B → GPT-OSS-120B, n=500; complements Table 2). (A) System accuracy against end-to-end generated tokens per query; colored curves are our gates, gray curves the baseline gates, and black is HaMI. (B) Gates ranked by end-to-end save-token% at teacher-level accuracy (red / navy: ours; black: HaMI; gray: other baselines).

Fusion objective ablation. On Qwen-7B, the default $\lambda = 1$ reaches AUROC 0.694 and Lift@50 9.68, compared with $0.672/8.59$ for DCP-MLP and $0.678/8.17$ for late score averaging (Figure 5(a)). Neighboring settings $\lambda \in \{0.5, 2\}$ remain close, while the correctness-only control does not recover the AUROC. The gain thus favors joint representation learning over post-hoc score combination and is not tied to a narrow loss weight. Appendix C.1 reports the full objective sweep and the concurrent-request cascade results, including mean completion versus batch makespan.



(a) Complementary error sectors (b) Unique-error phenotype (c) ARD information beyond DCP

Figure 3: **DCP–ARD complementarity** on Qwen2.5-7B across 11 LongBench tasks. (a) Error concentration under matched 40% per-task rejection. (b) Task-standardized ARD-target differences between ARD-only and DCP-only errors, with 95% task-bootstrap intervals. (c) Conditional AUROC gain from adding ARD; gray denotes the within-DCP-decile permutation null.

DCP–ARD complementarity. At a matched 40% per-task rejection quota, DCP and ARD correlate moderately (task-normalized Spearman $\rho = 0.489$), but their errors differ (Figure 3(a)). The jointly low-confidence sector has the highest risk (65.7%), while ARD-only and DCP-only rejections capture 16.1% and 18.0% of all errors, so each readout covers failures that the other misses.

Panel (b) further distinguishes these errors: relative to DCP-only errors, ARD-only errors have lower best similarity, support mass, and anchor score, but higher student-only stability, with 95% task-bootstrap intervals excluding zero for all four contrasts. This pattern matches internally stable yet externally unsupported answers, but is an association rather than a causal account. In panel (c), adding ARD to a cross-fitted DCP stack yields $+1.81$ AUROC points (95% CI $[0.36, 3.29]$); shuffling ARD within DCP deciles removes the gain (one-sided $p = 0.002$). Together, the ablation and conditional test attribute Fusion's improvement to complementary consistency information rather than to score recalibration alone. Appendix C.2 examines the sensitivity of this target to the teacher source through cross-teacher agreement and a variance decomposition.

5 Conclusion

GNOSIS estimates answer reliability from the prompt-final hidden state before decoding. DCP predicts correctness, ARD learns teacher-anchored cross-model consistency that needs no teacher at test time, and Fusion trains both on a shared trunk. GNOSIS is competitive on LongBench and lowers end-to-end cascade cost, and its two readouts capture complementary failures. Pre-generation signals are weaker on GPT-OSS-20B, likely because its answers depend on a reasoning trace that the prompt-final state cannot anticipate (Appendix E). In future work, we plan to aggregate information across layers instead of selecting one layer per task, to reduce ARD's reliance on a stronger teacher, and to test these findings with more student models, datasets, and graded rather than thresholded quality labels. For reasoning models such as GPT-OSS-20B, combining the prompt state with a few decoded tokens may recover the missing signal.

AI use statement

We used generative AI tools to aid writing, related-work search, and research ideation and execution. We have reviewed all AI-assisted content and take responsibility for the final content of this work.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. AutoMix: Automatically mixing language models. *Advances in Neural Information Processing Systems*, 37: 131000–131034, 2024.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Xiaoxue Chen, Yu Zhao, Yucheng Liu, and Zhihang Fu. INSIDE: LLMs' internal states retain the power of hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP)*, 2024.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0.

Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. In *International Conference on Learning Representations*, volume 2025, pp. 77965–78000, 2025.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Daniela Gottesman and Mor Geva. Estimating knowledge in large language models without generating a single token. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3994–4019, 2024.

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 *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 16209–16226, 2025.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM transactions on information systems*, 43(2):1–55, 2025.

Mandar Joshi, Eunsol Choi, Daniel S 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, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5684–5696, 2020.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*, 2023.

Siyang Liu, Ke Liu, Yicheng Chen, Yang You, et al. LLM internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop*, 2024.

David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. *Advances in neural information processing systems*, 31, 2018.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pp. 7076–7087. PMLR, 2020.

Mengjia Niu, Hamed Haddadi, and Guansong Pang. Robust hallucination detection in llms via adaptive token selection. *Advances in Neural Information Processing Systems*, 38:126355–126377, 2026.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, Mohammed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *International Conference on Learning Representations*, volume 2025, pp. 66880–66913, 2025.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. The curious case of hallucinatory (un) answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3607–3625, 2023.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14379–14391, 2024.

Egor Trapeznikov and Xiaoqi Wang. UCCI: Calibrated uncertainty for cost-optimal LLM cascade routing. *arXiv preprint arXiv:2605.18796*, 2024.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. Investigating selective prediction approaches across several tasks in iid, ood, and adversarial settings. In *Findings of the association for computational linguistics: Acl 2022*, pp. 1995–2002, 2022.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 13:529–556, 2025.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1040–1051, 2021.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, volume 2024, pp. 23650–23678, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. In *International Conference on Learning Representations*, volume 2024, pp. 21691–21728, 2024.

Zhenliang Zhang, Xinyu Hu, Huixuan Zhang, Junzhe Zhang, and Xiaojun Wan. ICR probe: Tracking hidden state dynamics for reliable hallucination detection in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17986–18002, 2025.

A Definition and Design of the Teacher Anchor

Section 3.1 introduces the teacher-anchor target $\Phi_T(x_i)$ of Eq. (5) in compact form. This appendix gives the exact construction of its seven coordinates and the reasoning behind them. The target describes the cross-model consistency of the student's answer, that is, where the answer lies relative to the answers of a stronger model. A correctness label records only the outcome, whereas the anchor separates four kinds of evidence behind it: how closely the student's answer matches a teacher answer, how much teacher mass supports it, whether the supporting teacher answers are correct, and whether the student holds the answer stably without teacher support. Building these quantities requires teacher answers, gold answers, and student samples, but only for the regression targets. At test time ARD predicts the whole vector from the prompt-final hidden state.

Teacher basins. For each training question the teacher produces $K{=}4$ answers. The canonical form $c(\cdot)$ in Eq. (3) lowercases an answer, replaces non-alphanumeric characters with spaces, removes the function words *a, an, the, of, to, in, on, for, and, or, with, by*, and keeps the first twelve tokens. Answers with the same canonical form make up one basin, whose mass is its share of the teacher answers. The earliest decoded answer of a basin serves as its representative $r^T_{i,j}$. We group teacher answers canonically rather than with NLI-based clustering because the grouping is deterministic and adds no model whose errors would propagate into the teacher-side coordinates. Variants that canonical grouping keeps apart, such as an answer with and without a first name, are reconnected by the similarity-based support defined below.

Answer alignment. Let $\bar{a}$ denote an answer after lowercasing and replacing non-alphanumeric characters with spaces. Student and teacher answers are compared with the graded similarity

$$\mathrm{sim}(a,b) = \begin{cases} 1, & \bar{a} = \bar{b}, \\ 0.95, & \bar{a} \text{ is a substring of } \bar{b}, \text{ or } \bar{b} \text{ of } \bar{a}, \\ \mathrm{F1}(a,b), & \text{otherwise}, \end{cases} \tag{12}$$

where F1 is token-level F1. Containment scores slightly below an exact match, so an exact match is preferred when both occur. The threshold $\tau_{\text{align}} = 0.8$ of Eq. (4) accepts exact and containment matches as well as close token overlaps such as *John Kennedy* and *John F. Kennedy* ($\mathrm{F1} = 0.8$). It rejects answers that share only part of a name, such as *Michelle Obama* and *Barack Obama* ($\mathrm{F1} = 0.5$).

Coordinates. Let $\hat{a}^S_i$ be the student answer, and let $\kappa_{i,j} \in \{0,1\}$ indicate whether basin $B^T_{i,j}$ contains a teacher answer that is correct under the task criterion. The best-matched basin $j^\star_i$ is the one whose representative maximizes $\mathrm{sim}(\hat{a}^S_i, r^T_{i,j})$, with ties broken in favor of the heavier basin, and $m^{\max}_i$ and $c^{\max}_i$ are its mass and correctness. When the similarity is zero for every basin, both are set to zero. Table 4 lists all seven coordinates. The student-only stability uses ρ_i, the mass of the semantic cluster that contains $\hat{a}^S_i$ among the student's own sampled answers. The anchor score is

$$a_i = 0.5\, m^{\text{sup}}_i + 0.2\, s^{\max}_i + 0.2\, v_i - 0.2\, m^{S\setminus T}_i, \tag{13}$$

where v_i is a fixed linear score of normalized student-side statistics of $\hat{a}^S_i$. It increases with the mean token log-probability and with the size and mass of the answer's cluster, and it decreases with token entropy and answer length. The weights of Eq. (13) and of v_i are shared by all tasks, base models, and teachers.

Design rationale. The first six coordinates form three groups, and the anchor score summarizes them.

The first group, $s^{\max}$, $m^{\max}$, and m^{sup}, measures agreement with the teacher without using gold answers. The similarity $s^{\max}$ is graded, so it still separates a near miss from an unrelated answer when no basin passes τ_{align} and both support coordinates are zero. The two masses differ when canonical grouping splits one answer into several basins. For the student answer *John F. Kennedy* and the teacher answers *John F. Kennedy*, *John F. Kennedy*, *Kennedy*, and *Lyndon Johnson*, the best-matched basin gives $s^{\max} = 1$ and $m^{\max} = 0.5$, while the containment match with *Kennedy* raises the support to $m^{\text{sup}} = 0.75$. The basin mass thus describes the single closest teacher answer, and the support describes all teacher answers that agree with the student.

Table 4: **Coordinates of the teacher anchor** $\Phi_T(x_i)$, in the order of Eq. (5). Names follow Figure 3(b). Sums run over the teacher basins of question i, and $\mathbf{1}_{ij} = \mathbf{1}[\mathrm{sim}(\hat{a}_i^S, r_{i,j}^T) \geq \tau_{\mathrm{align}}]$. The column Gold marks the coordinates that use the gold answers of the training set.

	Name	Definition	Gold	Evidence
$s_i^{\max}$	best similarity	$\max_j \mathrm{sim}(\hat{a}_i^S, r_{i,j}^T)$	no	closeness to the nearest teacher answer
$m_i^{\max}$	basin mass	$m_{i,j_i^\star}^T$	no	teacher mass on that answer
$c_i^{\max}$	basin correctness	$\kappa_{i,j_i^\star}$	yes	whether that answer is correct
m_i^{sup}	teacher support	$\sum_j m_{i,j}^T \mathbf{1}_{ij}$	no	teacher mass that agrees with the student
m_i^{corr}	correct support	$\sum_j m_{i,j}^T \kappa_{i,j} \mathbf{1}_{ij}$	yes	agreeing teacher mass that is correct
$m_i^{S\setminus T}$	student-only stability	$\rho_i\,(1 - m_i^{\mathrm{sup}})$	no	student stability without teacher support
a_i	anchor score	Eq. (13)	no	label-free summary used for ranking

Agreement alone cannot separate a correct consensus from an error that the student and the teacher share. The second group, $c^{\max}$ and m^{corr}, uses the gold answers of the training set to mark which part of the agreeing teacher mass is correct. The difference $m^{\mathrm{sup}} - m^{\mathrm{corr}}$ is the teacher mass that agrees with the student on a wrong answer.

The third group, $m^{S\setminus T}$, covers a case in which self-consistency signals mislead. An answer that the student reproduces across samples has low semantic entropy and looks reliable to self-consistency checks, whether or not it is correct. The product $\rho_i(1-m_i^{\mathrm{sup}})$ is large only when this stability comes without teacher support, that is, when the answer is internally stable but externally unsupported. In Figure 3(b), errors rejected only by ARD have higher student-only stability than errors rejected only by DCP.

The anchor score combines the label-free coordinates into the scalar used for ranking. Teacher support carries the largest weight, similarity and the student's own confidence refine the order, and the last term lowers answers that the student holds without teacher support. We keep $c^{\max}$ and m^{corr} out of a_i so that the ranking coordinate summarizes agreement structure instead of restating the correctness label that DCP already fits. Gold answers therefore reach ARD only through the auxiliary coordinates.

Use in training and inference. ARD regresses all seven coordinates with the squared loss of Eq. (6). In ARD-MLP and in the anchor head of Fusion the outputs share hidden layers, so the other six coordinates act as auxiliary targets for the representation from which a_i is read; ARD-Ridge fits each coordinate independently. On LongBench the ARD score is the predicted a_i, as in Eq. (7). In the cascade experiments a logistic regression on the seven predicted coordinates gives the correctness score (Section 4). Teacher answers, gold answers, and student samples enter only through the training targets, so inference uses a single forward pass of the frozen student.

Scope. With four teacher answers per question the mass coordinates are coarse. They indicate whether the student's answer is the teacher's consensus, a minority variant, or absent from the teacher's answers, rather than estimating the teacher distribution precisely. The string-based alignment is designed for short answers, so on summarization tasks it reflects lexical agreement with the teacher. Appendix C.2 analyzes how the anchor depends on the choice of teacher.

B TASK-NATIVE UTILITY ON LONGBENCH

Table 1 scores the detectors with threshold-free metrics. Figure 4 examines the same rankings from the user's perspective: for each task, base model, and coverage $c \in \{25, 50, 75\}$, only the $c\%$ most confident predictions are answered, and we plot Lift@c, the resulting change in the task's official LongBench metric relative to answering every query. Each marker is one detector at one coverage. Within each task–coverage cell, the best and runner-up detectors are drawn in orange and labeled with a short method name; all other detectors are shown in light blue. Panel (d) counts these top-two placements over all cells with available results.

Fusion collects the most top-two placements (35), followed by DCP-LR (27) and DCP-MLP (26); the strongest baselines, LexSim and P(IK), reach 19 each. The advantage is not uniform: on several summarization and short-answer cells the lexical-similarity and MSP baselines still take a top-two place, so the margins in Table 1 do not translate into the same utility gain at every operating point.

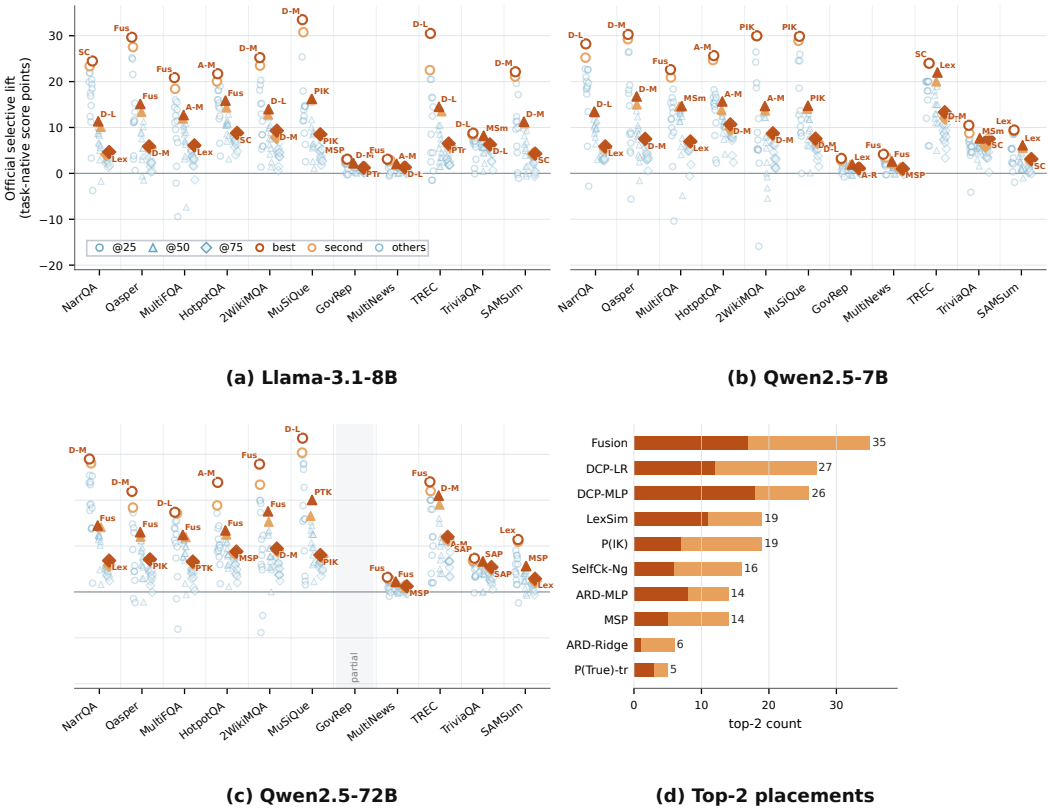


Figure 4: **Task-native LongBench utility across model scales.** For (a) Llama-3.1-8B, (b) Qwen2.5-7B, and (c) Qwen2.5-72B, markers report Lift@c for $c \in \{25, 50, 75\}$ (circles, triangles, and diamonds), the gain in each task's official score after retaining the top-c% most confident predictions. Orange marks the best and runner-up detector within each task, model, and coverage; light blue marks the remaining detectors. Short labels name the winners: D-M/D-L = DCP-MLP/DCP-LR, A-M/A-R = ARD-MLP/ARD-Ridge, Fus = Fusion, PIK = P(IK), PTr/PTK = P(True)-trained/P(True)-Kbrain, SAP = SAPLMA, Lex = LexSim, SC = SelfCheck (n-gram), MSP/MSm = MSP-mean/MSP-min. Panel (d) aggregates top-two placements over available cells. Panels (a)–(b) are shown in the upper row and (c)–(d) in the lower row.

C Additional Ablation and Mechanistic Analyses

C.1 Fusion objective and concurrent-request scaling

The Fusion objective is

$$\mathcal{L}_{\text{Fusion}} = \mathcal{L}_{\text{BCE}} + \lambda \left\| \widehat{\Phi}_T^{\text{fus}} - \Phi_T \right\|_2^2, \tag{14}$$

where the BCE coefficient is fixed to one and λ weights the auxiliary teacher-anchor regression term; λ=1 is the default. The objective ablation compares the shared dual-head probe with a correctness-only control and with post-hoc averaging of separately trained DCP and ARD scores. The relatively flat λ sweep indicates that the Fusion gain is not tied to a narrowly tuned loss trade-off (Figure 5(a)).

We also evaluate deployment under increasing concurrent load rather than cascade depth. W denotes simultaneous requests, and each route is measured at the escalation rate matching its all-strong

accuracy. Mean completion time and batch makespan answer different operational questions: a route can improve average latency without improving batch completion. At the matched operating point, the selected route saves 74.4% of strong-model tokens and 54.1% of parameter-weighted token compute; the observed speedup depends on both the weak–strong pairing and the workload (Figure 5(b)).

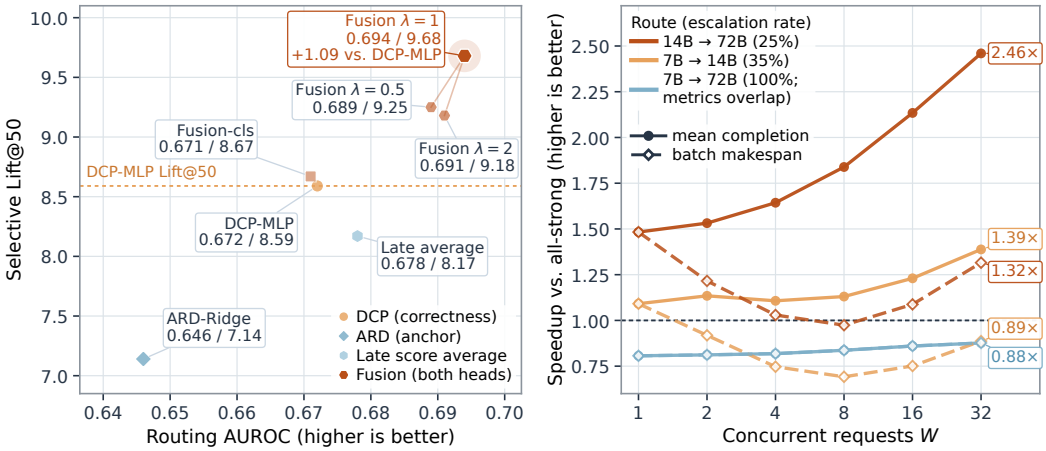


Figure 5: **Objective and systems ablations.** (a) Fusion objective ablation on Qwen-7B; labels give AUROC / Lift@50, and λ weights the auxiliary teacher-anchor regression term. (b) Concurrent-request scaling of cascade deployment; W is the number of simultaneous requests, not the number of cascade tiers. Solid and dashed curves denote mean completion and batch makespan speedup, respectively, and the values on the right are the speedups at $W{=}32$.

C.2 Sensitivity to the teacher source

The student, questions, canonical layer, and paired $K{=}4$ sampling protocol are fixed. Anchor-score rankings are more consistent among Qwen teachers than between Qwen and GPT-OSS. Question identity explains about 72% of variation, whereas the teacher main effect is about 4%; the remaining component includes question–teacher interaction and finite-K sampling noise. Downstream Lift@50 is not monotonic in teacher parameter count. ARD is therefore teacher-robust but not teacher-invariant, so the teacher remains part of the probe specification.

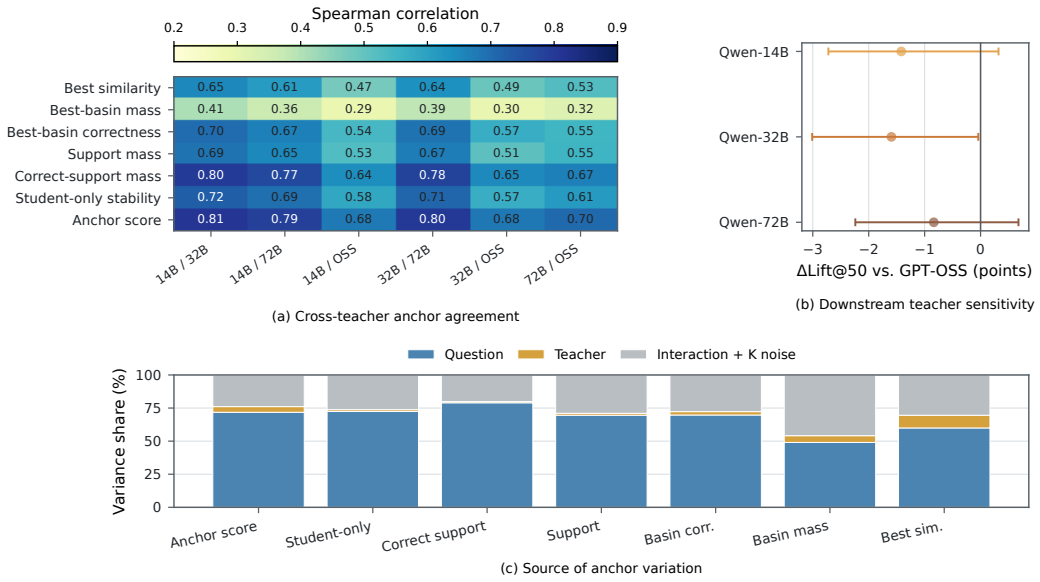


Figure 6: **Sensitivity of ARD to the teacher source.** The student, questions, canonical layer, and $K{=}4$ teacher sampling protocol are fixed. (a) Pairwise Spearman agreement of anchor coordinates across teachers. (b) Downstream Lift@50 difference relative to GPT-OSS-120B. (c) Variance decomposition of each anchor coordinate; the interaction term also contains finite-K sampling noise.

D Activation Steering in the Three-Tier Cascade

The three-tier cascade adds one option to the two-tier policy of Section 4. For an escalation fraction e, the lowest-confidence e fraction of queries is sent to the teacher as before. Among the remaining queries, those whose gate score lies in the middle tercile of that gate's scores on the evaluation set are answered by the student under activation steering, and all others by the unmodified student. The steered answer replaces the student answer, so this tier costs one student decode, the same as the student tier, and the cost axis of the cascade is unchanged. This accounting is favorable to post-generation gates: their score becomes available only after a first decode, so steering a query would require a second decode, which we do not charge.

Steering follows the mean-difference recipe of contrastive activation addition (Rimsky et al., 2024). From the prompt-final hidden states of Qwen2.5-7B at layer 20, we compute the difference between the mean state of correctly and incorrectly answered questions and normalize it to a unit vector $\mathbf{u}$. During generation, $\alpha\,\bar{r}\,\mathbf{u}$ is added to the residual stream at the output of decoder block 20, at the last position of every forward pass, which covers the final prompt token and each generated token; $\bar{r}$ is the mean norm of the layer-20 prompt-final states. Steered and unsteered answers are both decoded greedily, with at most 64 new tokens on HotpotQA.

The direction is estimated on the HotpotQA evaluation questions with five-fold cross-fitting: the vector applied to a question is computed from the other four folds, with the correctness of the student's greedy answers as labels, so no question's own label enters its steering vector. The gates are still trained only on TriviaQA, and the cross-fitted direction is the only component of the three-tier results that uses HotpotQA labels.

The layer and the strength were fixed before the cascade experiments. A pilot on the LongBench Qasper task compared layers 16, 20, and 24, after which layer 20 with $\alpha \in \{0.2, 0.4\}$ was applied to all eleven LongBench tasks without per-task tuning. With $\alpha{=}0.2$ the average gain in official score was positive and more uniform across tasks, whereas $\alpha{=}0.4$ amplified both the gains on tasks where steering helped and the losses on tasks where it hurt, with a slightly lower average. We therefore use $\alpha{=}0.2$ in all cascade experiments and do not tune it on HotpotQA. Tables 2 and 3 report the resulting AUC_3 and steerΔ; the gains are small, between 0.0 and 1.3 points, and steering is not part of the GNOSIS probes.

E PER-TASK LONGBENCH RESULTS FOR THE REMAINING BASE MODELS

Table 1 reports three of the eight base models, and Table 5 reports the other five under the same protocol. On Llama-3.2-3B, Qwen2.5-7B, Qwen2.5-14B, and Qwen2.5-32B, a GNOSIS variant attains the highest mean AUROC, and Fusion ranks first on Qwen2.5-7B and Qwen2.5-32B. GPT-OSS-20B is the exception. On this model, SelfCheckGPT with n-gram scoring leads on most tasks, and all pre-generation probes trail the best multi-sample and post-generation scores, although DCP-MLP remains the strongest pre-generation method (mean AUROC 0.67, against 0.62 for P(IK)). A plausible reason is that GPT-OSS-20B writes a reasoning trace before its final answer, so the prompt-final state anticipates less of the computation that determines correctness.

Table 5: **Selective prediction on LongBench for the remaining five base models.** Metrics, protocol, and marking follow Table 1. Llama-3.2-3B has no SAMSum cell because its labels are degenerate, and the two P(True) baselines are unavailable for GPT-OSS-20B.

		single-doc QA			**multi-doc QA**			**summary**		**classification**	**short-QA**	
	Method	NarrQA	Qasper	MultiFQA	HotpQA	2WikiM	MuSiQue	GovRep	MultiN	TREC	TrivQA	SAMSum
Llama-3.2-3B												
	PTrue-trained	0.65/0.29	0.59/0.15	0.62/0.28	0.58/0.21	0.58/0.23	**0.73/0.55**	0.60/0.18	0.50/0.01	0.38/-0.18	0.78/0.39	–
	PTrue-Kbrain	0.60/0.23	0.49/0.07	0.59/0.28	0.60/0.19	0.63/0.25	0.69/0.55	0.67/0.25	0.49/0.20	0.44/-0.40	0.78/0.37	–
Multi-sample	LexSim	0.73/0.32	0.73/0.43	0.65/0.35	0.68/0.37	0.60/0.30	0.66/0.48	**0.82/0.58**	0.70/0.48	0.59/0.16	0.87/0.68	–
	SelfCk-Ngram1-avg	0.80/0.39	**0.77**/0.38	0.68/0.37	0.67/0.26	0.53/0.16	0.63/0.36	0.74/0.48	0.71/0.44	0.73/0.64	0.91/0.78	–
	SelfCk-NLI-avg	0.63/0.28	0.61/0.21	0.57/0.25	0.63/0.27	0.57/0.16	0.60/0.40	0.55/-0.02	0.53/0.11	0.47/-0.12	0.88/0.73	–
	SemanticEntropy	0.59/0.34	0.50/-0.03	0.50/0.02	0.57/0.12	0.56/0.21	0.59/0.40	0.49/0.16	0.47/-0.15	0.49/-0.13	0.53/0.03	–
Post-gen.	MSP-mean	0.74/0.40	0.66/0.25	0.61/0.19	0.67/0.28	0.58/0.26	0.66/0.37	0.60/0.27	0.58/0.29	0.49/-0.18	0.74/0.28	–
	MSP-min	0.74/0.35	0.64/0.23	0.57/0.10	0.66/0.27	0.57/0.23	0.63/0.29	0.59/0.28	0.57/0.24	0.46/-0.23	0.74/0.28	–
Pre-gen.	P(IK)	0.76/0.36	0.74/0.35	0.69/0.38	0.71/0.37	0.57/0.18	0.65/0.40	0.71/0.40	0.70/0.40	0.90/0.88	0.93/0.80	–
	SAPLMA	0.47/-0.05	0.65/0.19	0.63/0.27	0.62/0.22	0.55/0.15	0.63/0.36	0.68/0.29	0.52/0.02	0.56/0.27	0.69/0.24	–
	SEPs-Ridge	0.64/0.31	0.51/0.08	0.57/0.06	0.63/0.11	0.64/0.26	0.62/0.31	0.53/0.09	0.56/-0.07	0.46/0.03	0.57/0.12	–
	DCP-MLP ⋆	0.78/0.40	0.75/0.35	**0.70/0.41**	0.72/0.42	0.64/**0.30**	0.68/0.53	0.73/0.41	**0.78/0.49**	0.89/0.93	0.93/**0.81**	–
	DCP-LR ⋆	0.80/0.39	0.76/0.38	0.69/0.38	0.71/0.37	0.64/0.29	0.67/0.51	0.73/0.42	0.75/0.43	**0.92/0.94**	0.93/0.80	–
	ARD-MLP ⋆	0.76/0.29	0.75/**0.46**	0.68/0.33	0.73/0.37	0.64/0.25	0.58/0.18	0.59/0.25	0.73/0.33	0.75/0.72	0.93/0.78	–
	ARD-Ridge ⋆	0.74/0.31	0.71/0.36	0.68/0.33	0.74/0.39	0.61/0.28	0.58/0.26	0.65/0.31	0.74/0.43	0.72/0.52	0.92/0.77	–
	Fusion-combo ⋆	**0.81/0.41**	0.77/0.42	0.69/0.32	**0.74/0.43**	**0.65**/0.24	0.62/0.32	0.73/0.39	0.77/0.48	0.84/0.84	**0.94**/0.79	–
Qwen2.5-7B												
	PTrue-trained	0.70/0.35	0.63/0.24	0.72/0.42	0.67/0.34	0.60/0.15	0.73/0.53	0.61/0.34	0.71/0.38	0.75/**0.55**	0.71/0.43	0.71/0.44
	PTrue-Kbrain	0.68/0.40	0.66/0.28	0.70/0.46	0.69/0.34	0.57/0.16	0.74/0.50	0.45/-0.02	0.48/0.18	0.68/0.33	0.75/0.42	0.39/0.03
Multi-sample	LexSim	0.74/**0.55**	0.61/0.24	0.73/0.45	0.66/0.32	0.54/0.14	0.68/0.51	0.71/0.42	0.78/0.38	0.75/0.44	0.75/0.51	0.74/0.48
	SelfCk-Ngram1-avg	0.74/0.46	0.51/0.02	0.69/0.37	0.71/0.36	0.48/0.08	0.67/0.50	0.69/0.39	0.77/0.23	0.73/0.43	**0.79/0.57**	0.73/**0.54**
	SelfCk-NLI-avg	0.52/0.28	0.60/0.27	0.61/0.13	0.55/0.26	0.48/0.00	0.63/0.44	0.56/0.05	0.41/0.02	0.68/0.41	0.44/0.09	0.67/0.38
	SemanticEntropy	0.59/0.33	0.55/0.09	0.53/0.09	0.58/0.17	0.49/0.09	0.57/0.14	0.53/0.06	0.54/-0.05	0.60/0.18	0.54/0.18	0.50/-0.11
Post-gen.	MSP-mean	0.73/0.36	0.61/0.27	**0.76/0.50**	0.66/0.33	0.54/0.11	0.70/0.51	0.65/0.38	0.75/0.41	**0.78**/0.55	0.76/0.49	**0.74**/0.47
	MSP-min	0.74/0.36	0.60/0.22	0.74/0.44	0.67/0.34	0.54/0.12	0.68/0.46	0.65/0.37	0.73/0.38	0.74/0.46	0.76/0.48	0.68/0.38
Pre-gen.	P(IK)	0.76/0.40	0.75/0.39	0.69/0.42	0.70/0.39	**0.70**/0.37	0.76/0.54	0.61/0.30	0.71/0.25	0.67/0.30	0.73/0.36	0.51/0.09
	SAPLMA	0.65/0.33	0.66/0.11	0.62/0.34	0.67/0.27	0.57/0.14	0.65/0.31	0.68/0.35	0.67/0.24	0.62/0.23	0.69/0.41	0.51/0.07
	SEPs-Ridge	0.64/0.34	0.61/0.17	0.58/0.10	0.65/0.22	0.58/0.21	0.62/0.28	0.53/-0.02	0.63/0.01	0.58/0.16	0.55/0.22	0.55/0.05
	DCP-MLP ⋆	0.78/0.36	**0.80/0.51**	0.71/0.46	0.72/0.41	0.70/**0.39**	0.75/**0.57**	0.67/0.36	0.74/0.39	0.72/0.40	0.70/0.34	0.57/0.17
	DCP-LR ⋆	0.79/0.40	0.79/0.46	0.71/0.45	0.70/0.39	0.70/0.37	**0.78**/0.57	0.70/0.39	0.71/0.25	0.67/0.35	0.73/0.36	0.56/0.24
	ARD-MLP ⋆	0.73/0.31	0.66/0.29	0.69/0.37	**0.76/0.42**	0.68/0.38	0.66/0.43	0.64/0.26	0.69/0.30	0.72/0.39	0.72/0.31	0.64/0.34
	ARD-Ridge ⋆	0.74/0.43	0.66/0.22	0.67/0.35	0.73/0.37	0.64/0.30	0.67/0.37	0.74/**0.51**	0.71/0.30	0.70/0.37	0.72/0.45	0.64/0.30
	Fusion-combo ⋆	**0.79**/0.44	0.79/0.46	0.73/0.42	0.75/0.41	0.69/0.35	0.75/0.53	**0.74**/0.45	**0.79/0.47**	0.73/0.43	0.73/0.45	0.63/0.32
Qwen2.5-14B												
	PTrue-trained	0.73/0.40	0.54/0.08	0.73/**0.45**	0.60/0.26	0.62/0.23	0.67/0.45	0.69/0.44	0.71/0.27	0.63/0.23	0.84/0.59	0.65/0.35
	PTrue-Kbrain	0.71/0.38	0.64/0.29	0.73/0.43	0.70/0.34	0.74/0.40	0.65/0.35	0.52/0.04	0.57/0.11	**0.70/0.37**	0.88/0.59	0.52/0.04
Multi-sample	LexSim	0.71/0.44	0.61/0.20	**0.74**/0.42	0.65/0.23	0.59/0.20	0.64/0.42	0.66/0.34	0.81/0.39	0.67/0.34	**0.89**/0.62	0.66/0.41
	SelfCk-Ngram1-avg	0.80/0.50	0.51/0.07	0.67/0.36	0.62/0.18	0.46/-0.01	0.56/0.25	0.66/0.31	0.75/0.33	0.67/0.30	0.86/**0.69**	0.68/**0.51**
	SelfCk-NLI-avg	0.39/-0.15	0.64/0.29	0.56/0.16	0.58/0.21	0.60/0.20	0.58/0.28	0.46/-0.11	0.47/0.02	0.65/0.28	0.51/0.17	0.58/0.20
	SemanticEntropy	0.51/0.14	0.54/0.08	0.52/-0.01	0.56/0.06	0.52/0.10	0.60/0.33	0.49/0.09	0.46/-0.03	0.52/-0.07	0.51/0.03	0.55/0.11
Post-gen.	MSP-mean	0.78/0.43	0.59/0.13	0.74/0.43	0.64/0.23	0.59/0.20	0.67/0.47	**0.72/0.48**	0.75/0.33	0.63/0.22	0.87/0.58	0.65/0.41
	MSP-min	0.78/0.40	0.59/0.14	0.74/0.44	0.64/0.24	0.60/0.20	0.67/0.46	0.69/0.40	0.69/0.23	0.64/0.25	0.88/0.60	0.62/0.37
Pre-gen.	P(IK)	0.78/0.46	0.75/0.52	0.66/0.26	0.74/0.44	0.74/0.47	0.80/0.58	0.61/0.33	0.74/0.42	0.62/0.14	0.78/0.49	0.60/0.16
	SAPLMA	0.72/0.39	0.69/0.34	0.67/0.32	0.69/0.33	0.59/0.08	0.64/0.36	0.61/0.25	0.67/0.39	0.53/0.03	0.77/0.47	0.54/0.10
	SEPs-Ridge	0.58/0.29	0.57/0.13	0.53/0.14	0.64/0.26	0.58/0.10	0.58/0.24	0.59/0.22	0.45/-0.03	0.59/0.10	0.55/0.05	0.58/0.23
	DCP-MLP ⋆	**0.86/0.57**	**0.76/0.54**	0.67/0.26	0.76/0.49	0.74/**0.48**	**0.81/0.61**	0.65/0.38	0.75/**0.45**	0.65/0.21	0.82/0.56	0.65/0.21
	DCP-LR ⋆	0.84/0.54	0.75/0.52	0.70/0.26	**0.78**/0.49	**0.75**/0.45	0.80/0.60	0.67/0.39	0.74/0.32	0.66/0.25	0.84/0.59	**0.73**/0.33
	ARD-MLP ⋆	0.76/0.32	0.57/0.11	0.69/0.34	0.73/0.42	0.69/0.39	0.61/0.28	0.63/0.12	0.74/0.26	0.64/0.26	0.81/0.47	0.60/0.23
	ARD-Ridge ⋆	0.74/0.30	0.56/0.08	0.66/0.30	0.71/0.39	0.67/0.33	0.61/0.31	0.68/0.31	0.75/0.27	0.64/0.22	0.78/0.46	0.60/0.21
	Fusion-combo ⋆	0.82/0.47	0.71/0.34	0.72/0.28	0.77/**0.49**	0.74/0.43	0.75/0.56	0.71/0.44	**0.82**/0.36	0.67/0.30	0.83/0.56	0.67/0.41

Table 5: **Selective prediction on LongBench for the remaining five base models (continued).**

		single-doc QA			multi-doc QA			summary		classification	short-QA	
	Method	NarrQA	Qasper	MultiFQA	HotpQA	2WikiM	MuSiQue	GovRep	MultiN	TREC	TrivQA	SAMSum
Qwen2.5-32B												
	PTrue-trained	0.70/0.43	0.63/0.26	0.69/0.41	0.57/0.19	0.74/0.37	0.63/0.34	0.62/0.22	0.77/0.49	0.56/0.17	0.80/0.59	0.61/0.37
	PTrue-Kbrain	0.77/0.54	0.70/0.38	**0.81/0.52**	0.69/0.30	0.75/0.46	0.69/0.37	0.51/-0.02	0.59/0.22	0.75/0.45	0.87/0.67	0.52/0.19
Multi-sample	LexSim	0.71/0.49	0.56/0.09	0.73/0.42	0.61/0.20	0.61/0.31	0.59/0.19	0.70/**0.48**	0.80/0.46	0.60/0.18	0.85/0.59	**0.70/0.50**
	SelfCk-Ngram1-avg	0.80/**0.61**	0.46/-0.03	0.70/0.45	0.56/0.06	0.51/0.06	0.55/0.23	0.60/0.24	0.71/0.32	0.66/0.25	0.88/**0.74**	0.63/0.44
	SelfCk-NLI-avg	0.42/-0.04	0.55/0.12	0.57/0.09	0.61/0.19	0.58/0.19	0.50/0.13	0.50/-0.00	0.46/-0.12	0.71/0.39	0.49/0.04	0.61/0.32
	SemanticEntropy	0.53/0.11	0.49/0.01	0.51/-0.01	0.53/0.08	0.55/0.22	0.56/0.07	0.52/0.09	0.52/-0.08	0.50/0.02	0.56/0.10	0.52/0.04
Post-gen.	MSP-mean	0.73/0.40	0.54/0.05	0.71/0.43	0.61/0.21	0.62/0.31	0.64/0.32	0.66/0.25	**0.82**/0.50	0.58/0.13	0.86/0.63	0.67/0.49
	MSP-min	0.72/0.39	0.53/0.06	0.71/0.40	0.61/0.21	0.62/0.31	0.63/0.31	0.65/0.25	0.78/0.51	0.57/0.09	0.85/0.62	0.66/0.43
Pre-gen.	P(IK)	0.84/0.52	0.80/**0.54**	0.63/0.21	0.72/0.42	**0.78**/0.44	0.66/0.35	0.68/0.38	0.74/0.41	0.72/0.42	0.86/0.64	0.58/0.13
	SAPLMA	0.75/0.56	0.65/0.35	0.63/0.18	0.66/0.29	0.64/0.23	0.60/0.20	0.63/0.27	0.69/0.27	0.53/0.05	0.75/0.48	0.57/0.18
	SEPs-Ridge	0.62/0.23	0.59/0.16	0.55/0.21	0.56/0.02	0.63/0.27	0.65/0.34	0.56/0.07	0.65/0.18	0.54/0.10	0.63/0.22	0.55/0.13
	DCP-MLP ⋆	0.85/0.53	**0.81**/0.50	0.68/0.31	0.74/**0.43**	0.78/**0.47**	0.69/**0.45**	**0.74**/0.43	0.74/0.49	0.74/0.43	0.86/0.59	0.62/0.19
	DCP-LR ⋆	**0.85**/0.53	0.80/0.51	0.68/0.31	0.73/0.43	**0.78**/0.44	0.66/0.35	0.73/0.41	0.74/0.41	0.74/0.42	0.86/0.64	0.64/0.20
	ARD-MLP ⋆	0.77/0.41	0.62/0.11	0.70/0.33	0.72/0.39	0.73/0.37	0.61/0.33	0.61/0.15	0.76/0.38	0.78/**0.52**	0.88/0.64	0.62/0.36
	ARD-Ridge ⋆	0.76/0.48	0.60/0.17	0.68/0.31	0.71/0.31	0.71/0.35	0.62/0.29	0.65/0.26	0.78/0.49	0.75/0.46	0.89/0.65	0.59/0.22
	Fusion-combo ⋆	0.84/0.48	0.73/0.31	0.69/0.30	**0.76**/0.38	0.77/0.44	**0.70**/0.43	0.72/0.45	0.76/**0.60**	**0.79**/0.52	**0.92**/0.73	0.62/0.28
GPT-OSS-20B												
	PTrue-trained	–	–	–	–	–	–	–	–	–	–	–
	PTrue-Kbrain	–	–	–	–	–	–	–	–	–	–	–
Multi-sample	LexSim	0.79/0.41	0.76/0.45	0.77/0.51	0.86/0.69	0.77/0.51	0.86/0.67	0.63/0.24	0.72/0.36	0.73/0.35	0.80/0.59	0.65/0.31
	SelfCk-Ngram1-avg	**0.88/0.82**	**0.79/0.51**	**0.78/0.56**	**0.88/0.75**	0.79/**0.60**	0.88/0.78	**0.76/0.42**	**0.80/0.41**	**0.81/0.56**	**0.83/0.66**	**0.68**/0.34
	SelfCk-NLI-avg	0.75/0.51	0.69/0.36	0.63/0.27	0.67/0.36	0.67/0.30	0.70/0.37	0.42/0.02	0.59/0.03	0.71/0.40	0.71/0.37	0.60/0.23
	SemanticEntropy	0.52/0.08	0.53/0.07	0.48/-0.09	0.53/0.04	0.50/0.07	0.44/-0.30	0.50/0.02	0.61/0.06	0.54/0.11	0.51/0.05	0.48/-0.17
Post-gen.	MSP-mean	0.81/0.62	0.66/0.32	0.74/0.47	0.78/0.59	**0.80**/0.51	**0.89/0.84**	0.68/0.23	0.65/0.41	0.65/0.25	0.72/0.40	0.56/0.31
	MSP-min	0.82/0.59	0.63/0.24	0.71/0.41	0.75/0.55	0.80/0.49	0.87/0.82	0.66/0.20	0.63/0.32	0.61/0.20	0.69/0.32	0.56/0.27
Pre-gen.	P(IK)	0.71/0.36	0.66/0.21	0.62/0.20	0.57/0.13	0.63/0.25	0.75/0.61	0.48/0.05	0.75/0.27	0.62/0.18	0.46/-0.03	0.53/0.17
	SAPLMA	0.58/0.10	0.52/0.04	0.65/0.38	0.69/0.34	0.59/0.10	0.72/0.59	0.50/0.03	0.67/0.21	0.54/0.04	0.62/0.26	0.47/-0.02
	SEPs-Ridge	0.57/0.06	0.58/0.11	0.57/0.13	0.52/0.08	0.57/0.07	0.47/-0.07	0.49/-0.13	0.49/-0.05	0.54/0.10	0.57/0.08	0.59/0.10
	DCP-MLP ⋆	0.68/0.19	0.71/0.31	0.70/0.32	0.61/0.18	0.72/0.39	0.75/0.65	0.56/0.30	0.77/0.27	0.66/0.23	0.61/0.06	0.63/0.37
	DCP-LR ⋆	0.71/0.36	0.75/0.35	0.67/0.28	0.63/0.21	0.68/0.31	0.79/0.80	0.55/0.19	0.78/0.30	0.64/0.20	0.54/0.01	0.61/0.35
	ARD-MLP ⋆	0.68/0.43	0.68/0.21	0.70/0.32	0.61/0.16	0.69/0.32	0.75/0.61	0.62/0.14	0.62/0.15	0.55/0.08	0.54/0.10	0.63/0.25
	ARD-Ridge ⋆	0.69/0.41	0.67/0.17	0.70/0.34	0.58/0.13	0.64/0.19	0.73/0.69	0.65/0.23	0.61/0.18	0.54/0.01	0.60/0.12	0.58/0.22
	Fusion-combo ⋆	0.68/0.43	0.70/0.24	0.71/0.33	0.63/0.26	0.71/0.39	0.77/0.66	0.63/0.19	0.61/0.30	0.63/0.13	0.58/0.16	0.64/**0.39**