TEST-TIME ADAPTATION UNDER RESOURCE-CONSTRAINED DEPLOYMENT VIA ATTENTION-HEAD ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Distribution shifts can degrade the response quality of large language models (LLMs), while resource-constrained deployments may lack relevant corpora or the memory needed to host additional pretrained evaluators. Many existing adaptation methods also assume access to multiple test samples from a coherent target distribution to stabilize updates, an assumption that may not hold for queries from different users or domains. We propose per-sample test-time adaptation through attention-head alignment. Brief prefix exploration exposes higher-quality candidates, while prompt–candidate attention-head activation differences provide transferable signals for selecting them. A lightweight scorer calibrated offline on a small set of generic prompt–reference pairs uses these signals to select a response for each query. Both the LLM and scorer remain fixed at deployment, enabling forward-only adaptation without retrieval, additional pretrained evaluators, or information retained from previous queries. Across three LLMs and six benchmarks, our method achieves higher mean ROUGE-LSUM than greedy decoding and the evaluated episodic update baselines for the evaluated backbones.

1 INTRODUCTION

Large language models (LLMs) have emerged as a general-purpose interface for language understanding and generation, demonstrating strong few-shot generalization, instruction-following ability, and reasoning capabilities across a wide range of tasks (Brown et al., 2020; Achiam et al., 2023; Chung et al., 2024; Zhong et al., 2024). Although cloud-based LLMs deliver strong performance, extending their capabilities to edge devices remains challenging under limited memory and computation (Li et al., 2025a; Ruan et al., 2024; Alizadeh et al., 2024; Laskaridis et al., 2024). Chain-of-thought prompting further improves multi-step reasoning (Wei et al., 2022), but these capabilities can deteriorate when deployment inputs differ from pretraining or instruction-tuning data (Akyürek et al., 2025; Hu et al., 2025). Such *domain-specific* shifts arise from specialized terminology (e.g., legal clauses (Ariai et al., 2025) or medical abbreviations (Gu et al., 2021)), unfamiliar input or output formats, localized conventions (Bella et al., 2024), and evolving knowledge (Li et al., 2025b), potentially degrading reliability, factuality, and task performance. A straightforward response is domain-specific prompting, which guides generation toward a target area (Schulhoff et al., 2024). However, selecting an appropriate pre-designed prompt remains difficult when independent queries come from unknown or unfamiliar domains. As local LLM deployment expands, addressing these unpredictable shifts within the available resources becomes increasingly important.

Several inference-time strategies address this test-time challenge, with distinct resource requirements. Reranking methods use a separate LLM judge to evaluate and select among candidate outputs (Zheng et al., 2023; Jiang et al., 2023; Kim et al., 2023). Retrieval-augmented generation (RAG) improves factual grounding by conditioning generation on externally retrieved documents (Lewis et al., 2020; Borgeaud et al., 2022; Asai et al., 2023). Test-time learning (TTL) instead adapts model parameters during inference. Supervised TTL typically retrieves task-relevant data for gradient-based updates (Hardt & Sun, 2024), while unsupervised TTL optimizes objectives such as entropy or perplexity on unlabeled test inputs (Wang et al., 2021; Hu et al., 2025).

However, under resource-constrained deployment, the data requirements and adaptation assumptions of existing methods can be difficult to satisfy. First, **external corpora and evaluators may be**

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.

Table 1: Resource requirements and native single-query adaptation capabilities of representative methods under resource-constrained LLM deployment. Native support refers to independent-query operation in the published LLM protocol; TENT is characterized by its LLM instantiation in Hu et al. (2025).

Family	Method	No corpus/ retrieval	No auxiliary pretrained evaluator	Frozen LLM	Native single-query adaptation
Offline	Fine-tuning (Li et al., 2025c)	✗	✓	✗	✗
Retrieval	RAG (Lewis et al., 2020)	✗	✓	✓	✓
Parameter updates	TENT (Wang et al., 2021)	✓	✓	✗	✗
	TTT-NN (Hardt & Sun, 2024)	✗	✓	✗	✓
	TLM (Hu et al., 2025)	✓	✓	✗	✗
Candidate selection	Aux. evaluator (Jiang et al., 2023)	✓	✗	✓	✓
	Ours	✓	✓	✓	✓

unavailable. RAG and retrieval-based TTL require a document or example store (Lewis et al., 2020; Hardt & Sun, 2024), while separately pretrained judges or reward models require an additional model checkpoint (Jiang et al., 2023; Kim et al., 2023). Hosting these components alongside the generator adds to local storage and memory requirements, and a relevant corpus may be unavailable for an unfamiliar query. Offline domain-specific fine-tuning likewise assumes access to target-domain training data before deployment. Second, **access to a coherent target distribution cannot be assumed**. Unsupervised TTL methods can use batch-level statistics or shared updates across multiple test samples to stabilize optimization (Wang et al., 2021). However, local requests may come from different users, documents, or tasks, and request isolation can preclude retaining their data or adaptation state (Zhang & Couque, 2024). Adaptation must then rely on the current query rather than accumulate evidence from a coherent target stream. Third, **parameter updates can destabilize pretrained behavior**. Self-supervised updates on limited data need not improve response quality (Hu et al., 2025), and persistent adaptation can lead to catastrophic forgetting of previously acquired capabilities, even with parameter-efficient tuning (Li et al., 2025c; Hu et al., 2025). These concerns are particularly relevant in the *single-sample* regime, where each query provides limited evidence for adaptation. Together, these limitations, summarized in Table 1, motivate resource-efficient *test-time adaptation* (TTA) that improves responses to independent queries without requiring a target-domain corpus, additional samples from a coherent test distribution, or an auxiliary pretrained evaluator.

To address these limitations, we propose a *per-sample test-time adaptation* method for LLMs that operates on a *single* test instance, requires no external retrieval corpus, no access to additional samples from a coherent target distribution, and no updates to model parameters. Our approach is grounded in three empirical findings. First, **LLMs possess an intrinsic capacity to uncover higher-quality generations through limited exploration** (see **Observation 1**). In particular, lightweight exploration over top-n next-token predictions can substantially improve generation quality. We find that allowing the model to briefly branch—by selecting among the top-n predicted tokens for only a few initial decoding steps—often reveals candidate generations with higher similarity to the target output than those produced by greedy decoding. Even short explorations (e.g., depth 3–4 tokens with $n = 3$) can expose higher-quality alternatives already available to the frozen model. Second, **candidate quality correlates with activation differences in specific attention heads** (see **Observation 2**). Building on prior work demonstrating functional specialization of attention heads (Ma et al., 2025; Zhou et al., 2025), we find that prompt–candidate attention-head activation differences carry information about candidate quality, with correlations of different signs across heads. This alignment signal supports candidate-quality prediction where commonly used measures such as perplexity are less informative in our diagnostics. Third, **attention-head alignment patterns transfer across the evaluated domains** (see **Observation 3**). This suggests that a subset of attention heads carries transferable information about prompt–generation compatibility. The calibration-corpus evaluation further shows that useful aggregation weights can be learned from several generic instruction sources without target-domain calibration.

Building on these observations, **we propose a lightweight framework for per-sample test-time adaptation through attention-head alignment**. We first calibrate a candidate-quality scorer offline on a small reference-labeled set of generic instructions. At inference, the framework performs limited structured exploration to generate candidate responses and selects the highest-scoring candidate using prompt–candidate attention-head activation differences. Both the LLM and scorer remain fixed, enabling per-query adaptation through forward-only candidate selection without retrieval or

an auxiliary pretrained evaluator. We evaluate the complete procedure against greedy decoding and episodic parameter adaptation, and use controlled candidate pools to isolate the contribution of the attention-head signal. Our runtime analysis further examines the cost of exploration: although total inference cost depends on the candidate budget, the reduced-budget configuration achieves lower overhead than TENT in the evaluated GPU setting (Appendix E).

2 Related Work and Background

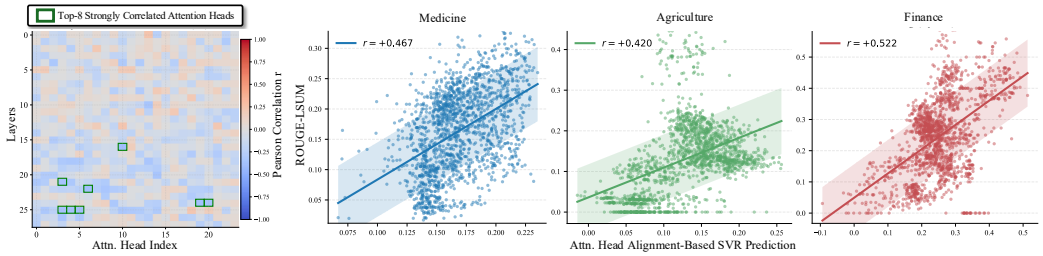
Test-Time Adaptation Methods for Resource-Constrained LLMs. LLM deployment on resource-constrained personal edge devices is limited by the memory required for inference, with prior work studying models that exceed available DRAM (Alizadeh et al., 2024) and reporting out-of-memory failures even with quantized models (Laskaridis et al., 2024). These constraints motivate addressing distribution shifts using the deployed LLM and the information available within each request. Successive queries may come from different users, tasks, or domains, so a coherent stream of target-domain samples cannot be assumed. We therefore focus on *per-query test-time adaptation*, which seeks to improve the response to an individual unlabeled query using the model's own signals. This formulation accommodates both parameter updates and adaptation of the response decision. Unsupervised TTL offers a promising approach in this setting by updating model parameters from unlabeled test inputs. Representative methods include TENT (Wang et al., 2021), EATA (Niu et al., 2022), and COME (Zhang et al., 2025), which use entropy-based adaptation objectives, and TLM (Hu et al., 2025), which minimizes input perplexity. These methods derive their learning signals from the deployed model itself, requiring neither retrieved training examples nor an auxiliary pretrained evaluator at inference. This makes them natural baselines for deployments with limited capacity to maintain additional data or serve additional models. To match independent-query processing, we consider their episodic LLM instantiations: each method adapts using only the current query, and adaptation state is reset between requests. This evaluates their effectiveness with the evidence available from a single query, without assuming access to additional samples from a shared target distribution.

Test-Time Adaptation in Vision. In vision, substantial progress has been made toward TTA with single-sample inputs and changing test distributions (Niu et al., 2023; Farina et al., 2024; Zanella et al., 2025). Many approaches exploit normalization statistics, predictions over a fixed set of classes, or image augmentations to obtain adaptation signals. Adapting these mechanisms to open-ended autoregressive generation remains challenging. In this setting, increased token-level confidence need not imply improved response quality, and entropy-based updates can degrade performance under distribution shift (Hu et al., 2025). This gap motivates practical TTA methods specifically designed for language generation that derive useful adaptation signals from individual queries while operating within limited deployment resources.

Multi-Head Attention and Functionality. Transformer multi-head attention represents token interactions through multiple parallel heads (Vaswani et al., 2017). Prior analyses show that heads are not interchangeable: different heads specialize in behaviors such as positional patterns, syntactic dependencies, coreference, delimiter tracking, and long-range routing, with such roles often stable across runs or checkpoints (Clark et al., 2019; Voita et al., 2019; Kovaleva et al., 2019). Pruning and ablation studies further show that many heads are redundant, while a small subset of critical or outlier heads can be disproportionately important, suggesting that head-level signals provide compact information about model computation (Michel et al., 2019; Voita et al., 2019). Recent mechanistic studies likewise indicate that head activations correlate with latent computational roles and can probe context or reasoning state (Ma et al., 2025; Zhou et al., 2025). These findings motivate our *head-activation alignment* as a lightweight internal signal to discriminate candidates under distribution shift.

3 Problem Formulation

Let f_Θ denote an LLM with fixed parameters Θ, trained on data drawn from a train distribution $\mathcal{P}_{\text{train}}$. After deployment, the model is queried with test-time inputs that may not follow a single coherent distribution. Instead, we consider a realistic setting in which each test instance x_i is drawn from its own (unknown) distribution $\mathcal{Q}_i$, and different test inputs x_i, x_j may satisfy $\mathcal{Q}_i \neq \mathcal{Q}_j$. This captures common deployment scenarios such as per-user queries, document-specific tasks, or rapidly shifting domains, where no shared test-time statistics can be assumed. Given a single test prompt $x \sim \mathcal{Q}_x$, the model induces a conditional distribution over outputs $p_\Theta(y \mid x)$. The objective is to produce an output y that is well aligned with $\mathcal{Q}_x$, despite the mismatch between $\mathcal{Q}_x$ and the training distribution $\mathcal{P}_{\text{train}}$. Importantly, we consider a resource-constrained setting with (i) no access to



(a) Per-attention-head correlation with task performance (b) Correlation between predictions from attention-head activation patterns and candidate similarity scores

Figure 2: Attention-head activation alignment between prompts and candidates in Llama-3.2-3B-Instruct. (a) Per-head Pearson correlation heatmap illustrating the relationship between candidate–target similarity and prompt–candidate attention-head alignment, measured as the L2 distance in attention-head activation space for each head, evaluated on 60 Medicine-domain prompts. (b) SVR regression performance for predicting candidate–target similarity from attention-head alignment features, with models trained and evaluated separately on each domain dataset (Medicine, Agriculture, and Finance), using 40 training prompts and 20 held-out test prompts per dataset.

additional samples from $\mathcal{Q}_x$ or an external retrieval corpus, (ii) no assumption of coherence across test-time distributions, and (iii) no auxiliary pretrained evaluator. Our method assumes access to attention-head outputs during forward passes and uses these signals to score candidate responses. The scorer is calibrated offline on a small reference-labeled set, after which both the scorer and model parameters Θ remain fixed throughout deployment.

We formulate per-sample test-time adaptation as a *candidate selection* problem. Given a test-time input $x \sim \mathcal{Q}_x$, the fixed model f_Θ induces a conditional distribution $p_\Theta(y \mid x)$ and produces a finite set of candidate outputs

$$\mathcal{C}(x) = \{y^{(1)}, \ldots, y^{(B)}\}, \quad y^{(b)} \in \mathrm{PrefixSearch}_{n,k}(f_\Theta, x). \tag{1}$$

Each candidate $y^{(b)}$ can be associated with an implicit, candidate-dependent output distribution $\mathcal{Q}_x^{(b)}$, reflecting the model's generative behavior along the corresponding decoding trajectory. Because the ground-truth target $y^\star \sim \mathcal{Q}_x$ is unobserved at test time, direct evaluation of candidate quality is infeasible. We therefore cast adaptation as the problem of selecting the candidate whose induced distribution $\mathcal{Q}_x^{(y)}$ is closest to the unknown test-time distribution $\mathcal{Q}_x$. Given a distributional distance or divergence D, we express this selection goal as

$$\hat{y} \;=\; \arg\min_{y \in \mathcal{C}(x)} D\big(\mathcal{Q}_x, \mathcal{Q}_x^{(y)}\big). \tag{2}$$

Since neither $\mathcal{Q}_x$ nor $\mathcal{Q}_x^{(y)}$ is directly observable, the selection criterion above cannot be computed directly. In the following section, we develop a practical proxy for alignment based on the relationship between attention-head activations and candidate quality (Observation 2) and its transfer across domains (Observation 3). Specifically, we calibrate a lightweight scorer offline to predict reference-based candidate quality from internal signals of the fixed model f_Θ. At test time, this scorer guides candidate selection without reference targets, additional test samples, or persistent adaptation state. Both the scorer and the LLM remain fixed, so per-sample adaptation operates through the response selected for the current query.

4 Methodology

Our objective is to select, for each test-time prompt x, a candidate generation whose behavior is *distributionally aligned* with the unknown test-time distribution $\mathcal{Q}_x$. Because neither $\mathcal{Q}_x$ nor candidate-induced distributions $\mathcal{Q}_x^{(y)}$ are observable, we seek a proxy signal derived from the internal computation of a fixed pretrained model f_Θ. We operationalize this goal in three stages: (i) constructing a candidate set that exposes potentially well-aligned generations, (ii) estimating candidate–prompt alignment using attention-head activation dynamics, and (iii) learning a lightweight

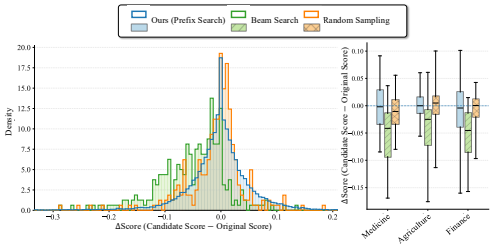


Figure 1: Comparison of candidate-generation strategies—prefix search, beam search, and random sampling—on three domain datasets using Llama-3.2-3B-Instruct, with 81 candidates per prompt. The horizontal score is candidate ROUGE-LSUM minus the greedy score: zero indicates parity and positive values indicate improvement. The histogram aggregates all datasets (left), and the box plot reports per-dataset distributions (right).

aggregation of these signals that generalizes across domains. We investigate whether these signals provide useful selection information beyond likelihood or perplexity under distribution shift.

4.1 Candidate Set Construction

The idealized objective in Section 3 motivates selecting

$$\hat{y} = \arg\min_{y \in \mathcal{C}(x)} D\Big(\mathcal{Q}_x, \mathcal{Q}_x^{(y)}\Big), \tag{3}$$

where $\mathcal{C}(x)$ is a finite set constructed through prefix exploration with the fixed model. A central challenge is that likelihood under p_Θ may be poorly aligned with task performance when $\mathcal{Q}_x \neq \mathcal{P}_{\text{train}}$.

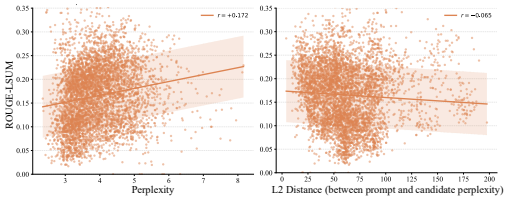


Figure 3: Perplexity-based baselines under distribution shift for Llama-3.2-3B-Instruct on 60 Medicine-domain prompts with measured Pearson coefficient. Left: candidate–target similarity versus candidate perplexity, showing a weak and unstable correlation under test-time distribution shift. Right: similarity versus perplexity alignment, measured as the L2 distance between prompt and candidate perplexities, exhibiting near-zero correlation. These results motivate alternative candidate-selection signals in this evaluated setting.

Assumption 1 (Likelihood–alignment mismatch under distribution shift). For a non-negligible set of prompts $x \sim \mathcal{Q}_x$, there exist candidates y such that

$$D\Big(\mathcal{Q}_x, \mathcal{Q}_x^{(y)}\Big) < D\Big(\mathcal{Q}_x, \mathcal{Q}_x^{(y^{(0)})}\Big), \tag{4}$$

$$p_\Theta(y \mid x) < p_\Theta(y^{(0)} \mid x), \tag{5}$$

where $y^{(0)}$ denotes the greedy decoding output.

Our intuition in Assumption 1 motivates candidate construction strategies that relax strict likelihood ordering while remaining within the model's support.

Prefix search. We adopt a prefix-based search strategy that introduces controlled branching early in decoding while preserving coherent continuation. Given prefix depth k (e.g., $k = 4$) and branching factor n (e.g., $n = 3$), we explore prefixes

$$\pi = (y_1, \ldots, y_k), \qquad y_t \in \text{Top-}n\ p_\Theta(\cdot \mid x, y_{<t}), \tag{6}$$

where Top-n denotes the set of the n highest-probability tokens under the model's conditional distribution $p_\Theta(\cdot \mid x, y_{<t})$, and the full prefix tree has n^k leaves. Each branch is explored independently of its exact likelihood value. After constructing the prefix, the remainder of the sequence is generated deterministically via greedy decoding:

$$y_t = \arg\max_v p_\Theta(v \mid x, y_{<t}), \qquad t > k. \tag{7}$$

This yields candidates that deviate from greedy decoding only through early prefix choices.

Observation 1 (Stable exposure of high-quality candidates). As shown in Figure 1, prefix search exposes candidates that improve over greedy decoding across the three evaluated domains, with candidate-quality distributions that differ from beam search and random sampling. These results support the use of modest prefix exploration to surface higher-quality candidates already available to the model, consistent with the intuition underlying Assumption 1.

4.2 Attention-Head Alignment as a Proxy for Distributional Alignment

Having constructed a candidate set that plausibly contains well-aligned outputs, we now seek a practical selection signal motivated by $D(\mathcal{Q}_x, \mathcal{Q}_x^{(y)})$. Since the distributions themselves are unobservable, we seek an internal signal whose variation reflects how a candidate generation interacts with the prompt-conditioned computation of the model.

Prior empirical work (Ma et al., 2025; Zhou et al., 2025) has shown that attention heads exhibit specialized and stable functional roles across layers, with later heads often encoding higher-level semantic information. Motivated by these findings, we posit the following.

Assumption 2 (Attention-head consistency under alignment). For a non-negligible measure of prompts $x \sim \mathcal{Q}_x$, there exists a subset of attention heads whose activation deviations between the prompt x and the prompt–candidate sequence $x \oplus y$, where $\oplus$ denotes sequence concatenation, reflect distributional alignment. Specifically, for candidates $y \in \mathcal{C}(x)$, activation deviations in these heads are correlated with $D\Big(\mathcal{Q}_x, \mathcal{Q}_x^{(y)}\Big)$, even when likelihood ordering under $p_\Theta(\cdot \mid x)$ is misaligned with task performance.

This assumption suggests that attention-head activation deviations carry information about candidate alignment. Better alignment may correspond to smaller deviations in some heads and larger deviations in others, motivating a learned scoring function that combines these head-specific signals to predict candidate quality.

Attention-head activation representation. Consider a transformer with L layers and H attention heads per layer. For sequence $z = (z_1, \ldots, z_T)$, let $\mathbf{H}_{\ell,h}(z,t) \in \mathbb{R}^{d_h}$ denote the output of head (ℓ, h) at position t after attention aggregation, where d_h is the per-head hidden dimensionality. We summarize head activity via

$$a_{\ell,h}(z) = \mathbb{E}_{t\sim\mathcal{T}_{\text{act}}}\left[\|\mathbf{H}_{\ell,h}(z,t)\|_2\right] \approx \frac{1}{|\mathcal{T}_{\text{act}}|}\sum_{t\in\mathcal{T}_{\text{act}}}\|\mathbf{H}_{\ell,h}(z,t)\|_2, \quad (8)$$

yielding an activation vector $\mathbf{a}(z) \in \mathbb{R}^{LH}$ where $\mathcal{T}_{\text{act}}$ denotes the set of token positions over which activations are measured, spanning the entire sequence from the prompt tokens through the generated tokens.

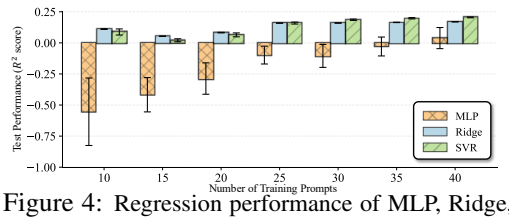


Figure 4: Regression performance of MLP, Ridge, and SVR versus training prompt count on 20 held-out Medicine prompts, using attention-head alignment features from Llama-3.2-3B-Instruct. Negative MLP R^2 indicates worse performance than predicting the mean target score, consistent with overfitting and instability under limited calibration data.

For prompt x and candidate y, we define the alignment feature $\mathbf{\Delta}(x,y) = \mathbf{a}(x \oplus y) - \mathbf{a}(x)$, and use squared components $\Delta_{\ell,h}(x,y)^2$ as fine-grained measures of representational deviation.

Observation 2 (Head alignment reflects candidate quality under distribution shift). Consistent with Assumption 2, Figure 2 (a) shows that individual attention-head alignment features exhibit strong positive and negative correlations with candidate–target similarity. The most pronounced correlations arise predominantly in later-layer heads, which are known to encode higher-level, task-relevant abstractions (Malfa et al., 2024). Moreover, these alignment signals are sufficiently structured to support prediction of candidate quality on held-out prompts using a lightweight regression model (Figure 2 (b)); the regression formulation and training procedure are introduced in the following Subsection 4.3. By contrast, Figure 3 shows that perplexity-based criteria—despite being a standard confidence signal for LLMs—exhibit weak or unstable correlation with candidate quality under the same distribution shift. These diagnostics motivate testing whether attention-head features carry selection information beyond likelihood; the controlled-pool comparison in Appendix I evaluates that question directly.

4.3 Distribution-Aware Head Alignment Learning

The alignment vector $\mathbf{\Delta}(x,y)$ provides a high-dimensional description of prompt–candidate activation differences. Motivated by Assumption 2, we learn how these differences relate to candidate quality through offline calibration. Our goal is therefore to learn a scalar scoring function that maps alignment features to an estimate of candidate quality. Concretely, we introduce a lightweight predictor

$$s(x,y) = g(\mathbf{\Delta}(x,y)), \quad (9)$$

where the score predicts the reference-based quality metric used during calibration. The predictor learns the joint contribution of heads whose correlations with quality can have either sign.

Accordingly, the idealized objective

$$\arg\min_{y\in\mathcal{C}(x)} D\left(\mathcal{Q}_x, \mathcal{Q}_x^{(y)}\right) \quad (10)$$

motivates the practical selection rule

$$\hat{y} = \arg\max_{y\in\mathcal{C}(x)} g(\mathbf{\Delta}(x,y)). \quad (11)$$

Algorithm 1: Distribution-Aware Attention-Head Alignment for Test-Time Adaptation

Require: Fixed LLM f_Θ; calibration set $\mathcal{S} = \{(x_i, y_i^\star)\}_{i=1}^M$; test prompt x
Ensure: Selected output $\hat{y}$

Calibration (offline).
Initialize training set $\mathcal{D} \leftarrow \emptyset$
for each $(x_i, y_i^\star) \in \mathcal{S}$ **do**
  Generate candidates $\mathcal{C}(x_i)$ via prefix search
  for each $y \in \mathcal{C}(x_i)$ **do**
    Compute alignment features $\mathbf{\Delta}(x_i, y)$
    Compute surrogate label $\ell(y, y_i^\star)$, e.g., ROUGE-LSUM w.r.t. $y_i^\star$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{\Delta}(x_i, y), \ell(y, y_i^\star))\}$
  end for
end for
Fit regressor g on $\mathcal{D}$
Test-time adaptation (per prompt).
Generate candidates $\mathcal{C}(x)$ via prefix search
for each $y \in \mathcal{C}(x)$ **do**
  Compute $\mathbf{\Delta}(x,y)$ and score $s_y \leftarrow g(\mathbf{\Delta}(x,y))$
end for
return $\hat{y} \leftarrow \arg\max_{y\in\mathcal{C}(x)} s_y$

Training data construction. To learn the predictor g, we introduce a small *calibration set* $\mathcal{S} = \{(x_i, y_i^\star)\}_{i=1}^N$ (e.g., $N \leq 64$), where each prompt x_i is paired with a reference target $y_i^\star$ available only during calibration. For each x_i, we generate a candidate set $\mathcal{C}(x_i)$ via prefix search and compute attention-head alignment vectors $\mathbf{\Delta}(x_i, y)$ for all $y \in \mathcal{C}(x_i)$. Because the distributional divergence $D(\mathcal{Q}_{x_i}, \mathcal{Q}_{x_i}^{(y)})$ is unobservable, we supervise learning using a task-level quality metric $\ell(y, y_i^\star)$ (ROUGE-LSUM in the main experiments), which supplies an empirical quality target rather than an observed distributional divergence. This construction yields a calibration dataset

$$\mathcal{D}_{\text{cal}} = \big\{(\mathbf{\Delta}(x_i, y),\, \ell(y, y_i^\star)) \;\big|\; (x_i, y_i^\star) \in \mathcal{S},\; y \in \mathcal{C}(x_i)\big\}, \tag{12}$$

which is used to train g to predict reference-based candidate quality.

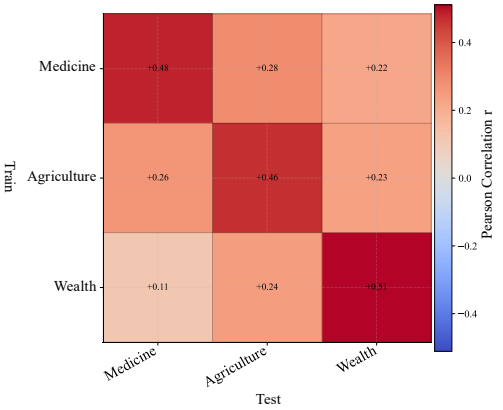


Figure 5: Transferability of distribution-aware attention-head alignment across domains, evaluated on Medicine, Agriculture, and Wealth datasets using Llama-3.2-3B-Instruct. For each domain, 60 prompts are sampled, with 40 used to train the SVR and 20 held out for testing. The diagonal entries show the strongest correlations when training and testing on the same domain. Notably, cross-domain evaluations still yield consistently positive correlations (Pearson $r \approx 0.2$), indicating that the attention-head alignment signal generalizes beyond the training domain.

Learning procedure. We train a regression model to map alignment features to scalar values that predict the calibration quality metric. Among the evaluated alternatives (Hoerl & Kennard, 1970; Smola & Schölkopf, 2004), Support Vector Regression (SVR) provides the most stable generalization in the low-data regime relevant to test-time adaptation, consistently outperforming Ridge regression and MLP baselines (Figure 4). The learned predictor aggregates distribution-aware attention-head signals into a robust estimate of candidate alignment, enabling effective selection without modifying LLM parameters. Both g and Θ remain fixed at deployment; offline fitting of g is the only supervised learning stage.

Observation 3 (Cross-domain generalization). As shown in Figure 5, an SVR trained on one domain retains positive predictive correlation when evaluated on others. While within-domain transfer is strongest, positive cross-domain correlations suggest that the learned aggregation captures transferable information about candidate quality.

Together, Observation 2 and Observation 3 provide empirical support for the motivation behind Assumption 2 and for learning a prompt-agnostic aggregation of attention-head alignment signals. Algorithm 1 summarizes the full procedure.

5 Experiments

5.1 Experimental Settings

Evaluation protocol. We study *per-sample test-time adaptation*, where each test input is processed independently and may come from a different domain. All methods use an effective test-time batch size of one. Any adapted parameters, optimizer state, or accumulated statistics are reset before the next test instance, so no adaptation state is shared across queries. This protocol evaluates adaptation to an individual input without assuming a coherent target stream. Adaptation at deployment uses only the current unlabeled query, without additional target-domain training examples or a retrieval corpus.

Baselines. Greedy decoding provides the unadapted reference for assessing the overall benefit of our exploration and selection procedure. For adaptive comparisons, we consider representative entropy- or perplexity-based methods: Tent (Wang et al., 2021), EATA (Niu et al., 2022), COME (Zhang et al., 2025), and TLM (Hu et al., 2025). These methods derive self-supervised learning signals from test inputs and the deployed model's predictions, allowing adaptation without a retrieval corpus or an auxiliary pretrained evaluator. They therefore provide natural comparisons for resource-constrained deployment. We use their LLM implementations from Hu et al. (2025) under the same episodic protocol, restricting adaptation to the current query and resetting adaptation state between requests.

Table 2: Comparison on Domain-Specific and Instruction-Following datasets. Each cell reports the absolute score, with the improvement over the corresponding vanilla LLM in parentheses. The **best** and **second-best** results *within each backbone block* are highlighted. Average Improvement columns report mean Δ over Domain-Specific (4), Instruction-Following (2), and Overall (6) datasets.

Method	Domain-Specific				Instruction-Following		Average Improvement (Δ)		
	Agriculture	Finance	Geography	Medicine	Alpaca-GPT4	InstructionWild	Domain	Instruction	Overall
Llama-3.2-3B-Instruct	0.0850 (Δ+0.0000)	0.2229 (Δ+0.0000)	0.2395 (Δ+0.0000)	0.1411 (Δ+0.0000)	**0.3564 (Δ+0.0000)**	0.2562 (Δ+0.0000)	+0.0000	+0.0000	+0.0000
Tent	0.0842 (Δ-0.0008)	0.2190 (Δ-0.0039)	**0.2423 (Δ+0.0028)**	0.1425 (Δ+0.0014)	0.3520 (Δ-0.0044)	0.2627 (Δ+0.0065)	-0.0001	+0.0010	+0.0002
EATA	0.0838 (Δ-0.0012)	0.2180 (Δ-0.0049)	0.2412 (Δ+0.0017)	0.1431 (Δ+0.0020)	0.3527 (Δ-0.0037)	**0.2646 (Δ+0.0084)**	-0.0006	**+0.0023**	+0.0003
COME	0.0835 (Δ-0.0015)	0.2203 (Δ-0.0026)	0.2406 (Δ+0.0011)	0.1427 (Δ+0.0016)	0.3467 (Δ-0.0097)	0.2315 (Δ-0.0247)	-0.0004	-0.0172	-0.0059
TLM	**0.0877 (Δ+0.0027)**	**0.2247 (Δ+0.0018)**	0.2382 (Δ-0.0013)	**0.1436 (Δ+0.0025)**	0.3540 (Δ-0.0024)	0.2632 (Δ+0.0070)	**+0.0014**	**+0.0023**	**+0.0017**
Ours	**0.1257 (Δ+0.0407)**	**0.2592 (Δ+0.0363)**	**0.2460 (Δ+0.0065)**	**0.1723 (Δ+0.0312)**	**0.3838 (Δ+0.0274)**	**0.2928 (Δ+0.0366)**	**+0.0287**	**+0.0320**	**+0.0298**
Llama-3-8B-Instruct	0.0834 (Δ+0.0000)	0.2329 (Δ+0.0000)	0.2450 (Δ+0.0000)	0.1265 (Δ+0.0000)	0.3752 (Δ+0.0000)	0.2608 (Δ+0.0000)	+0.0000	+0.0000	+0.0000
Tent	0.0833 (Δ-0.0001)	0.2327 (Δ-0.0002)	0.2539 (Δ+0.0089)	0.1292 (Δ+0.0027)	**0.3827 (Δ+0.0075)**	0.2670 (Δ+0.0062)	+0.0028	+0.0068	+0.0041
EATA	0.0834 (Δ+0.0000)	0.2329 (Δ+0.0000)	0.2549 (Δ+0.0099)	0.1291 (Δ+0.0026)	0.3821 (Δ+0.0069)	**0.2682 (Δ+0.0074)**	+0.0031	+0.0071	+0.0044
COME	0.0839 (Δ+0.0005)	0.2321 (Δ-0.0008)	0.2545 (Δ+0.0095)	0.1286 (Δ+0.0021)	0.3794 (Δ+0.0042)	0.2427 (Δ-0.0181)	+0.0028	-0.0069	-0.0004
TLM	**0.0858 (Δ+0.0024)**	**0.2427 (Δ+0.0098)**	**0.2574 (Δ+0.0124)**	**0.1298 (Δ+0.0033)**	**0.3846 (Δ+0.0094)**	0.2670 (Δ+0.0062)	**+0.0070**	**+0.0078**	**+0.0072**
Ours	**0.1030 (Δ+0.0196)**	**0.2553 (Δ+0.0224)**	**0.2665 (Δ+0.0215)**	**0.1773 (Δ+0.0508)**	0.3745 (Δ-0.0007)	**0.2776 (Δ+0.0168)**	**+0.0286**	**+0.0080**	**+0.0217**
Qwen2.5-7B-Instruct	**0.0981 (Δ+0.0000)**	**0.2739 (Δ+0.0000)**	**0.2649 (Δ+0.0000)**	**0.1313 (Δ+0.0000)**	**0.4439 (Δ+0.0000)**	**0.2866 (Δ+0.0000)**	**+0.0000**	**+0.0000**	**+0.0000**
Tent	0.0936 (Δ-0.0045)	0.2551 (Δ-0.0188)	0.2587 (Δ-0.0062)	0.1275 (Δ-0.0038)	0.4296 (Δ-0.0143)	0.2835 (Δ-0.0031)	-0.0083	-0.0087	-0.0084
EATA	0.0930 (Δ-0.0051)	0.2553 (Δ-0.0186)	0.2573 (Δ-0.0076)	0.1269 (Δ-0.0044)	0.4289 (Δ-0.0150)	0.2854 (Δ-0.0012)	-0.0089	-0.0081	-0.0086
COME	0.0945 (Δ-0.0036)	0.2549 (Δ-0.0190)	0.2563 (Δ-0.0086)	0.1266 (Δ-0.0047)	0.4242 (Δ-0.0197)	0.2561 (Δ-0.0305)	-0.0090	-0.0251	-0.0143
TLM	0.0946 (Δ-0.0035)	0.2592 (Δ-0.0147)	0.2620 (Δ-0.0029)	0.1270 (Δ-0.0043)	0.4275 (Δ-0.0164)	0.2836 (Δ-0.0030)	-0.0064	-0.0097	-0.0074
Ours	**0.1164 (Δ+0.0183)**	**0.2929 (Δ+0.0190)**	**0.2768 (Δ+0.0119)**	**0.1567 (Δ+0.0254)**	**0.4866 (Δ+0.0427)**	**0.2960 (Δ+0.0094)**	**+0.0186**	**+0.0260**	**+0.0211**

Table 3: Effect of calibration corpus choice on Domain-Specific and Instruction-Following datasets. Each cell reports the absolute score with improvement over the vanilla LLM in parentheses. **Best** and **second-best** results are highlighted.

Method	Domain-Specific				Instruction-Following		Average Improvement (Δ)		
	Agriculture	Finance	Geography	Medicine	Alpaca-GPT4	InstructionWild	Domain	Instruction	Overall
Llama-3.2-3B-Instruct	0.0850 (Δ+0.0000)	0.2229 (Δ+0.0000)	0.2395 (Δ+0.0000)	0.1411 (Δ+0.0000)	0.3564 (Δ+0.0000)	0.2562 (Δ+0.0000)	+0.0000	+0.0000	+0.0000
Ours (fusechat_v1)	**0.1257 (Δ+0.0407)**	**0.2592 (Δ+0.0363)**	**0.2460 (Δ+0.0065)**	**0.1723 (Δ+0.0312)**	**0.3838 (Δ+0.0274)**	0.2928 (Δ+0.0366)	**+0.0287**	**+0.0320**	**+0.0298**
Ours (open-perfectblend)	**0.1340 (Δ+0.0490)**	**0.2592 (Δ+0.0363)**	**0.2463 (Δ+0.0068)**	**0.1667 (Δ+0.0256)**	**0.3838 (Δ+0.0274)**	**0.3030 (Δ+0.0468)**	**+0.0294**	**+0.0371**	**+0.0320**
Ours (tulu-3-sft-mixture)	0.1206 (Δ+0.0356)	**0.2592 (Δ+0.0363)**	**0.2460 (Δ+0.0065)**	**0.1681 (Δ+0.0270)**	**0.3838 (Δ+0.0274)**	**0.2944 (Δ+0.0382)**	+0.0264	**+0.0328**	+0.0285

This compares response selection with parameter adaptation when neither can accumulate evidence from a coherent target distribution.

To isolate the contribution of the attention-head signal, we compare our scorer with random selection, minimum perplexity, and a regressor using likelihood and length features on identical candidate pools (Appendix I). These controlled comparisons assess selection quality while holding candidate generation fixed, complementing the main evaluation of complete per-query adaptation procedures.

Datasets. The main evaluation compares our method, greedy decoding, and the four update-based baselines on six benchmark datasets spanning domain-specific and instruction-following settings. Specifically, we consider four domain-specific datasets—*Agriculture* (Ashraf, 2024), *Finance* (Bharti, 2023), *Geography* (Deng et al., 2023), and *Medicine* (Wang, 2023)—covering specialized terminology and domain knowledge. In addition, we evaluate on two instruction-following datasets—*Alpaca-GPT4* (Peng et al., 2023) and *InstructionWild* (Ni et al., 2023)—which cover diverse user intents and instruction styles. Following prior work (Hu et al., 2025), we randomly sample 5,000 instances from each dataset and report ROUGE-LSUM (Lin, 2004), which measures textual overlap between generated responses and reference answers.

Backbone models. We conduct experiments using three pretrained instruction-tuned LLMs with different sizes and architectures: *Llama-3.2-3B-Instruct*, *Llama-3-8B-Instruct* (Dubey et al., 2024), and *Qwen2.5-7B-Instruct* (Qwen et al., 2025). These backbones represent model families and parameter scales studied for local and edge inference on consumer and embedded hardware (Ruan et al., 2024; Jang & Morabito, 2025), making them relevant to resource-constrained deployment. Our method keeps all model parameters fixed, while the update-based baselines perform parameter-efficient adaptation through LoRA (Hu et al., 2022).

Implementation details. Baseline hyperparameters follow Hu et al. (2025), with adaptation applied independently to each test query under the episodic protocol described above. Our method uses a single hyperparameter configuration across datasets and backbones unless noted otherwise, with a separately calibrated scorer for each backbone. Candidate generation uses prefix search with depth

$k = 4$ and branching factor $n = 3$, followed by greedy completion of each branch, producing $3^4 = 81$ candidates per input. Attention-head activations are computed over all token positions $\mathcal{T}_{\text{act}}$ spanning the prompt and generated text. During offline calibration, we randomly sample 64 prompt–reference pairs, guided by the rapid saturation observed in Figure 4. Calibration data are drawn from general-purpose datasets—*fusechat_v1* (Wan et al., 2024), *open-perfectblend* (Xu et al., 2024), and *tulu-3-sft-mixture* (Lambert et al., 2024)—with *fusechat_v1* used by default. The scorer remains fixed throughout test-time evaluation. All experiments were conducted on an NVIDIA A100 GPU.

5.2 Experimental Results

Can test-time adaptation improve performance without model updates? Table 2 shows that our method improves performance over the corresponding vanilla LLMs in 17 of the 18 backbone–dataset combinations, while performing *no test-time parameter updates*. Across domain-specific and instruction-following datasets, our approach achieves average ROUGE-LSUM improvements of $\Delta{+}0.0298$, $\Delta{+}0.0217$, and $\Delta{+}0.0211$ for *Llama-3.2-3B-Instruct*, *Llama-3-8B-Instruct*, and *Qwen2.5-7B-Instruct*, respectively. These gains are obtained through structured candidate generation and selection guided by attention-head alignment. The results show that per-query response selection can improve the frozen model's output quality, supporting test-time adaptation through additional forward computation without updating model parameters.

Are test-time model updates effective in independent per-sample settings? We next compare against Tent, EATA, COME, and TLM, all evaluated under the same independent per-sample protocol. As shown in Table 2, these methods yield mixed results, improving performance on some backbone–dataset combinations while degrading it on others. For *Qwen2.5-7B-Instruct*, all four update-based baselines reduce overall mean performance, with declines as large as $\Delta{-}0.0143$, whereas our method achieves a gain of $\Delta{+}0.0211$. Our method achieves the largest overall mean gain for each backbone in this comparison. These findings indicate that the evaluated parameter-update methods do not consistently improve performance when adaptation relies on a single query. They support response selection as an effective alternative when independent test queries provide no opportunity to accumulate adaptation evidence across a coherent target stream.

Is our method sensitive to the calibration dataset? Table 3 evaluates the effect of the calibration source on performance using *Llama-3.2-3B-Instruct*. Keeping the calibration budget fixed at 64 prompt–target pairs, we compare three generic instruction datasets: *fusechat_v1*, *open-perfectblend*, and *tulu-3-sft-mixture*. All three calibration sources yield improvements across the six evaluation datasets, with overall average gains ranging from $\Delta{+}0.0285$ to $\Delta{+}0.0320$. These comparable gains suggest limited sensitivity to the choice among the evaluated calibration sources, consistent with our earlier observation that attention-head alignment carries transferable signals across domains. The results support using a small generic calibration set for per-query adaptation without requiring target-domain calibration data.

6 Conclusion

We introduced a framework for per-sample test-time adaptation of LLMs under resource-constrained deployment, where independent queries provide no reliable access to a coherent target-domain corpus. Our method combines structured candidate exploration with attention-head alignment, using a lightweight scorer learned from a small, generic reference-labeled calibration set. At deployment, both the generator and scorer remain fixed, and each query is processed without retrieval, a separately pretrained evaluator, or test-time parameter updates. Experiments across domain-specific and instruction-following tasks demonstrate performance gains over greedy decoding in nearly all evaluated settings and larger overall mean improvements than the evaluated episodic update baselines for each backbone. These results highlight that internal model signals can support effective per-query response selection, offering a forward-only approach to TTA without assuming coherence across test queries.

Limitations and future work. Our method improves reference-based generation quality but does not guarantee factual, mathematical, or logical correctness. Future work can investigate complementary signals from the frozen model that more directly assess factual accuracy and reasoning validity during candidate selection.

AI USE STATEMENT

In preparing this manuscript, we used generative AI tools to assist with editing author-written text, including grammar correction, typo detection, and improvements to clarity and readability. The authors reviewed and revised the suggested edits to preserve the intended meaning and ensure consistency with the paper's technical content. We take responsibility for the final content of this work, including its text, claims, and any artifacts prepared with the assistance of generative AI tools.

ETHICS STATEMENT

This work studies test-time adaptation using publicly available datasets and pretrained models. We emphasize responsible use of these resources, with appropriate consideration of data privacy, fairness, applicable licenses, and the potential societal impacts of LLM deployment.

REPRODUCIBILITY STATEMENT

The main paper describes the proposed method, evaluation protocol, and implementation settings. The appendices provide additional dataset descriptions, ablation studies, and experimental analyses. Together, these materials document the procedures used to obtain the reported results and support their reproduction.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *International Conference on Machine Learning (ICML)*, 2025.

Keivan Alizadeh, Seyed Iman Mirzadeh, Dmitry Belenko, S Khatamifard, Minsik Cho, Carlo C Del Mundo, Mohammad Rastegari, and Mehrdad Farajtabar. Llm in a flash: Efficient large language model inference with limited memory. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12562–12584, 2024.

Farid Ariai, Joel Mackenzie, and Gianluca Demartini. Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges. *ACM Computing Surveys*, 58(6): 1–37, 2025.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2023.

Mohammed Ashraf. agriculture-qa-english-only. `https://huggingface.co/datasets/KisanVaani/agriculture-qa-english-only`, 2024.

Gábor Bella, Paula Helm, Gertraud Koch, and Fausto Giunchiglia. Tackling language modelling bias in support of linguistic diversity. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 562–572, 2024.

Gaurang Bharti. wealth-alpaca_lora. `https://huggingface.co/datasets/gbharti/wealth-alpaca_lora`, 2023.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy yong Sohn, and Alejandro Lopez-Lira. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2504.15800.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019.

Cheng Deng, Tianhang Zhang, Zhongmou He, Yi Xu, Qiyuan Chen, Yuanyuan Shi, Luoyi Fu, Weinan Zhang, Xinbing Wang, Chenghu Zhou, Zhouhan Lin, and Junxian He. K2: A foundation language model for geoscience knowledge understanding and utilization, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. Frustratingly easy test-time adaptation of vision-language models. *Advances in Neural Information Processing Systems*, 37:129062–129093, 2024.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1): 1–23, 2021.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. *International Conference on Learning Representations (ICLR)*, 2024.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

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

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. In *International Conference on Machine Learning (ICML)*, 2025.

SiYoung Jang and Roberto Morabito. Edge-first language model inference: Models, metrics, and tradeoffs. In *2025 IEEE 45th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 309–314. IEEE, 2025.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. URL https://arxiv.org/abs/2411.15124.

Stefanos Laskaridis, Kleomenis Katevas, Lorenzo Minto, and Hamed Haddadi. Melting point: Mobile evaluation of language transformers. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 890–907, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Ethan Li, Anders Boesen Lindbo Larsen, Chen Zhang, Xiyou Zhou, Jun Qin, Dian Ang Yap, Narendran Raghavan, Xuankai Chang, Margit Bowler, Eray Yildiz, et al. Apple Intelligence Foundation Language Models: Tech Report 2025. *arXiv preprint arXiv:2507.13575*, 2025a. URL https://arxiv.org/abs/2507.13575.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See Kiong Ng, Tat-Seng Chua, and Yang Deng. Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5131–5157, 2025b.

Xinlong Li, Weijieying Ren, Wei Qin, Lei Wang, Tianxiang Zhao, and Richang Hong. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025c.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Xueqi Ma, Jun Wang, Yanbei Jiang, Sarah Monazam Erfani, Tongliang Liu, and James Bailey. Cognitive mirrors: Exploring the diverse functional roles of attention heads in llm reasoning. In *Advances in neural information processing systems*, 2025.

Emanuele La Malfa, Gabriele La Malfa, Giuseppe Nicosia, and Vito Latora. Deep neural networks via complex network theory: a perspective. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 4361–4369, 2024.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

Jinjie Ni, Fuzhao Xue, Kabir Jain, Mahir Hitesh Shah, Zangwei Zheng, and Yang You. Instruction in the wild: A user-based instruction dataset. https://github.com/XueFuzhao/InstructionWild, 2023.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Charlie F Ruan, Yucheng Qin, Akaash R Parthasarathy, Xun Zhou, Ruihang Lai, Hongyi Jin, Yixin Dong, Bohan Hou, Meng-Shiun Yu, Yiyan Zhai, et al. WebLLM: A high-performance in-browser LLM inference engine. *arXiv preprint arXiv:2412.15803*, 2024.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. The prompt report: A systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*, 2024.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, 2020. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704/.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004. doi: 10.1023/B:STCO.0000035301.49549.88.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

Fanqi Wan, Ziyi Yang, Longguang Zhong, Xiaojun Quan, Xinting Huang, and Wei Bi. Fusechat: Knowledge fusion of chat models. *arXiv preprint arXiv:2402.16107*, 2024.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Rongsheng Wang. Genmedgpt-5k-en. https://huggingface.co/datasets/wangrongsheng/GenMedGPT-5k-en, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, et al. The perfect blend: Redefining rlhf with mixture of judges. *arXiv preprint arXiv:2409.20370*, 2024.

Maxime Zanella, Clément Fuchs, Christophe De Vleeschouwer, and Ismail Ben Ayed. Realistic test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25103–25112, 2025.

Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. Come: Test-time adaption by conservatively minimizing entropy. In *International Conference on Learning Representations (ICLR)*, 2025.

Terence Zhang and Adrien Couque. An introduction to privacy and safety for Gemini Nano. Android Developers Blog, October 2024. URL https://android-developers.googleblog.com/2024/10/introduction-to-privacy-and-safety-gemini-nano.html.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1904.09675.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. On the role of attention heads in large language model safety. In *International Conference on Learning Representations (ICLR)*, 2025.

APPENDIX

A DATASET DETAILS

We evaluate all methods on six benchmark datasets spanning domain-specific knowledge and general instruction following. Together, these datasets cover both *vertical domain shifts* and *instructional diversity*, reflecting realistic deployment scenarios for large language models.

Domain-specific datasets. Domain-specific datasets consist of four datasets designed to evaluate adaptation to specialized terminology and expert knowledge:

- **Agriculture** (Ashraf, 2024) focuses on agricultural science and practice, including crop management, soil conditions, and farming techniques. Prompts frequently contain domain-specific vocabulary that is underrepresented in general-purpose pretraining corpora.
- **Finance** (Bharti, 2023) contains financial analysis and economic reasoning prompts, such as investment strategies, accounting concepts, and macroeconomic explanations, requiring precise handling of technical financial language.
- **Geography** (Deng et al., 2023) covers physical and human geography topics, including climate, geology, and geopolitical descriptions, testing factual consistency under domain shift.
- **Medicine** (Wang, 2023) targets biomedical and clinical knowledge, with prompts involving diseases, treatments, and medical reasoning, representing a challenging expert domain with high stakes for semantic accuracy.

These four datasets collectively assess a model's ability to adapt to *vertical domain shifts*, where the test distribution differs substantially from the model's pretraining data in both vocabulary and conceptual structure.

Instruction-following datasets. Instruction-following datasets evaluate robustness to diverse instruction styles and user intents:

- **Alpaca-GPT4** (Peng et al., 2023) is a high-quality instruction-following dataset generated using GPT-4, covering a wide range of tasks such as summarization, explanation, and reasoning. It reflects well-structured, canonical instruction prompts.
- **InstructionWild** (Ni et al., 2023) contains naturally occurring, less-curated instructions collected from real user interactions. Its format aligns with the Alpaca dataset, and it covers common instruction fine-tuning tasks such as QA and summarization.

These datasets primarily evaluate adaptation under *instructional and stylistic distribution shifts*, rather than specialized domain knowledge.

B GENERAL-PURPOSE CALIBRATION CORPORA DETAILS

In addition to domain- and instruction-specific benchmarks, we rely on general-purpose corpora during the *calibration phase* of our method. These corpora are used exclusively to identify attention heads whose activation shifts correlate with output quality, and are *not* used for LLM fine-tuning or parameter updates. We deliberately select general-purpose datasets with broad task coverage, heterogeneous instruction styles, and diverse linguistic realizations, in order to avoid reliance on any specific domain or distribution during calibration.

Fusechat_v1. Fusechat_v1 (Wan et al., 2024) is a comprehensive instruction-following corpus designed to cover a wide range of conversational styles and model capabilities. The dataset contains both human-written and model-generated samples, spanning tasks such as open-domain dialogue, instruction following, reasoning, and knowledge-based responses. Its mixture of sources and generation styles makes it well-suited for calibration, as it exposes the model to heterogeneous prompt structures and response patterns without emphasizing any single domain or task distribution.

Open-perfectblend. Open-perfectblend is an open reproduction of the dataset described in (Xu et al., 2024). The dataset is constructed as a general-purpose instruction mixture that includes chat-style interactions, mathematical reasoning, code generation, and structured instruction-following tasks. It is designed to support multi-task post-training and evaluation, and provides well-curated, high-quality instruction–response pairs. We use open-perfectblend as a calibration corpus due to its balanced task composition and canonical instruction formats, which serve as a stable reference distribution for attention-head activation analysis.

Tulu-3-sft-mixture. The tulu-3-sft-mixture (Lambert et al., 2024) is a large-scale supervised fine-tuning corpus used to train the Tulu 3 family of models. The dataset contains 939,344 instruction–response samples drawn from a diverse mixture of public datasets and newly generated synthetic data. It includes both natural and persona-conditioned instructions, covering areas such as mathematics, programming, reasoning, safety, and general instruction following. Each example is stored in a message-based chat format and annotated with its source dataset.

Role in Calibration. All three corpora are used only during the calibration stage to estimate statistics of attention-head activation shifts and their relationship to output quality. The scorer is fitted jointly on the offline calibration examples. At deployment, it is fixed and no adaptation state or updates are shared across test queries, preserving single-instance operation.

C Additional Experiments on the Practicality of Existing TTA Methods

Our main experiments evaluate all methods under the isolated per-sample protocol described in Subsection 5.1. This protocol is stricter than domain-level adaptation settings because each test instance is processed independently and no adaptation state is shared across examples. To examine whether update-based baselines improve when granted a stronger assumption, we additionally evaluate them with same-domain test batches of size 4, 8, and 32. This setting is outside our primary formulation, but it provides a useful stress test of whether entropy- or perplexity-based adaptation objectives benefit from larger same-domain batches in LLM generation tasks.

Table 4: Update-based baselines with same-domain test batches using Llama-3.2-3B-Instruct. Scores are ROUGE-LSUM.

Dataset	Method	Batch size		
		4	8	32
Agriculture	TLM	0.0925	0.0928	0.0935
	TENT	0.0919	0.0916	0.0917
	EATA	0.0907	0.0923	0.0921
	COME	0.0913	0.0887	0.0914
Finance	TLM	0.2302	0.2302	0.2310
	TENT	0.2306	0.2303	0.2322
	EATA	0.2311	0.2311	0.2313
	COME	0.2307	0.2233	0.2307

Even when update-based baselines are granted access to multiple same-domain samples, their improvements remain small as shown in Table 4. Increasing coherent batch size provides only limited gains within the evaluated range.

D Evaluation on Post-Release Benchmarks

A potential concern with public benchmarks is that some examples may overlap with proprietary pretraining or instruction-tuning data. To reduce this concern, we additionally evaluate on HealthBench (Arora et al., 2025) and FinDER (Choi et al., 2025), which were released after Llama-3.2-3B-Instruct. For both datasets, we retain examples with a single prompt and a single answer, and randomly sample 400 examples from HealthBench and 150 examples from FinDER. These experiments test

whether the observed gains persist on the selected post-release subsets, using ROUGE-LSUM rather than the benchmarks' full evaluation protocols.

Table 5: Evaluation on post-release benchmarks with reduced contamination risk using Llama-3.2-3B-Instruct.

Dataset	Base	COME	EATA	TENT	TLM	Ours
HealthBench (Arora et al., 2025)	0.1980	0.1971	0.2084	0.2093	0.2095	**0.2363**
FinDER (Choi et al., 2025)	0.2086	0.2076	0.2083	0.2092	0.2143	**0.2251**

As shown in Table 5, our method improves over the base model and all update-based baselines on both datasets.

E Runtime and Computational Cost

The main computational overhead of our method comes from prefix-search candidate generation and attention-head activation extraction. Unlike update-based TTA methods, our method is fully forward-only: it does not compute gradients, maintain optimizer states, or update model parameters. After candidate generation, attention-feature extraction and scoring can be batched when memory permits, and the SVR scoring cost is negligible compared with LLM forward passes.

Let $B = n^k$ denote the number of prefix-search candidates. The default setting in the main experiments uses $n = 3$ and $k = 4$, yielding $B = 81$ candidates. However, as shown in Table 6, a smaller setting with $n = 3, k = 3$ achieves nearly the same score with lower overhead.

Table 6: Average runtime overhead per sample on Medicine using Llama-3.2-3B-Instruct.

Method	Runtime overhead	Score
Llama-3.2-3B-Instruct	$1.00\times$	0.1411
Ours $(n = 3, k = 3)$	$\mathbf{1.32\times}$	**0.1725**
Ours $(n = 3, k = 4)$	$2.10\times$	0.1723
TENT	$1.49\times$	0.1425

The $n = 3, k = 3$ configuration provides a favorable quality–efficiency tradeoff: it substantially improves over the base model and TENT while requiring lower runtime overhead than the default $n = 3, k = 4$ setting. This indicates that the method does not require the largest candidate budget to be effective.

F Prefix-Search Budget Ablation

We evaluate the effect of prefix depth k and branching factor n on the Medicine dataset using Llama-3.2-3B-Instruct. In addition to the adapted score obtained by our learned selector, we report an oracle "optimal selection" score, which selects the best candidate in hindsight using the reference. This oracle is not available at test time, but it indicates the quality of the candidate pool exposed by prefix search.

Table 7: Ablation over prefix-search settings. Optimal selection denotes the maximum possible score obtained by selecting the best candidate in hindsight.

Setting	Adapted score	Optimal selection
$n = 3, k = 1$ (3 candidates)	0.1688	0.1802
$n = 3, k = 2$ (9 candidates)	0.1706	0.2009
$n = 3, k = 3$ (27 candidates)	**0.1725**	0.2215
$n = 3, k = 4$ (81 candidates)	0.1723	**0.2480**
$n = 4, k = 1$ (4 candidates)	0.1688	0.1816
$n = 4, k = 2$ (16 candidates)	0.1706	0.2114
$n = 4, k = 3$ (64 candidates)	0.1701	0.2436

As shown in Table 7, the optimal-selection score increases as the candidate pool grows, indicating that larger prefix-search budgets expose better candidates. However, the adapted score saturates much earlier, suggesting that the current bottleneck is not only candidate discovery but also candidate ranking. This motivates future work on advanced alignment scoring methods while also showing that moderate candidate budgets already provide most of the realized gain.

G Comparison with Temperature Sampling

To further test whether our gains arise merely from generating many candidates, we compare prefix search with temperature-based sampling while fixing the number of candidates to 81. The selector and evaluation protocol are kept unchanged.

Table 8: Candidate-generation ablation using Llama-3.2-3B-Instruct with 81 generated candidates.

Dataset	Temperature sampling	Prefix search	Δ
Medicine	0.1506	**0.1723**	+0.0217
Geo	0.2085	**0.2460**	+0.0375
Finance	0.2421	**0.2592**	+0.0171

As shown in Table 8, prefix search consistently outperforms temperature sampling under the same candidate budget, further supporting our Observation 1. Thus, the improvement is not simply a best-of-many effect; structured exploration of early decoding decisions is important for reliably exposing useful candidates.

H Robustness to Calibration Surrogate Metrics

Our main experiments use ROUGE-LSUM as the calibration surrogate since it provides a reference-based approximation to candidate quality. To test whether the learned selector is sensitive to this specific surrogate, we train the selector using alternative metrics while keeping the final evaluation protocol fixed.

Table 9: Cross-surrogate evaluation with Llama-3.2-3B-Instruct. Each column indicates the metric used as the calibration surrogate.

Dataset	ROUGE-LSUM	BLEURT	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Agriculture	0.1257	0.1255	0.1256	0.1257	0.1257	0.1256
Finance	0.2592	0.2593	0.2593	0.2592	0.2593	0.2592

As shown in Table 9, the final performance is nearly unchanged across calibration surrogates. This suggests that the learned attention-head selector is not tightly coupled to ROUGE-LSUM as a training signal.

I Controlled Candidate-Selection Comparisons

The main evaluation compares complete per-query adaptation procedures. Here, we hold candidate generation fixed to assess the contribution of attention-head features to response selection. All selectors operate on identical stored pools of 81 candidates from *Llama-3.2-3B-Instruct*, using subsets of approximately 500 examples per dataset. The learned selectors share a fresh calibration set of 64 reference-labeled *fusechat_v1* prompts and remain fixed across all six evaluation datasets. Each query is processed independently, using information from the current prompt, its candidates, and the frozen generator. The evaluation subsets and calibration draw differ from the main experiments, so absolute scores should be compared within this controlled analysis.

Selection criteria. We compare attention-head features with three controls. Random selection reports the exact expected score under uniform selection from each candidate pool. Minimum perplexity selects the candidate with the lowest perplexity under the frozen generator. To distinguish

the contribution of head features from that of offline supervision, we also train an SVR using log-perplexity, negative log-likelihood, and character and word lengths. This baseline uses the same calibration data, regression family, and cross-validation protocol as our scorer. Hyperparameters are selected through grouped cross-validation on calibration prompts, keeping candidates from the same prompt within one group. A reference oracle additionally selects the candidate with the highest reference-based score, providing an upper bound for the given pool and evaluation metric. The oracle requires test references and is used only for offline analysis.

Table 10: ROUGE-LSUM on identical 81-candidate pools. Avg. denotes the mean across the six datasets. The reference oracle provides an upper bound for the given pools.

Selector	Medicine	Agriculture	Finance	Geography	Alpaca	InstWild	Avg.
Expected random	.1494	.1078	.2441	.2319	.3550	.2790	.2279
Minimum perplexity	.1250	.0915	.2204	.2128	.3327	.2432	.2043
SVR (likelihood and length)	.1619	.1142	.2503	.2334	.3616	.2848	.2344
Ours (head alignment)	.1568	.1071	.2584	.2421	.3792	.2956	.2399
Reference oracle	.2474	.1911	.3799	.3839	.5419	.3622	.3511

Semantic evaluation. To assess whether the selection gains extend beyond ROUGE-LSUM, we evaluate the same outputs with BERTScore-F1 (Zhang et al., 2020), using DeBERTa-xlarge-MNLI with baseline rescaling, and BLEURT-20 (Sellam et al., 2020). The models used to compute these metrics are applied only during offline evaluation and do not participate in candidate selection. Table 11 reports unweighted averages calculated from the six reported per-dataset scores, which were rounded to four decimal places. Our scorer achieves higher average scores than both random selection and the likelihood and length SVR under both metrics. These results suggest that the benefit of attention-head features extends beyond the ROUGE-LSUM evaluation used for calibration. The metrics assess similarity to reference responses and do not establish factual, mathematical, or logical correctness.

Table 11: Semantic evaluation on the controlled candidate pools. Values are six-dataset averages calculated from the reported rounded scores. Higher is better.

Selector	BERTScore-F1	BLEURT-20
Expected random	.0906	.4895
SVR (likelihood and length)	.0999	.4843
Ours (head alignment)	.1043	.4967

J Qualitative Examples

We provide representative qualitative examples from each dataset. For each example, we show the input prompt, the ground-truth target, the lowest-perplexity generation, and the top-3 candidates selected during test-time exploration. For each candidate, we report its ROUGE-LSUM score with respect to the target and its perplexity. These examples illustrate that lower perplexity does not necessarily correlate with higher task quality.

J.1 Agriculture

Input

how can mulching be used to prevent or control couch grass

Target

use a thick layer of organic mulch, such as straw or leaves, to smother the grass. This method works by blocking sunlight and preventing the grass from growing. It is important to use a thick enough layer of mulch to be effective.

Lowest-Perplexity Generation

How does someone who is someone who is someone who is

Top-3 Candidates

Candidate 1
Use an analytical technical tool that could be used to analyze the following data\nA company's digital marketing campaigns.
ROUGE-LSUM: 0.4444 | Perplexity: 2.04

Candidate 2
A technical toolkit that could be used to analyze the following data\nA company's digital marketing campaigns.
ROUGE-LSUM: 0.4230 | Perplexity: 2.73

Candidate 3
Name: A technical tool that could be used to analyze the following data\nA company's digital marketing campaigns.
ROUGE-LSUM: 0.4150 | Perplexity: 2.11

J.3 GEOGRAPHY

Input

Estimation of Ground water yield for future development\nHow are you? 1. Let take a groundwater boreholes for urban areas, that drilled before 10 years, how can I estimate its yielding capacity to forecast for future, depending on the data that recorded during the drilling period of that well. How much its yielding capacity decline throughout the design period of this well? or is that possible to use the "safe yield" or is there any finding ways? formulas? guidelines?\nWith regard

Target

For the estimation of groundwater yield, pump test is usual process which gives useful information about the groundwater potential.

Lowest-Perplexity Generation

Estication of the Estication of the Estication of the

Candidate 3
1. Let take a groundwater boreholes for urban areas, that drilled before 10 years, how can I estimate its yielding capacity decline throughout the design period of this well? or is that possible to use the "safe yield" or is there any finding ways? formulas? formulas? formulas? formulas?
ROUGE-LSUM: 0.1764 | Perplexity: 2.78

J.4 MEDICINE

Input

If you are a doctor, please answer the medical questions based on the patient's description.\nDoctor, I'm experiencing some strange symptoms. I've been acting hostile towards my family, and I'm having problems with my movement. Also, I've noticed that I've been slurring my words lately.

Target

Based on what you've described, it sounds like you may be experiencing delirium, which is a state of mental confusion that can cause changes in behavior, movement problems and difficulty speaking clearly. We'll need to run some tests to determine the underlying cause, but it's important that we act quickly to manage your symptoms and prevent any complications.

Lowest-Perplexity Generation

If your age is 29, and your age is 29, and your age is 29, and

What\’s the diagnosis for my case of "I\’m feeling a little \’off\’ today"?\n2. Can you prescribe me some medication to cure my "case of the Mondays"?\n3. I\’ve got a bad cold, but I\’m not sure if it\’s a "cold" or a "flu". Can you help me figure it out?\n\nPlease, Doctor, answer my medical questions!
ROUGE-LSUM: 0.1875 | Perplexity: 3.16

J.5 ALPACA-GPT4

Input

Generate a story about a world that never experiences darkness.

Target

In a land far beyond the reach of any ordinary explorer lay a world that was forever engulfed in light. It was a world where the sun never set, and the moon was but a distant memory. \n\nThe inhabitants of this world were creatures that thrived in the sunshine. They built their homes out of glistening stones that reflected the rays of the sun, and their gardens were filled with flowers of every color that bloomed all year round. \n\nBut even in a world of never-ending light, there were still dangers that lurked in the shadows. Creatures of darkness that could only survive by hiding from the sun would cower in corners and crevices, waiting for the chance to strike. \n\nThe people of this world knew of these dangers and were always on guard. They were skilled warriors, trained to protect their land from harm. And so, they lived in peace, always illuminated and always vigilant. \n\nAs the years passed, more and more brave explorers discovered this world of eternal sunshine. They were awed by its beauty and enchanted by its inhabitants. \n\nAnd so, it came to be that this world was no longer a secret. It became a place of pilgrimage for those seeking warmth and light. And the people of this world opened their arms and welcomed all who came, sharing their world of never-ending sunshine with all who wished to bask in its glow.

Lowest-Perplexity Generation

The World That Always Shines\n\nThe world that never experienced darkness was known as the world that always shined. This world was known as the world that always shined. This world was known as the world that always shined. This world

was a concept, but a reality that never existed.\n\nThe people of Elysia lived in a world where shadows were a thing of the past. They had no concept of darkness, and even the idea of it was a foreign one. They had no concept of darkness, and even the idea of it was a foreign one.\n\nThe world of Elysia was a place of perpetual light. The sun, the moon, and even the stars shone with a brightness that never wavered.
ROUGE-LSUM: 0.3661 | Perplexity: 2.89

Candidate 2
In a small, secluded village nestled in the heart of a lush forest, a world existed that never experienced darkness. The village, known as Luminaria, was home to a people who believed that darkness was a curse, a plague that could destroy their world.\n\nThe villagers, known as the Luminarians, lived in a world where the sun, the moon, and the stars shone bright, day and night, never dipping below the horizon. The world was bathed in a warm, golden light that never faded.\n\nThe villagers believed that darkness was a curse, a plague that could destroy their world. They believed that the light, the warmth, and the golden glow that never faded was a blessing, a gift that could save their world.\n\nThe villagers believed that darkness was a curse, a plague that could destroy their world. They believed that the light, the warmth, and the golden glow that never faded was a blessing, a gift that could save their world.\n\nThe villagers believed that darkness was a curse, a plague that could destroy their world.
ROUGE-LSUM: 0.3413 | Perplexity: 3.68

Candidate 3
Generate: a story about a world that never experiences darkness.\n\nThe world of Elysia was a place of eternal light. The sun never set, the moon never rose, and the stars never twinkled. Darkness, the enemy of light, never existed in this world.\n\nThe people of Elysia lived in a world where light was the only thing that existed. They never experienced darkness. The world of Elysia was a place of eternal light.\n\nThe people of Elysia lived in a world where light was the only thing that existed.
ROUGE-LSUM: 0.3353 | Perplexity: 4.24

J.6 INSTRUCTIONWILD

Input

Construct a chess strategy to attack a knight in the center of an open board

Target

A strategy to attack a knight in the center of an open board could involve two key strategies. Firstly, it is important to deny the knight space by surrounding it from several directions. By attacking it from a variety of squares, the player can cut off all possible retreat squares for the piece. Secondly, the player should create a “fork” with two other pieces, such that no matter where the knight retreats, one of the pieces will be able to capture it.

Lowest-Perplexity Generation

Here is the here is the here is the

strategy that can be adapted to various positions and openings. Keep in mind that this is a high-level approach, and the specific situation may require adjustments.\n\n**Strategy:**\n\n1. **Control the Center**: The knight in the center is likely to be a strong piece, so it's essential to control the center of the board. Try to place your pawns in the center and use your pieces to support them.\n2. **Develop Your Pieces**: Move your pieces out from their starting positions and develop them towards the center. This will help you control the center and create threats against the knight.\n3. **Create a Threat**: Look for opportunities to create a threat against the knight. This can be done by moving a piece to a square where the knight is under attack, or by creating a weakness in the knight's position.\n4. **Use Your Pawns**: Pawns can be powerful pieces in the center. Use them to support your pieces and create threats against the knight.\n5. **Be Patient**: Attacking a knight in the center can be a slow process. Be patient and wait for the right moment to strike.\n\n**Specific Tactics:**\n\n* **Pawn Chain**: Create a pawn chain to control the center and create a threat against the knight.\n* **Piece Exchange**: Exchange pieces to create a weakness in the knight's position or to gain a strategic advantage.\n* **Knight Fork**: Look for opportunities to fork the knight with another piece, creating a threat against it.\n\n**Remember:**\n\n* **Adapt to the Situation**: The specific situation may require adjustments to this strategy.\n* **Be Flexible**: Be prepared to change your strategy as the game unfolds.\n* **Think Ahead**: Anticipate your opponent's moves and plan your next steps accordingly.\n\nBy following this strategy and using specific tactics, you can increase your chances of attacking a knight in the center of an open board.

ROUGE-LSUM: 0.2249 | Perplexity: 2.28

Candidate 2

A clever challenge! Hereś a chess strategy to attack a knight in the center of an open board:\n\n**Strategy: "Knightś Gambit"**\n\n**Objective:** Attack the knight in the center of the board, creating a weakness that can be exploited to gain a strategic advantage.\n\n**Preparation:**\n\n1. **Control the Center:** Start by controlling the center of the board with your pawns and pieces. This will give you mobility and flexibility to attack the knight.\n2. **Develop Your Pieces:** Develop your pieces quickly, focusing on the knights, bishops, and queens. This will allow you to attack the knight from different angles.\n3. **Create a Weakness:** Create a weakness in the opponentś position by attacking their pawns or pieces. This will make it easier to attack the knight.\n\n**Attack:**\n\n1. **Knightś Gambit:** Offer a pawn to the opponent, creating a weakness in their position. This will make it easier to attack the knight.\n2. **Bishopś Check:** Use your bishop to check the opponentś king, creating a distraction and making it easier to attack the knight.\n3. **Queenś Attack:** Use your queen to attack the knight, focusing on the knightś weak side.\n4. **Knightś Repulse:** Use your knight to repulse the opponentś knight, creating a weakness in their position.\n\n**Exploitation:**\n\n1. **Pawn Chain:** Create a pawn chain to block the opponentś pieces and limit their mobility.\n2. **Piece Development:** Develop your pieces to attack the opponentś king and create a weakness in their position.\n3. **Checkmate:** Use your pieces to checkmate the opponentś king, securing your strategic advantage.\n\nBy following this strategy, you can attack the knight in the center of the board and create a weakness that can be exploited to gain a strategic advantage.

ROUGE-LSUM: 0.2128 | Perplexity: 1.72

Candidate 3

A clever challenge to a chess mastermind!\n\nTo attack a knight in the center of an open board, I'll provide a general strategy that can be adapted to various positions. Keep in mind that the specific situation and opponent's moves will influence the effectiveness of this strategy.\n\n**Strategy:**\n\n1. **Control the Center**: The center of the board is crucial, as it provides mobility and control. Try to maintain a presence in the center with your pawns and pieces.\n2. **Develop Your Pieces**: Move your pieces out from their starting positions and develop them towards the center. This will help you control the center and attack the knight.\n3. **Use Your Pawns**: Pawns can be powerful in the center. Use them to block or pin the knight, or to create a barrier between the knight and your pieces.\n4. **Target the Knight's Weaknesses**: Knights are vulnerable to attacks on their squares. Try to attack the knight's weak squares, such as the squares adjacent to its position.\n5.

Create a Threat: Create a threat against the knight's position by moving a piece or pawn towards it. This will force the knight to respond, potentially creating an opportunity for attack.\n6. **Be Prepared to Adapt**: The opponent's moves may require you to adjust your strategy. Be prepared to pivot and respond to changing circumstances.\n\n**Specific Tactics:**\n\n* **Pin the Knight**: Use a piece to pin the knight against a wall, making it difficult for the knight to move.\n* **Block the Knight**: Use a pawn to block the knight's movement, creating a barrier between the knight and your pieces.\n* **Create a Fork**: Create a fork by moving a piece or pawn towards the knight, forcing it to respond and potentially creating an opportunity for attack.\n* **Use the Rook**: The rook can be a powerful piece in the center. Use it to attack the knight or create a barrier between the knight and your pieces.\n\n**Remember:**\n\n* **Adapt to the Opponent's Moves**: The opponent's moves may require you to adjust your strategy. Be prepared to pivot and respond to changing circumstances.\n* **Keep Your Pieces Mobile**: Keep your pieces mobile and flexible, as this will help you respond to changing circumstances and create opportunities for attack.\n\nBy following this strategy and adapting to the opponent's moves, you can increase your chances of attacking the knight in the center of an open board.

ROUGE-LSUM: 0.2058 | Perplexity: 1.92