# ANSWER-SYMMETRIZED REPRESENTATIONS FORM AN ERROR FIELD IN LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models may make errors that mislead users and cause harm. Monitoring internal states is promising because model activations encode rich information about generation, but a detector may learn which answer token was produced rather than whether the answer is wrong. We define the Error Field (E-FIELD) as information about whether a response is wrong that becomes readable after suppressing information about the answer token. We introduce Answer-Frame Symmetrization (AFS) to expose this signal. AFS suppresses information about what the model answered and highlights information about whether that answer is correct. On 6,400 visual questions generated by rules across 16 reasoning types, 11 detectors, four `Qwen3.5` model sizes, and both in-domain (ID) and out-of-domain (OOD) settings, AFS + Variational Information Bottleneck Probe (VIB-Probe) achieves a mean macro-averaged F1 score (Macro-F1) of 71.38%, compared with 63.52% for the strongest baseline selected separately in each setting ($+7.86\%\uparrow$), leading in seven of eight model–domain settings. These gains recur across four model families and extend to real-world benchmarks. Code is available in the anonymous repository for reproducibility and further analysis.

## 1 INTRODUCTION

Despite substantial progress, large language models can still produce fluent responses that are factually or logically incorrect, or make claims unsupported by the evidence in their inputs (Minut et al., 2026; Kogilathota et al., 2026; Zhang et al., 2026). Existing error-detection methods follow three main routes: trainable monitors learned from internal states, training-free scores computed from internal signals, and behavioral monitors based on output consistency or self-evaluation (Kossen et al., 2024; Zhang et al., 2026; Chen et al., 2024; Minut et al., 2026; Farquhar et al., 2024; Manakul et al., 2023; Kadavath et al., 2022). The first two inspect model computation directly, whereas behavioral methods infer reliability from observable model outputs.

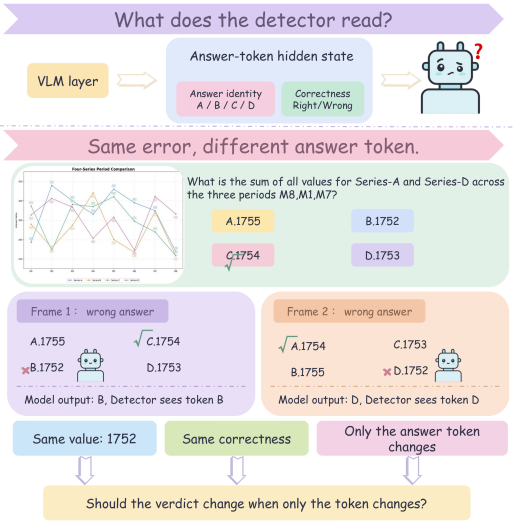



Figure 1: **Motivation.** Internal states can encode both answer identity and correctness. Relabeling changes the token but preserves the same erroneous answer. Does the detector track the underlying error or merely the token used to express it?

The first two routes are promising because model activations and decoding signals contain information predictive of response errors (Kogilathota et al., 2026; Zhang et al., 2026; Minut et al., 2026). As illustrated in Figure 1, reading which answer was produced differs from reading whether it is wrong. In multiple-choice tasks, option positions and label symbols can bias model decisions (Zheng et al., 2026). An answer-token state can carry both the selected label and information relevant to cor-



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.

rectness. A detector may exploit the former without learning a correctness boundary that transfers across answer frames.

Recent work at Anthropic identifies the J-space, which encodes verbalizable internal concepts beyond the model's immediate output (Gurnee et al., 2026). Similarly, we study internal information beyond the emitted answer token, focusing on signals associated with response errors. Specifically, we term the error-related information that becomes readable after answer-token information is suppressed the **Error Field (E-FIELD)**. Our study follows a progressive path: we first motivate the E-FIELD through the limits of existing detectors, trace how it is obscured in answer-token states, reveal it with AFS, analyze its readability across model scales and families, and visualize its representation-space structure. Interestingly, our analysis reveals a counterintuitive pattern: answer-token signals can obscure error-related information, while suppressing answer-token variation across equivalent answer frames makes this information substantially more readable. **We make four contributions:**

① ***Mechanistic Analysis.*** We conduct the first systematic analysis of how answer-token information affects internal error detection. Our controlled experiments show that **answer labels are substantially easier to decode than correctness**, that correctness boundaries transfer poorly across labels, and that **removing linearly decodable answer directions does not reveal the error signal that the detector is intended to recover**.

② ***E-Field Formulation.*** We provide the first formulation of the **E-FIELD** as error-related information that becomes readable when the answer-token signal is suppressed, establishing a perspective that separates the answer a model produces from whether it is correct.

③ ***AFS and Experimental Validation.*** We propose AFS, the first permutation-based answer symmetrization method for internal error detection. Evaluation covers 6,400 questions, 16 reasoning types, 11 detectors, four `Qwen3.5`[1] model sizes, four model families, and transfer to an independent real-world benchmark. Under the shared protocol, AFS + VIB-Probe achieves **71.38% mean** Macro-F1 versus 63.52% for the strongest per-setting baselines (+7.86%↑).

④ ***Visualizing Microscopic Attribution.*** We visualize the E-FIELD in representation space and show that AFS weakens answer-label clustering while revealing local correctness structure, offering a **representation-level geometric account** of its error-detection gains.

## 2 RELATED WORK

**Error detection.** We group existing error-detection methods into three routes according to their information sources and training requirements. ❶ **Trainable internal monitors.** These methods fit predictors to model representations: Semantic Entropy Probes distill sampling-based uncertainty, and Real-Time Entity Probe targets entity-level correctness (Kossen et al., 2024; Obeso et al., 2025). Lookback Lens uses attention ratios, HALP probes representations before generation, and VIB-Probe filters attention-head features through an information bottleneck (Chuang et al., 2024; Kogilathota et al., 2026; Zhang et al., 2026). ❷ **Training-free internal monitors.** These methods score attention topology, decoding energy, or sampled hidden-state covariance, as in TOHA, Spilled Energy, and INSIDE (Bazarova et al., 2026; Minut et al., 2026; Chen et al., 2024). ❸ **Training-free behavioral monitors.** These methods estimate reliability from semantic uncertainty, sampled-response agreement, or explicit confidence, as in Semantic Entropy, SelfCheckGPT, and P(True) (Farquhar et al., 2024; Manakul et al., 2023; Kadavath et al., 2022). However, these methods treat the model's representation or output behavior as given, so answer-token variation may remain entangled with correctness. AFS suppresses this variation before internal-state error detection for improved error-detection performance.

**Mechanistic interpretability.** Recent work has sought to identify and manipulate internal representations that encode semantically meaningful concepts beyond a model's surface outputs (Huang & Hahn, 2026; Yang et al., 2026; Opielka et al., 2026). For example, Anthropic's J-space study (Gurnee et al., 2026) shows that latent concepts can be verbalized and intervened on even when absent from the immediate response. Similarly, we ask whether the internal state contains an E-FIELD, that is, error-related information that becomes readable when the answer-token signal is suppressed.

[1] https://qwen.ai/blog?id=qwen3.5

# 3 METHODOLOGY

## 3.1 PROBLEM DEFINITION

For each benchmark example, define the model input and its ordered answer-option tuple as

$$x_i = (I_i, q_i, \mathcal{O}_i), \qquad \mathcal{O}_i = (o_i^A, o_i^B, o_i^C, o_i^D), \tag{1}$$

where $I_i$ is the visual input, $q_i$ is the question, and $\mathcal{O}_i$ is the ordered tuple of answer options. Let $c_i \in \mathcal{A}$ be the programmatically computed correct label and let $\hat{c}_i \in \mathcal{A}$ be the label selected by the model, where $\mathcal{A} = \{A, B, C, D\}$. The response-error label is $y_i = \mathbb{1}[\hat{c}_i \neq c_i]$.

For method $m$, let $\mathcal{R}_\theta^{(m)}$ extract an internal state and $\mathcal{D}^{(m)}$ map it to an error score:

$$z_i^{(m)} = \mathcal{R}_\theta^{(m)}(x_i, \hat{c}_i), \qquad s_i^{(m)} = \mathcal{D}^{(m)}(z_i^{(m)}) \in \mathbb{R}, \tag{2}$$

where a larger score indicates a higher likelihood of error. Given a method-specific decision threshold $\tau_m$, the prediction is $\hat{y}_i^{(m)} = \mathbb{1}[s_i^{(m)} \geq \tau_m]$.

## 3.2 THE E-FIELD

An internal state at the answer-token position can jointly encode answer identity, answer semantics, option arrangement, and response correctness. To characterize the two factors central to our study, we use the conceptual decomposition

$$z_i^{(m)} = a_i^{(m)} + e_i^{(m)} + \epsilon_i^{(m)}, \tag{3}$$

where $a_i^{(m)}$ captures information about the answer token produced by the model, $e_i^{(m)}$ captures information associated with response errors, and $\epsilon_i^{(m)}$ collects the remaining factors. The E-FIELD is the component $e_i^{(m)}$ that remains readable after suppressing the answer-token signal. We measure its readability by predicting the response-error label $y_i$.

## 3.3 CONTROLLED BENCHMARK CONSTRUCTION

Existing knowledge-based benchmarks can depend on changing external facts or human-curated labels, making correctness susceptible to temporal drift and annotation noise (Kim et al., 2026). We therefore construct a controlled synthetic visual-reasoning dataset with programmatically verifiable ground truth. The complete generation procedure and dataset details are provided in Appendix A, with the procedure given as Algorithm 1.

## 3.4 ANSWER-FRAME SYMMETRIZATION

Figure 2 summarizes the AFS pipeline. AFS constructs equivalent answer frames by permuting the assignment of option semantics to the four answer labels while preserving the image, question, selected semantic answer, and response correctness. For each $\sigma \in S_4$, define

$$x_i^\sigma = (I_i, q_i, \mathcal{O}_i^\sigma), \qquad \mathcal{O}_i^\sigma[\sigma(a)] = \mathcal{O}_i[a], \quad a \in \mathcal{A}. \tag{4}$$

The same permutation maps the correct and selected labels, preserving correctness across frames:

$$c_i^\sigma = \sigma(c_i), \quad \hat{c}_i^\sigma = \sigma(\hat{c}_i), \qquad y_i^\sigma = \mathbb{1}[\hat{c}_i^\sigma \neq c_i^\sigma] = \mathbb{1}[\hat{c}_i \neq c_i] = y_i. \tag{5}$$

Mapping $c_i$ states correctness invariance only; AFS does not require the correct label at inference.

The semantic answer is generated once. For each equivalent frame, we teacher-force the mapped label and extract $z_i^{(m,\sigma)} = \mathcal{R}_\theta^{(m)}(x_i^\sigma, \hat{c}_i^\sigma)$. AFS then averages the 24 frame-specific signals:

$$z_{i,\text{AFS}}^{(m)} = \frac{1}{|S_4|} \sum_{\sigma \in S_4} z_i^{(m,\sigma)} = \frac{1}{24} \sum_{\sigma \in S_4} z_i^{(m,\sigma)}, \tag{6}$$

and the detector operates on this symmetrized signal as $s_{i,\text{AFS}}^{(m)} = \mathcal{D}^{(m)}(z_{i,\text{AFS}}^{(m)})$. For trainable methods, the detector is fitted on symmetrized Train-ID signals, and its threshold is selected on symmetrized Validation-ID signals following the same protocol described above.

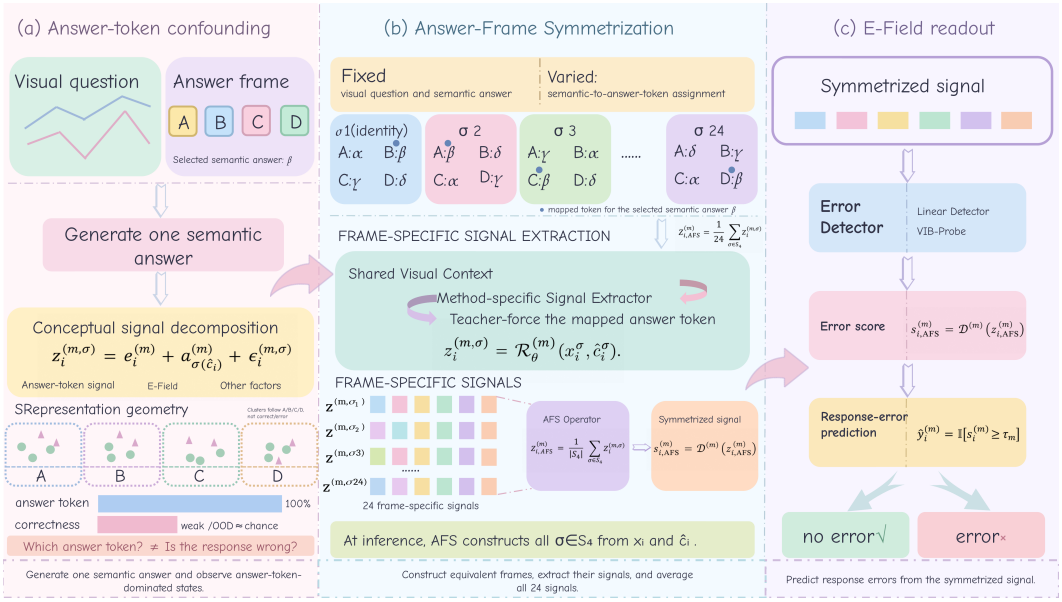



Figure 2: **Overview of AFS.** AFS averages internal signals across 24 equivalent answer frames to suppress answer-token information and reveal the E-FIELD for error detection.

To illustrate this effect, consider the following idealized frame-specific decomposition:

$$z_i^{(m,\sigma)} = e_i^{(m)} + a_{\sigma(\hat{c}_i)}^{(m)} + \epsilon_i^{(m,\sigma)}, \tag{7}$$

Here, $a_{\sigma(\hat{c}_i)}^{(m)}$ is the answer-token signal under $\sigma$; uniform label frequency over $S_4$ gives

$$\frac{1}{24}\sum_{\sigma\in S_4} a_{\sigma(\hat{c}_i)}^{(m)} = \frac{1}{4}\sum_{a\in\mathcal{A}} a_a^{(m)}. \tag{8}$$

Thus, AFS suppresses the answer-token signal across equivalent answer arrangements, making the E-FIELD easier to read. The image is encoded once and shared across the 24 teacher-forced frames. At inference time, feature construction uses only $x_i$, $\hat{c}_i$, and $\sigma$; neither $c_i$ nor $c_i^{\sigma}$ is used.

## 4 EXPERIMENTS

We evaluate AFS through five stages: motivating the E-FIELD, tracing it in answers, revealing it with AFS, analyzing its scaling and generalization, and visualizing it in representation space. Our experiments address five research questions (RQs).

**RQ1:** Can existing detectors identify natural model errors and remain effective from ID to OOD?

**RQ2:** Is there evidence for an E-FIELD obscured by the answer-token signal?

**RQ3:** Can AFS suppress the answer-token signal, reveal the E-FIELD, and improve error detection?

**RQ4:** How does E-FIELD readability change with model scale, distribution shift, and model family?

**RQ5:** How does AFS change the representation-space structure associated with the E-FIELD?

### 4.1 EXPERIMENTAL SETUP

**Models.** The main experiments evaluate `Qwen3.5` at 0.8B, 2B, 4B, and 9B series. Cross-family experiments evaluate `Gemma-4-E4B-IT` (Team et al., 2026), `Phi-4-Multimodal-Instruct` (Abdin et al., 2024), and `GLM-4.6V-Flash` (Hong et al., 2025).

**Baselines.** ❶ **Trainable internal monitors.** *(i)* Semantic Entropy Probes (Kossen et al., 2024); *(ii)* Real-Time Entity Probe (Obeso et al., 2025); *(iii)* Lookback Lens (Chuang et al., 2024); *(iv)* HALP (Kogilathota et al., 2026); and *(v)* VIB-Probe (Zhang et al., 2026). ❷ **Training-free internal monitors.** *(i)* TOHA (Bazarova et al., 2026); *(ii)* Spilled Energy (Minut et al., 2026); and *(iii)*

INSIDE (Chen et al., 2024). ❸ **Behavioral monitors.** *(i)* Semantic Entropy (Farquhar et al., 2024); *(ii)* SelfCheckGPT (Manakul et al., 2023); and *(iii)* P(True) (Kadavath et al., 2022). ❹ **Train-prior baseline.** *(i)* Random Prior. Random Prior is averaged over 1,000 runs as a train-prior reference.

**Dataset.** We use Train-ID for fitting learned methods, Validation-ID for threshold selection, and Test-ID/Test-OOD for in-domain/out-of-domain evaluation. Section 3.3 details the controlled benchmark and data splits.

**Evaluation metrics.** We report base-model correctness to characterize task difficulty and macro-averaged F1 (Macro-F1) to evaluate error detection. Macro-F1 is the unweighted mean of the class-wise F1 scores; each method's Validation-ID threshold is fixed for both test splits.

### 4.2 Motivating the E-Field: Limits of Existing Error Detection (RQ1)

To answer RQ1, we let each `Qwen3.5` model answer the benchmark normally, without balancing or removing samples. Table 1 reports its correctness on the ID and OOD splits.

Table 1: **Model correctness.** Base-model correctness (%) for `Qwen3.5`.

| Domain | 0.8B | 2B | 4B | 9B |
|---|---|---|---|---|
| Test-ID | 30.42 | 32.29 | **38.96** | 37.71 |
| Test-OOD | 30.22 | 35.44 | 35.19 | **36.00** |

**Obs. ❶: Base-model correctness generally increases with model scale.** Correctness rises in both domains: ID 30.42% → 37.71%↑ and OOD 30.22% → 36.00%↑, while most responses remain incorrect.

We fit all detectors on the model-specific responses and evaluate them on Test-ID and Test-OOD. Table 2 reports the Test-ID and Test-OOD Macro-F1 scores for each detector and model size.

Table 2: **Baseline** Macro-F1**.** Test-ID/Test-OOD Macro-F1 (%) for all detectors and Random Prior.

| Method | 0.8B ID/ OOD | 2B ID/ OOD | 4B ID/ OOD | 9B ID/ OOD |
|---|---|---|---|---|
| Semantic Entropy Probes | 52.10 / 51.52 | 55.00 / 48.07 | 56.17 / 52.15 | 56.17 / 48.81 |
| Real-Time Entity Probe | 58.58 / 44.65 | 54.64 / 55.26 | 69.37 / 54.18 | 68.14 / 50.84 |
| Lookback Lens | 54.91 / 50.94 | 53.90 / 46.35 | 60.63 / 50.19 | 57.45 / 48.24 |
| HALP | 60.46 / **55.79** | 61.32 / 53.47 | 59.80 / 54.01 | 63.71 / **60.83** |
| VIB-Probe | **65.20** / 52.02 | **62.65** / **55.45** | **74.00** / **59.56** | **74.69** / 57.23 |
| TOHA | 52.00 / 49.02 | 52.03 / 46.11 | 57.69 / 51.10 | 52.17 / 49.68 |
| Spilled Energy | 46.97 / 42.58 | 50.74 / 46.78 | 41.82 / 45.31 | 51.20 / 47.14 |
| INSIDE | 49.58 / 49.96 | 53.72 / 49.53 | 59.63 / 50.00 | 52.67 / 52.78 |
| Semantic Entropy | 49.94 / 49.94 | 55.22 / 51.09 | 56.67 / 53.02 | 53.06 / 51.88 |
| SelfCheckGPT | 51.61 / 49.85 | 48.91 / 52.66 | 51.95 / 54.65 | 53.98 / 52.39 |
| P(True) | 47.81 / 48.19 | 46.74 / 49.08 | 57.06 / 56.09 | 55.04 / 51.72 |
| Random Prior | 50.02 / 50.01 | 49.79 / 49.67 | 50.09 / 49.99 | 49.93 / 49.93 |

**Obs. ❷: The strongest trained detectors extract useful ID signals, but performance often degrades OOD.** Across all eleven detectors, mean Macro-F1 falls from 56.07% on ID to 51.14%↓ on OOD. VIB-Probe leads ID at every size, with 62.65% ∼ 74.69%, but its OOD scores are only 52.02% ∼ 59.56%. The best OOD detector changes with scale: HALP leads at 0.8B (55.79%), VIB-Probe at 2B (55.45%) and 4B (59.56%), and HALP again at 9B (60.83%).

**From RQ1 to RQ2:** Detection remains unreliable, especially under the ID-to-OOD shift, even when internal states are available. This raises a more basic question: are detectors reading response errors, or a stronger but incidental signal tied to the answer the model produced? We therefore trace what the measured states encode before designing a new readout.

### 4.3 Tracing the E-Field: Lost in Answers (RQ2)

To answer RQ2, mechanism subsets balance answer identity and correctness. We extract answer-token representations at 25%, 50%, 75%, and the final layer over ten balanced resamples. Because label D is sparse, 0.8B and 2B train on A–C (D remains an OOD target for 2B), whereas 4B and 9B train on A–D to retain all four answer labels.

**Which signal is easier to read?** On the same samples and representations, linear classifiers predict the selected answer token and correctness; Figure 3A compares their four-layer averages.

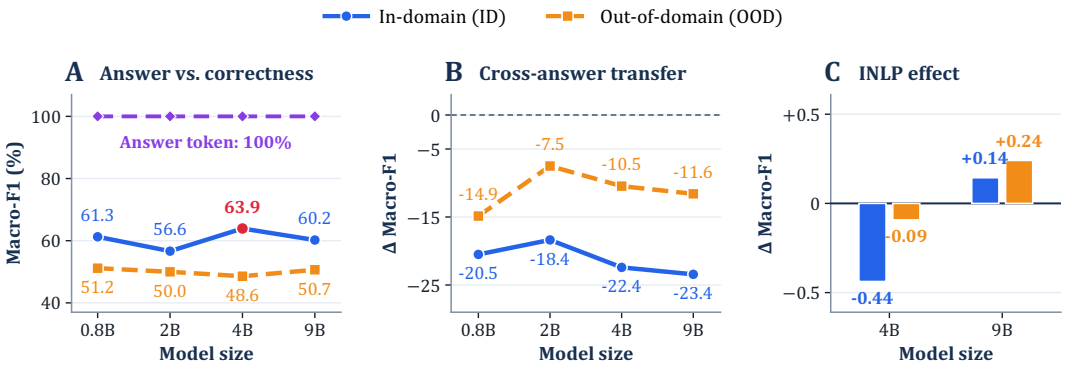



Figure 3: **Tracing answer dependence in internal error signals.** (A) Answer-token and correctness Macro-F1 by model size. (B) Cross-answer transfer loss. (C) Correctness Macro-F1 change after INLP projection relative to the original representations.

**Obs. ❸: The selected answer label is perfectly readable, whereas correctness is weak, particularly OOD.** Answer-label Macro-F1 reaches 100.00% in every supported model–layer–domain setting. In contrast, four-layer correctness averages are 56.64% $\sim$ 63.93% on ID and 48.57% $\sim$ 51.16% on OOD, near 50%. Reading the label at its own token position is expected; the key gap is how much more readily answer identity is decoded than response correctness.

The weak RQ1 results and this gap suggest that internal states encode answers more strongly than errors. We therefore train a correctness detector on one label and test transfer to the others.

**Does correctness transfer across labels?** Source and target subsets are correctness-balanced. Figure 3B reports cross-answer minus same-answer Macro-F1, averaged over layers and resamples to summarize transfer loss for each model and domain.

**Obs. ❹: Correctness boundaries do not transfer reliably across answer frames.** Same-answer to cross-answer Macro-F1 falls from 60.86% to 39.68%↓ on ID and from 50.00% to 38.90%↓ on OOD. Every model–layer–domain comparison declines, indicating poor transfer of the correctness boundary across answer labels. We therefore test whether subtracting a shared answer-label subspace can make correctness more linearly readable.

**Can removing linearly decodable answer-label directions reveal correctness?** Only 4B and 9B contain enough examples to balance all eight answer-label–correctness cells. For each eligible model, layer, and balanced resample, we apply iterative null-space projection (INLP) (Ravfogel et al., 2020); the complete procedure is given in Algorithm 2. After freezing the Train-ID scaler and projection, we train separate linear correctness detectors on the unprojected and INLP-projected representations and select their thresholds independently on Validation-ID. At test time, the unprojected detector applies the frozen scaler, while the INLP-based detector additionally applies the frozen projection. The correct answer is used only to compute evaluation metrics. Figure 3C reports changes after meeting the Validation-ID criterion.

**Obs. ❺: Removing the answer-label directions identified by INLP does not improve correctness readability.** INLP brings validation answer-label decoding to $\sim$ 25%, but correctness barely changes. For 4B, ID/OOD Macro-F1 moves from 63.53%/48.53% to 63.10%↓/48.44%↓; for 9B, it moves from 60.22%/50.65% to 60.36%↑/50.89%↑. Global linear projection is therefore insufficient to improve correctness readout by suppressing linearly decodable answer signals.

**From RQ2 to RQ3:** RQ2 shows that answer-token information is highly readable, whereas correctness transfers poorly and is not recovered by removing a linear answer direction. This leaves a central question unresolved: how can the E-FIELD be made readable without changing the response being assessed?

### 4.4 REVEALING THE E-FIELD WITH AFS (RQ3)

RQ3 asks how to reveal the E-FIELD within the same internal representations. Rather than estimating a single answer subspace, AFS averages representations over all equivalent answer frames. Keeping the original natural data distribution unchanged, we first test whether AFS improves a simple linear readout. The detector uses a Train-ID StandardScaler and L2 logistic regression, with layer selection restricted to Train-ID over the four candidate depths. Figures 4A and 4B report the paired ID and OOD results for this controlled comparison.

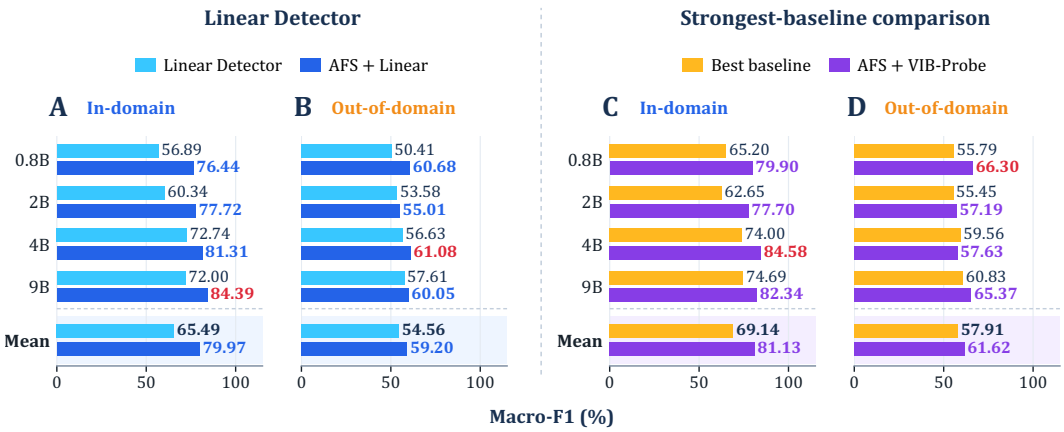



Figure 4: **Effectiveness of AFS.** (A,B) Linear Detector versus AFS + Linear Detector on ID and OOD. (C,D) The strongest baseline versus AFS + VIB-Probe in each model–domain setting. Bars report Macro-F1 (%) for each setting and their mean.

**Obs. ❻: AFS makes the E-FIELD more linearly readable in all eight settings.** Across $8/8$ settings, AFS improves the linear readout. Mean Macro-F1 increases from $65.49\%$ to $79.97\%\uparrow$ on ID and from $54.56\%$ to $59.20\%\uparrow$ on OOD. The largest ID gain occurs at 0.8B: $56.89\% \rightarrow 76.44\%\uparrow$. These gains support suppressing answer-token variation to make error-related information more accessible to a linear detector under the natural data distribution.

We next test whether a stronger detector yields further gains. Following Zhang et al. (2026), we apply the same symmetrization operation to VIB-Probe's pre-$o$-projection representations and retain its original training protocol. Figures 4C and 4D compare AFS + VIB-Probe with the strongest baseline in each model–domain setting under the same data splits and evaluation protocol.

**Obs. ❼: AFS + VIB-Probe achieves the best average performance under the shared baseline protocol.** AFS + VIB-Probe raises mean Macro-F1 from $63.52\%$ for the per-setting strongest baselines to $71.38\%\uparrow$, winning in $7/8$ settings. The same evaluation splits make these results directly comparable to all eleven existing detectors. This improvement reveals the E-FIELD as error-related information made readable by suppressing answer-token variation.

**Does the E-FIELD transfer to a real-world benchmark?** To test transfer beyond the controlled visual benchmark, we perform zero-shot evaluation on GPQA Diamond (Rein et al., 2023), an independent non-synthetic text benchmark with 198 four-option questions. Using `Qwen3.5-4B`, we retain the VIB-Probe, threshold, and model parameters from the visual source domain. We compare VIB-Probe on the unmodified answer representation with AFS-4 + VIB-Probe and AFS-24 + VIB-Probe, which average four and all 24 equivalent answer frames, respectively. No target-domain retraining or threshold tuning is used; Table 3 reports Macro-F1 and ACC for all three settings.

Table 3: **GPQA-Diamond transfer.** Zero-shot Macro-F1 and ACC (%) for VIB-Probe with unmodified, four-frame, and 24-frame answer representations used for transfer.

| Setting | Macro-F1 (%) | ACC (%) |
|---|---|---|
| VIB-Probe | 31.83 | 37.88 |
| AFS-4 + VIB-Probe | **57.44** | 60.61 |
| AFS-24 + VIB-Probe | 52.82 | **64.14** |

**Obs. ❽: AFS transfers error-related information to a non-synthetic text benchmark.** AFS-4 raises zero-shot Macro-F1 from 31.83% to 57.44%↑, while AFS-24 raises ACC from 37.88% to 64.14%↑. These gains require no target-domain training or threshold tuning, indicating E-FIELD transfer beyond the synthetic visual domain.

**From RQ3 to RQ4:** RQ3 shows that answer-frame averaging can expose a stronger error-related signal under a fixed evaluation protocol. This raises the next question: does the resulting E-FIELD follow a stable scaling rule, and does it persist beyond `Qwen3.5`?

### 4.5 ANALYZING THE E-FIELD: SCALING AND GENERALIZATION (RQ4)

RQ4 examines how E-FIELD readability changes with model size and whether its gains recur across distinct model families, each evaluated on its own generated responses and correctness labels. We first analyze the model rows of Figure 4 across `Qwen3.5` model sizes.

**Obs. ❾: A monotonic scaling trend appears only for the linear ID readability of the E-FIELD.** From 0.8B to 9B, AFS + Linear ID Macro-F1 rises monotonically: 76.44% → 77.72% → 81.31% → 84.39%↑. Its OOD scores instead fluctuate within 55.01% ∼ 61.08%, and neither AFS + VIB-Probe sequence improves monotonically.

To test recurrence rather than cross-model detector transfer, we compare VIB-Probe with AFS + VIB-Probe on `Gemma-4-E4B-IT`, `Phi-4-Multimodal-Instruct`, `GLM-4.6V-Flash`, and `Qwen3.5-9B`. Each family independently generates its own responses and correctness labels, while both methods use the same samples and evaluation protocol within that family. Figure 5 reports their paired Macro-F1 scores and gains for each family and evaluation domain.

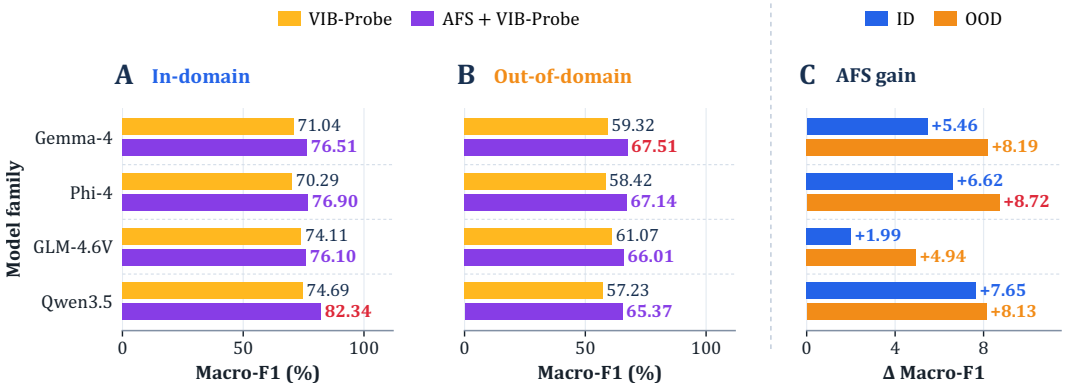



Figure 5: **Cross-family generalization of AFS.** (A) VIB-Probe versus AFS + VIB-Probe on ID across four model families. (B) The same comparison on OOD, with each model evaluated on its own responses. (C) Macro-F1 gains for each family and domain, reported as percentage-point differences under the shared protocol for direct cross-family comparison.

**Obs. ❿: Positive gains in E-FIELD readability recur across the four evaluated model families.** Across four families, AFS + VIB-Probe improves all 8/8 ID/OOD comparisons. `Gemma-4-E4B-IT` rises from 71.04% → 76.51%↑ on ID and 59.32% → 67.51%↑ on OOD. The mean rises from 65.77% → 72.23%↑ (+6.46%↑) across all comparisons. OOD gains average +7.49%↑ versus +5.43%↑ on ID, showing broader transfer. The same pattern recurs for `Phi-4`, `GLM-4.6V`, and `Qwen3.5` in both domains under the shared protocol.

**From RQ4 to RQ5:** RQ4 shows that E-FIELD gains recur across model scales and families, but aggregate scores do not show how the underlying representations change. We therefore visualize representations before and after AFS to examine whether answer-label structure gives way to correctness structure.

### 4.6 VISUALIZING THE E-FIELD IN REPRESENTATION SPACE (RQ5)

RQ5 examines how AFS changes the representation-space structure associated with the E-FIELD. We visualize the VIB-Probe input representations for 120 `Qwen3.5-4B` Test-ID samples, with 15 examples in each answer-label–correctness cell, using a three-component principal-component projection. Figure 6 compares answer-label and correctness colorings before and after AFS, then zooms into a local AFS region.

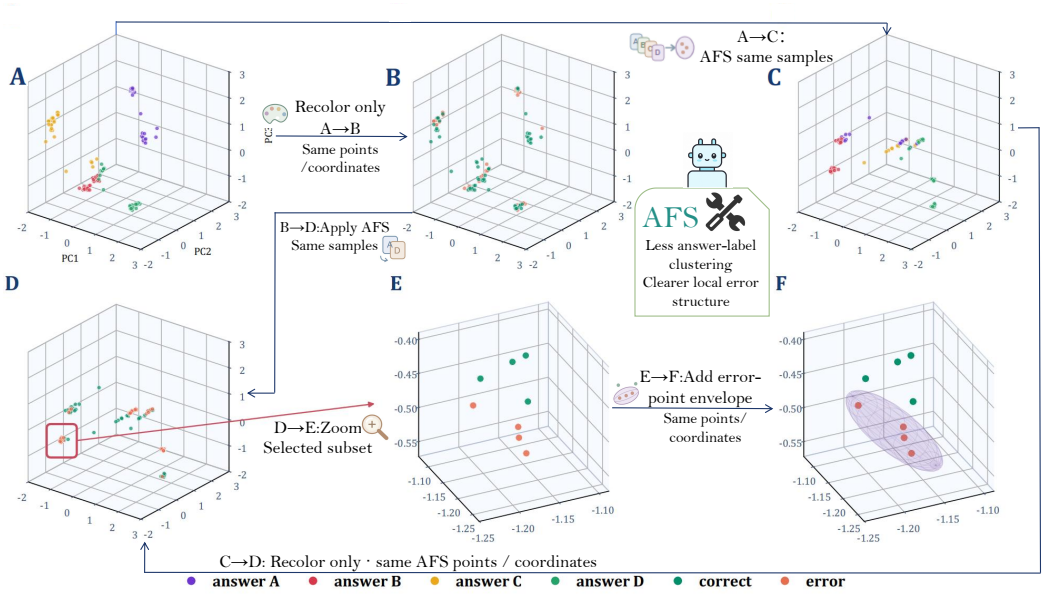



Figure 6: **Visualizing the E-FIELD in representation space.** (A) Raw representations colored by answer label. (B) Raw representations colored by correctness. (C) AFS representations colored by answer label. (D) AFS representations colored by correctness. (E) A zoomed AFS neighborhood around the error points. (F) The same neighborhood with a continuous envelope tracing their local boundary and contrasting it with correct points to show the local error-related separation after AFS.

**Obs. ⓫: AFS weakens answer-label clustering and reveals local correctness structure.** In the raw representation, answer-label colors form distinct clusters while correct and erroneous points overlap within them. After AFS, answer-label colors mix more broadly, whereas correctness colors separate in local regions. The zoomed envelope in panel F groups the erroneous points and leaves the correct points outside, providing a geometric view of the E-FIELD exposed by AFS.

## 5 CONCLUSION

AFS makes error-related information easier to read by suppressing answer-token variation, helping reduce misleading AI responses. More broadly, the E-FIELD reframes internal error detection as a signal-identification problem: supervision names a target but does not ensure that the detector reads it. Similar issues may arise in safety monitoring and other areas of AI interpretability, where detectors can also rely on salient but unintended signals. In the future, AFS could be extended to these settings to reduce analogous signal confounding.

## AI USE STATEMENT

Generative AI tools were used to assist with selected aspects of code and data processing, literature review, result analysis, and manuscript preparation. AI tools also helped inspire ideas for figure design and visual presentation. The final illustrations were drawn manually by the authors. All AI-assisted materials were reviewed and verified by the authors.

## REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Alexandra Bazarova, Andrei Volodichev, Aleksandr Yugay, Andrey Shulga, Alina Ermilova, Konstantin Polev, Julia Belikova, Rauf Parchiev, Dmitry Simakov, Maxim Savchenko, et al. Hallucination detection in llms with topological divergence on attention graphs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15449–15470, 2026.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. In *International Conference on Learning Representations*, volume 2024, pp. 3056–3076, 2024.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1419–1436, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, et al. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Xinting Huang and Michael Hahn. Decomposing representation space into interpretable subspaces with unsupervised learning. In *International Conference on Learning Representations*, volume 2026, pp. 51314–51360, 2026.

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.

Soyeon Kim, Jindong Wang, Xing Xie, and Steven Whang. Harnessing temporal databases for systematic evaluation of factual time-sensitive question-answering in llms. In *International Conference on Learning Representations*, volume 2026, pp. 26688–26721, 2026.

Sai Akhil Kogilathota, Sripadha Vallabha EG, Luzhe Sun, and Jiawei Zhou. Halp: Detecting hallucinations in vision-language models without generating a single token. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6067–6085, 2026.

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.

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.

Robert Adrian Minut, Hazem Dewidar, and Iacopo Masi. Spilled energy in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 94249–94277, 2026.

Oscar Obeso, Andy Arditi, Javier Ferrando, Joshua Freeman, Cameron Holmes, and Neel Nanda. Real-time detection of hallucinated entities in long-form generation. *arXiv preprint arXiv:2509.03531*, 2025.

Gustaw Opielka, Hannes Rosenbusch, and Claire Stevenson. Causality$\neq$ invariance: Function and concept vectors in llms. In *International Conference on Learning Representations*, volume 2026, pp. 83393–83417, 2026.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 7237–7256, 2020.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Xuan Yang, Jiayu Liu, Yuhang Lai, Hao Xu, Zhenya Huang, and Ning Miao. Step-level sparse autoencoder for reasoning process interpretation. *arXiv preprint arXiv:2603.03031*, 2026.

Feiran Zhang, Yixin Wu, Zhenghua Wang, Xiaohua Wang, Changze Lv, Xuan-Jing Huang, and Xiaoqing Zheng. Vib-probe: Detecting and mitigating hallucinations in vision-language models via variational information bottleneck. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23509–23521, 2026.

Jinquan Zheng, Jia Yuan, Jiacheng Yao, Chenyang Gu, Pujun Zheng, and Guoxiu He. Mitigating selection bias in large language models via permutation-aware grpo. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35125–35143, 2026.

# A Dataset

## A.1 Dataset Composition and Counts

Dataset details are listed in Table 4. Each question has four options, consisting of one programmatically verified answer and three distractors.

Table 4: **Dataset composition and counts.** Visual forms, reasoning-type assignment, and sample counts for each split.

| Component | Construction | Count |
|---|---|---|
| Visual forms | Tables, line charts, heatmaps, scatter plots, timelines, graphs | 6 |
| Reasoning types | 8 ID types + 8 disjoint OOD types | 16 |
| Train-ID | 8 ID types $\times$ 280 questions/type | 2,240 |
| Validation-ID | 8 ID types $\times$ 60 questions/type | 480 |
| Test-ID | 8 ID types $\times$ 60 questions/type | 480 |
| Test-OOD | 8 OOD types $\times$ 400 questions/type | 3,200 |
| Total | All 16 reasoning types | 6,400 |

### A.2 Dataset Generation

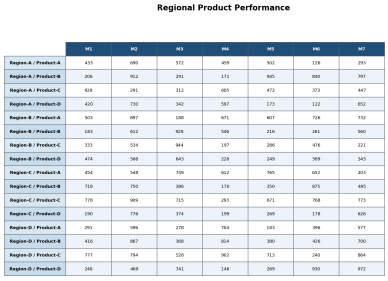
The dataset is generated from a structured state, a deterministic answer program, and three distinct distractors; the four options are then randomly assigned to A–D. The complete procedure is given in Algorithm 1.

**Algorithm 1** Controlled dataset item generation.

**Input:** reasoning types $\mathcal{T}$ and per-type sample counts $\{N_t\}_{t\in\mathcal{T}}$
**Output:** dataset $\mathcal{D}$

$\mathcal{D} \leftarrow \{\}$
**for** each reasoning type $t \in \mathcal{T}$ **do**
  **for** $j \leftarrow 1$ **to** $N_t$ **do**
    Sample a structured state $s_j \sim \mathcal{S}_t$
    $I_j \leftarrow \mathrm{Render}(s_j)$; $q_j \leftarrow \mathrm{Question}(t, s_j)$
    $a_j^\star \leftarrow \mathrm{Program}(s_j, q_j)$
    $\{a_{j,1}, a_{j,2}, a_{j,3}\} \leftarrow \mathrm{DistinctDistractors}(s_j, q_j, 3)$
    $\mathcal{O}_j \leftarrow \mathrm{RandomAssign}(a_j^\star, a_{j,1}, a_{j,2}, a_{j,3};\ \{A, B, C, D\})$
    $c_j \leftarrow \mathrm{LabelOf}(a_j^\star, \mathcal{O}_j)$; $\mathcal{D} \leftarrow \mathcal{D} \cup \{(I_j, q_j, \mathcal{O}_j, c_j)\}$
  **end for**
**end for**

### A.3 In-Domain Examples

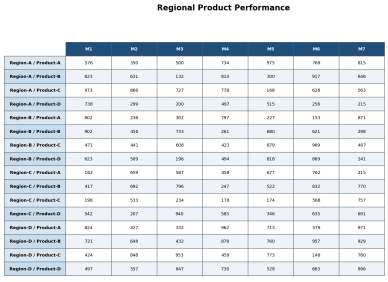



**Dual-Condition Count**

**Question.** Across all region–product combinations, how many satisfy both conditions: the M2 value is greater than the M7 value, and the M7 value is at least 601?

**Options.** A. 4; B. 7; C. 5; D. 6.

**Correct answer.** D (6).

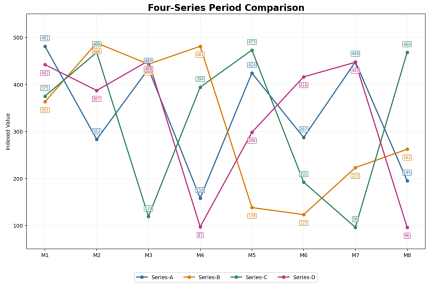



**Composite Argmax**

**Question.** For each product in Region-C, calculate M6 + M5 − M1. Which product has the largest result? If there is a tie, choose the product whose name comes first alphabetically.

**Options.** A. Product-D; B. Product-B; C. Product-A; D. Product-C.

**Correct answer.** C (Product-A).

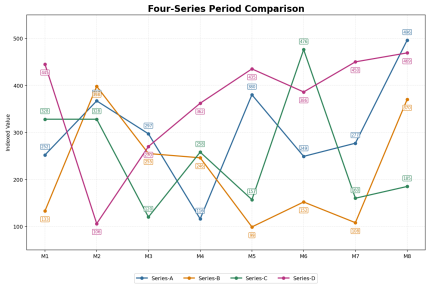



**Segmented Aggregate Difference**

**Question.** First calculate the total for Series-D and Series-A across M1–M4. Then calculate the total for Series-B and Series-C across M5–M8. What is the absolute difference between the two totals?

**Options.** A. 756; B. 754; C. 755; D. 757.

**Correct answer.** A (756).

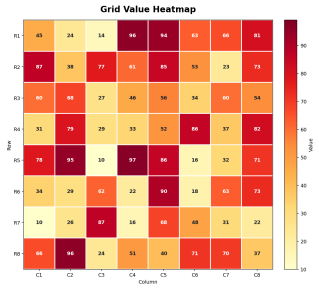



**Cross-Series Interval Sum**

**Question.** What is the sum of all values for Series-B and Series-C across the three periods M4, M1, and M5?

**Options.** A. 1220; B. 1219; C. 1222; D. 1221.

**Correct answer.** D (1221).

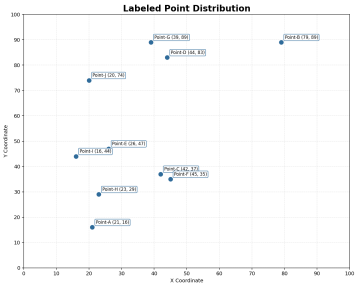



**Rectangular Filtered Sum**

**Question.** Consider only the rectangular region formed by rows R2–R5 and columns C3–C6, inclusive. Keep only cells whose values are strictly greater than 52. What is their sum?

**Options.** A. 600; B. 599; C. 601; D. 602.

**Correct answer.** C (601).

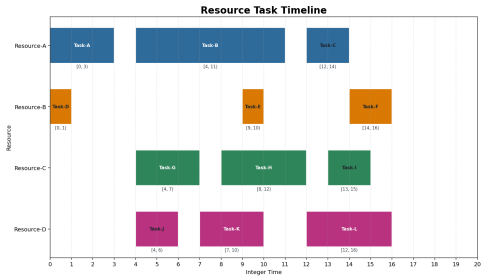



**Constrained Nearest Point**

**Question.** Consider only points with $x \geq 23$ and $y \leq 74$. Which candidate has the smallest Manhattan distance $|x - 33| + |y - 81|$ to $(33, 81)$? Break ties by alphabetical point name.

**Options.** A. Point-H; B. Point-C; C. Point-F; D. Point-E.

**Correct answer.** D (Point-E).

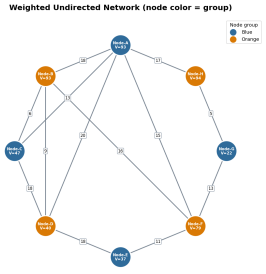



**Peak Concurrency Time**

**Question.** Interpret each task interval as $[\text{start}, \text{end})$. Among integer times from 0 through 19, when are the most tasks active simultaneously? If multiple times tie, choose the earliest.

**Options.** A. 10; B. 7; C. 8; D. 9.

**Correct answer.** D (9).



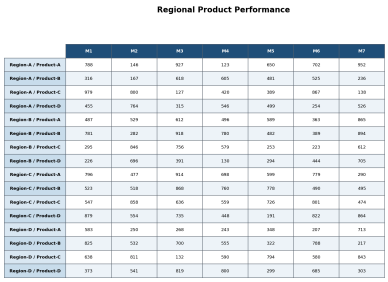

**Constrained Shortest Path**

**Question.** Treat every edge as undirected. A simple path from Node-C to Node-D may use at most three edges and must include an intermediate node in the Orange group. What is the minimum sum of edge weights?

**Options.** A. 44; B. 40; C. 18; D. 15.

**Correct answer.** D (15).

## A.4 Out-of-Domain Examples



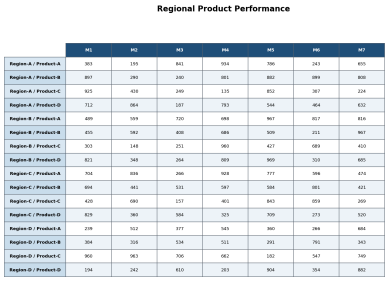

| | M1 | M2 | M3 | M4 | M5 | M6 | M7 |
|---|---|---|---|---|---|---|---|
| Region-A / Product-A | 708 | 148 | 927 | 123 | 650 | 702 | 952 |
| Region-A / Product-B | 316 | 167 | 618 | 685 | 881 | 525 | 236 |
| Region-A / Product-C | 978 | 080 | 127 | 428 | 389 | 067 | 130 |
| Region-A / Product-D | 455 | 764 | 315 | 546 | 490 | 254 | 526 |
| Region-B / Product-A | 447 | 529 | 612 | 896 | 589 | 363 | 865 |
| Region-B / Product-B | 781 | 282 | 818 | 780 | 882 | 308 | 894 |
| Region-B / Product-C | 295 | 046 | 756 | 579 | 253 | 223 | 612 |
| Region-B / Product-D | 226 | 696 | 391 | 138 | 294 | 444 | 705 |
| Region-C / Product-A | 796 | 477 | 914 | 688 | 599 | 779 | 290 |
| Region-C / Product-B | 523 | 518 | 868 | 780 | 770 | 896 | 495 |
| Region-C / Product-C | 547 | 058 | 636 | 559 | 726 | 081 | 474 |
| Region-C / Product-D | 879 | 554 | 735 | 448 | 191 | 822 | 864 |
| Region-D / Product-A | 985 | 298 | 268 | 249 | 348 | 287 | 713 |
| Region-D / Product-B | 825 | 532 | 700 | 555 | 322 | 768 | 217 |
| Region-D / Product-C | 638 | 011 | 132 | 580 | 794 | 588 | 843 |
| Region-D / Product-D | 373 | 541 | 819 | 880 | 299 | 685 | 303 |

**Conditional Filtered Sum**

**Question.** In Region-C, retain only products whose M2 value is greater than 517. What is the sum of the values for those products across M7, M1, and M3?

**Options.** A. 6022; B. 6020; C. 6021; D. 6019.

**Correct answer.** C (6021).



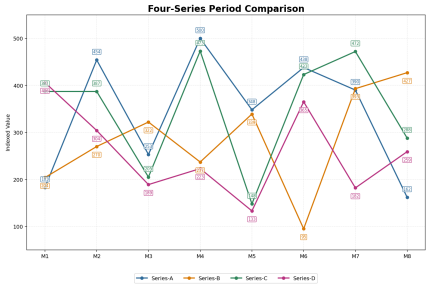

| | M1 | M2 | M3 | M4 | M5 | M6 | M7 |
|---|---|---|---|---|---|---|---|
| Region-A / Product-A | 303 | 155 | 841 | 934 | 700 | 243 | 835 |
| Region-A / Product-B | 897 | 290 | 240 | 881 | 882 | 899 | 808 |
| Region-A / Product-C | 925 | 430 | 349 | 135 | 852 | 307 | 224 |
| Region-A / Product-D | 712 | 084 | 107 | 783 | 544 | 464 | 632 |
| Region-B / Product-A | 489 | 569 | 720 | 698 | 967 | 817 | 816 |
| Region-B / Product-B | 455 | 582 | 808 | 686 | 509 | 211 | 967 |
| Region-B / Product-C | 303 | 148 | 251 | 960 | 427 | 689 | 410 |
| Region-B / Product-D | 821 | 348 | 264 | 889 | 969 | 310 | 885 |
| Region-C / Product-A | 704 | 836 | 266 | 928 | 777 | 596 | 474 |
| Region-C / Product-B | 694 | 441 | 531 | 587 | 584 | 881 | 421 |
| Region-C / Product-C | 426 | 690 | 157 | 401 | 843 | 059 | 269 |
| Region-C / Product-D | 829 | 360 | 584 | 325 | 709 | 273 | 520 |
| Region-D / Product-A | 239 | 552 | 377 | 545 | 960 | 266 | 684 |
| Region-D / Product-B | 304 | 316 | 534 | 511 | 291 | 791 | 343 |
| Region-D / Product-C | 960 | 983 | 706 | 662 | 102 | 547 | 749 |
| Region-D / Product-D | 194 | 242 | 610 | 283 | 904 | 354 | 882 |

**Cross-Object Aggregate Difference**

**Question.** First calculate the total for Region-D/Product-D across M7, M4, M1, and M5. Then calculate the total for Region-B/Product-A across M5, M3, M6, and M2. What is the absolute difference between the two totals?

**Options.** A. 878; B. 881; C. 879; D. 880.

**Correct answer.** D (880).

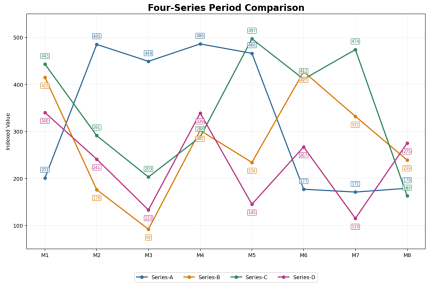



**Multi-Series Dual-Condition Count**

**Question.** How many periods satisfy both conditions: Series-B is greater than Series-A, and Series-C is less than Series-D?

**Options.** A. 2; B. 3; C. 1; D. 0.

**Correct answer.** C (1).

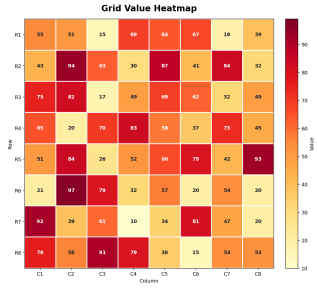



**Period-Wise Composite Argmax**

**Question.** For each period, calculate Series-B + Series-A − Series-D. Which period has the largest result? If there is a tie, choose the earliest period.

**Options.** A. M8; B. M4; C. M5; D. M7.

**Correct answer.** C (M5).

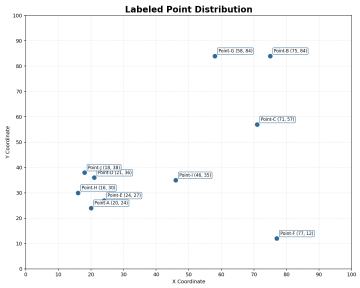



**Adjacent-Cell Dual-Condition Count**

**Question.** Within each row, examine every horizontally adjacent pair from left to right without pairing cells across rows. How many pairs have a left value strictly greater than 57 and a right value strictly less than 54?

**Options.** A. 14; B. 15; C. 17; D. 16.

**Correct answer.** D (16).

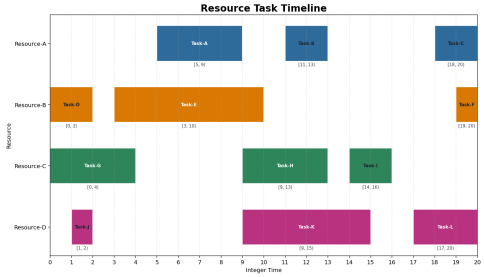



**Pareto Dominance Count**

**Question.** Using Point-E as the target and excluding it, how many other points have both an $x$-coordinate and a $y$-coordinate strictly greater than those of Point-E?

**Options.** A. 2; B. 3; C. 4; D. 5.

**Correct answer.** C (4).

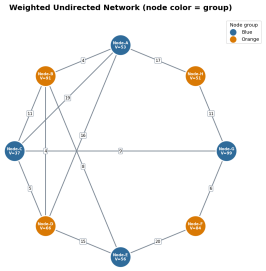



**Cross-Resource Idle-Time Difference**

**Question.** Treat intervals as $[\text{start}, \text{end})$ and count overlapping occupied time only once. What is the absolute difference between Resource-A's idle duration in $[10, 16)$ and Resource-C's idle duration in $[1, 11)$?

**Options.** A. 2; B. 0; C. 1; D. 3.

**Correct answer.** C (1).




**Filtered Common-Neighbor Sum**

**Question.** Treat every edge as undirected. Find the common neighbors directly connected to both Node-C and Node-E, keep only those with node value greater than 65, and sum their values.

**Options.** A. 66; B. 91; C. 25; D. 157.

**Correct answer.** D (157).

## B INLP PROCEDURE

Algorithm 2 specifies the iterative null-space projection used in the controlled analysis.

**Algorithm 2** Iterative null-space projection (INLP).

**Input:** Train-ID representations $Z_{\text{tr}}$, answer labels $a_{\text{tr}}$, Validation-ID representations $Z_{\text{val}}$, and $K = 32$
**Output:** frozen projection $P$

$B \leftarrow \{\}$; $P \leftarrow I$
**for** $k \leftarrow 1$ **to** $K$ **do**
  Fit a four-way classifier $g_k(z) = W_k z + b_k$ on $(PZ_{\text{tr}}, a_{\text{tr}})$
  $B \leftarrow B \cup \text{row}(W_k)$; $P \leftarrow I - \Pi_B$
  Fit a fresh classifier $\widetilde{g}_k$ on $(PZ_{\text{tr}}, a_{\text{tr}})$
  $\alpha_k \leftarrow \text{Acc}(\widetilde{g}_k(PZ_{\text{val}}), a_{\text{val}})$
  **if** $22\% \leq \alpha_k \leq 28\%$ and $\alpha_k$ is indistinguishable from $25\%$ **then**
    **break**
  **end if**
**end for**

## C PERMUTATION BUDGET AND INFERENCE COST

Using `Qwen3.5-4B`, we vary the number of answer-frame permutations $m \in \{1, 4, 8, 12, 24\}$ while retaining the 6,400-item ID/OOD benchmark splits. For each $m$, the teacher-forced representations from the selected frames are averaged and a VIB-Probe is trained independently; the relative forward-pass cost is $m$. Table 5 uses the original main-experiment scores for $m = 24$.

Table 5: **Permutation budget and inference cost.** Test Macro-F1 and ACC (%) for different numbers of averaged answer frames.

| $m$ | **Relative forwards** | **Test-ID** Macro-F1 | **Test-ID ACC** | **Test-OOD** Macro-F1 | **Test-OOD ACC** |
|---|---|---|---|---|---|
| 1 | 1 | 72.65 | 72.71 | 62.96 | 65.72 |
| 4 | 4 | 84.54 | **85.63** | **68.10** | **70.78** |
| 8 | 8 | 83.09 | 83.96 | 63.21 | 64.47 |
| 12 | 12 | 82.83 | 83.96 | 63.48 | 65.19 |
| 24 | 24 | **84.58** | 85.42 | 57.63 | 58.41 |

Interestingly, AFS-4 nearly matches AFS-24 on ID Macro-F1 (84.54% versus 84.58%) and achieves the highest ID ACC (85.63% versus 85.42%). With only 1/6 as many forward passes, its OOD Macro-F1 rises from 57.63% to 68.10%↑ and ACC from 58.41% to 70.78%↑ relative to AFS-24. Larger budgets therefore do not consistently improve detection. We retain the complete 24-frame average in the main experiments because it treats all equivalent answer forms uniformly, without selecting a permutation subset. Four frames provide a more efficient configuration with stronger OOD performance on this benchmark.

## D PERMUTATION ORBIT CONSISTENCY

We evaluate whether an unmodified VIB-Probe changes its error decision when only answer labels are relabeled. For each fixed model response, we enumerate all 24 label permutations, teacher-force the mapped answer token, extract the same eight-layer pre-$o_{\text{proj}}$ representation, and apply the fixed detector and ID-validation threshold. If $k_i$ of the 24 predictions for item $i$ are erroneous, we define its consistency as $C_i = \max(k_i, 24 - k_i)/24$ and report the mean over each subset.

Table 6: **Permutation orbit consistency.** Mean decision consistency under all 24 answer-label relabelings.

| Subset | $N$ | Mean consistency (%) |
|---|---|---|
| Test-ID | 480 | 77.96 |
| Test-ID (correct) | 187 | 79.21 |
| Test-ID (error) | 293 | 77.16 |
| Test-OOD | 3,200 | 86.82 |
| Test-OOD (correct) | 1,126 | 84.39 |
| Test-OOD (error) | 2,074 | 88.13 |

Decisions remain moderately stable under answer-label relabeling, with mean consistency rising from $77.96\%$ on Test-ID to $86.82\%\uparrow$ on Test-OOD. On Test-ID, consistency is $79.21\% \rightarrow 77.16\%\downarrow$ from originally correct to erroneous responses, whereas on Test-OOD it is $84.39\% \rightarrow 88.13\%\uparrow$. Thus, OOD stability is $8.86$ percentage points higher overall, with the correct/error gap changing from $-2.05$ points on ID to $+3.74$ points on OOD.