EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

CrossVerify: low-payload external validation signals for heterogeneous black-box LLM cascades

Li Luo[1,2], Keng Yap Ng[1,3,*] and Wei Chong Choo[4]

[1]Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
[2]Department of Artificial Intelligence and Data Science, Guangzhou Xinhua University, 510520 Guangzhou, Guangdong, China
[3]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
[4]School of Business and Economics, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Abstract

INTRODUCTION: Heterogeneous black-box LLM cascades need routing signals that are reliable, calibratable, and computable without hidden-state access.
OBJECTIVES: This study evaluates whether external verifier-support signals can complement proposer self-confidence in low-payload cascade routing.
METHODS: CrossVerify is designed and tested on six datasets, three proposer-verifier pairs, and five random seeds under both signal-analysis and fixed-fallback escalation settings.
RESULTS: Proposer confidence is the strongest average single-feature baseline, while verifier support is the strongest external signal; their combination yields the best overall routing performance, with clear task dependence and a 28-byte black-box-compatible interface.
CONCLUSION: CrossVerify offers a deployment-compatible signal design whose main value is complementing self-confidence rather than replacing it.

Keywords: LLM cascade, cross-verification, black-box routing, selective prediction, confidence calibration

Received on 16 January 2026, accepted on 19 April 2026, published on 16 September 2026

doi: 10.4108/eetsis.14497

*Corresponding author. Email: kengyap@upm.edu.my

1. Introduction

LLM systems have become increasingly embedded in larger decision systems rather than deployed as isolated predictors. In many realistic settings, especially those involving black-box APIs or heterogeneous model families, the central systems problem is no longer only how to improve standalone accuracy, but how to coordinate multiple models under resource constraints [1-3] . Existing LLM multi-agent and cascade systems demonstrate that specialized roles such as planning, reflection, debate, or coordination can improve task performance [3-5]. However, they leave a basic decision question insufficiently answered: when should a cheap first-stage model be trusted, and when should the system spend additional budget on stronger verification?

The default answer is to rely on self-confidence. This is attractive because confidence-like signals are easy to compute and widely available. Yet self-confidence alone is often insufficient in heterogeneous black-box collaboration. First, confidence scores from different model families are rarely calibrated on the same numerical scale [6-8]. Second, internal confidence does not directly indicate whether a second model would support or challenge the current decision. Third, when only output-space statistics are

available, standard alternatives such as hidden-state routing or representation-level uncertainty analysis become inaccessible or expensive [6,8,9]. These issues are especially important for deployable expert systems, where routing decisions must be reliable, auditable, and lightweight [9-11]. Recent work has extended this discussion to black-box uncertainty quantification, multicalibration, confidence elicitation, and trustworthiness-oriented confidence supervision for LLM reliability [12-14].

This paper investigates a narrower and more practical problem: how to design compact, black-box compatible external verification signals for heterogeneous LLM cascades. The significance of the second model lies not merely in its own confidence, but rather stems from its broader contextual contributions. What matters is whether the second model supports the answer proposed by the first model, and whether this support remains informative after the confidence of the proposer has already been observed. Specifically, the probability that a verifier assigns an answer —$V_{prob,B\to A}$ to an applicant is a different quantity from the verifier's confidence in his own answer $P_{max,B}$, and making this distinction explicit is the main signal design step of this paper. This motivates CrossVerify, an output space method for studying cross-model validation signals under black-box constraints. This paper does not claim that cross-validation universally dominates confidence, nor does it claim that the verifier -support scalars emerge out of nowhere. Instead, it makes a more reasonable and useful argument:

Cross-validation provides a family of externally validated signals that depend on the task, black-box compatibility, and low payloads, with values that are complementary to confidence, most evident in calibration signal combinations.

This contribution is important for two reasons. Empirically, it provides a concrete answer to the signal design problem in heterogeneous LLM collaborations and clarifies whether the information supported by the verifier adds value beyond the confidence of the proposer. Methodologically, it relates LLM cascade design to calibration, selective prediction, rejection option learning, and uncertainty-aware deployment [6-8]. Thus, systems such as cross-validation should be judged not by whether a certain scalar dominates every benchmark, but by whether they provide compact, transferable, and decision-relevant signaling interfaces under deployment constraints.

The contributions of this paper are as follows.

1. Signal design. Verifier support for the proposer's answer is studied and formalized as a black-box routing signal that is distinct from the verifier's own self-confidence. Empirical results show that this support signal is the strongest external validation variable, that it joins proposer confidence in the best-performing feature combination (A+B+agree+V), and that grouped Shapley attribution confirms its near-equal explanatory contribution to the self-confidence block.

2. Empirical framework. A structured multi-axis evaluation is conducted, covering six datasets across two task families (claim verification and multiple-choice reasoning), three model pairs, and five random seeds. The study covers single-feature evaluation, progressive ablation, and grouped Shapley attribution. It also tests cross-pair and cross-task transfer with leave-one-dataset-out analysis, multiseed robustness across five seeds, and task-topology dependence.

3. Deployment analysis. Canonical ECDF alignment supports cross-pair and cross-task transfer without annotation, quantile-space threshold routing closely matches the offline budget-constrained optimum in a fixed stronger-fallback setting, and the output-space signal interface requires only 28 bytes per routing decision, independent of model hidden-state size, while remaining compatible with black-box APIs. Any end-to-end system-cost advantage over one-pass baselines is therefore conditional rather than universal.

2. Related work

2.1. LLM-based multi-agent and cascade systems

Recent work has explored LLM-based multi-agent collaboration, expert systems, and cascade architectures in domains such as autonomous driving, negotiation, knowledge-centric question answering, urban management, and human-robot interaction [1,3,5]. These studies show that LLMs can support planning, reflection, communication, and coordination. However, most of them emphasize system architecture and task performance rather than the lower-level problem of routing signal design under heterogeneous black-box collaboration. This paper does not aim to propose another complex agent protocol, but rather to isolate and directly study the routing problem.

2.2. LLM cascade and routing

A rapidly growing body of work directly addresses the problem of routing queries across multiple LLMs to balance cost and accuracy. MixLLM [15] proposes dynamic routing across heterogeneous LLM ensembles using query-level features. Large Language Model Routing with Benchmark Datasets [16] emphasizes benchmark-driven evaluation for router development, while SATER [17] and confidence-driven model selection [18] study when adaptive routing and confidence can support cost-efficient delegation. Cost-aware contrastive routing [19] and related efficiency-oriented routing methods further develop this line under explicit compute constraints. More recent work also explores test-time confidence and uncertainty estimation, as well as uncertainty-aware decision policies that can support routing and fallback choices [20,21].

Recent work close to CrossVerify has increasingly focused on confidence estimation, calibration, and self-verification for reliable selective prediction. Factual Confidence of LLMs [22] provides a systematic comparison of current factual-confidence estimators, while CCPS [23] improves confidence estimation by probing perturbed representation stability. At the same time, self-verification and verifier-strength studies [24,25] show both the promise and the limits of verifier-like checks, and they highlight the importance of strong verifiers

for improving reasoning reliability. Confidence-elicitation and trustworthiness studies further show that selective generation depends not only on raw confidence, but also on whether models can express uncertainty in a calibrated and decision-useful form [20,26]. CrossVerify is differentiated by its explicit decomposition of verifier support for the proposer answer versus verifier self-confidence, its black-box compatibility, and its transfer-oriented evaluation design.

2.3. Confidence, calibration, and selective prediction

Our methodological foundation is more closely tied to calibration, uncertainty-aware deployment, selective prediction, and reject-option learning [6-8]. This literature confirms that confidence-sensitive decisions require calibrated probabilities and that abandoning or upgrading may be preferable to forcing predictions when the cost of error is high [6,7,27]. Uncertainty research further highlights the distinction between reducible and irreducible uncertainty and the importance of deploy-aware risk control [8,28]. Related integration and cascade systems also use consensus or consensus style heuristics in practice [18,27,29], but usually as AD hoc scoring rules, voting signals, or expert selection devices, rather than as explicitly separated black-box support signals. We extend this literature from the single-model setting to heterogeneous LLM cooperation. Instead of asking how to calibrate one predictor, we ask how to combine self-confidence with an external validation signal produced by another model. Recent work has further broadened this line toward black-box UQ under data uncertainty, multi-calibration, confidence elicitation, and benchmark-driven evaluation [20,30].

2.4. Black-box deployment and trusted expert systems

The third line deals with black-box deployment, explainability, and trust assurance for AI systems [9-11]. The LMaaS setting complicates reproducibility, calibration, and trust evaluation because internal representations are unavailable [9]. At the same time, the decision-support literature increasingly emphasizes interpretability, uncertainty awareness, and auditability as operational requirements rather than optional add-ons [11,31]. CrossVerify is broadly consistent with this view: it uses only output-space statistics, avoids hidden-state dependence, and produces interpretable routing signals in the form of support, agreement, and difference.

2.5. Positioning

Thus, the gap in the literature is clear. LLM routing and cascading studies [17,18] substantially improve the cost-accuracy trade-off, but they often rely on benchmark supervision or routing objectives that do not directly isolate verifier support for proposer answers. Meanwhile, recent work on confidence estimation and self-verification has clarified how calibration, factual confidence, verifier strength, and self-verification affect trustworthiness, while uncertainty elicitation studies [20,26] have further shown that decision quality depends on whether confidence estimates are calibrated and actionable. The contribution of this paper is not to introduce a protocol-like scalar from scratch, but rather to isolate verifier support for proposer answers as a deployable family of signals and evaluate its value added after calibration and alignment [32]. Compared to neighboring confidence- and verification-oriented approaches, the difference here is three-fold: the object is heterogeneous black-box routing rather than independent confidence scoring, the interface is constrained-decoding logprob rather than hidden states or learned embeddings, and the goal is budget-aware trust and upgrade decisions rather than static confidence estimation.

3. Data, signal construction, and experimental design

3.1. Problem formulation

The study considers a heterogeneous two-stage cooperative environment. For an input query X, the lightweight proposer A first produces an answer and a predictive distribution. The verifier (B) generates its own distribution over the same answer space from which the cross-validated features are extracted. Two associative decision problems are studied. First, in a signal evaluation setting, we ask whether these features help estimate whether the proposer's answer is trustworthy; Thus, $Z_A = 1[f_A(X) = Y]$ is the proposer correctness measure for the ground truth label Y from Sections 4.1-4.5. Second, in the budget deployment setting, we ask whether the sample should be accepted at L1 or upgraded to a fixed stronger alternative model L2 under an explicit cost limit. The routing rule

$$g(X) \in \{L1, L2\} \quad (1)$$

Whether or not is evaluated by the final cascade accuracy on Y, but the escalation action is instantiated with a non-oracle stronger model rather than a ground-truth substitution. The goal is therefore not for the router alone to estimate Y, but the escalation action is instantiated with a non-oracle stronger model rather than a ground-truth substitution. The goal is therefore not for the router itself to predict Y directly, but to learn decision-relevant risk signals for trusting A and, in the budgeted setting, for upgrading to a stronger fallback.

3.2. CrossVerify signal construction

CrossVerify constructs two signal families from the proposer and verifier output distributions (Fig. 1).

The self-confidence family is

$$S_{self} = [P_{max,A}, \text{Entropy}_A, P_{max,B}, \text{Entropy}_B] \quad (2)$$

The cross-verification family is

$$S_{cv} = [\text{agree}, V_{prob,B\to A}, \text{Disagreement}] \quad (3)$$

where $\text{agree} = 1[\hat{y}_A = \hat{y}_B]$ is a zero-cost binary indicator derived from the two proposer outputs, $V_{prob,B\to A}$ is the verifier probability assigned to the proposer's answer, and

$$\text{Disagreement} = |P_{max,A} - V_{prob,B\to A}| \quad (4)$$

The central distinction is between $P_{max,B}$ and $V_{prob,B\to A}$. The former reflects the verifier's confidence in itself; the latter reflects the verifier's support for the proposer. This distinction is the main signal-design idea of the paper.

This distinction has a formal interpretation. The cross-verification family S_{cv} should carry routing value only if it is conditionally non-redundant once self-confidence is observed, i.e., $I(Z_A; S_{cv} \mid S_{self}, X) > 0$. Because $V_{prob,B\to A}$ is constructed from the verifier's distribution evaluated at the proposer's answer—rather than at the verifier's own argmax—it captures a different projection of the verifier's uncertainty and is therefore not algebraically subsumed by $P_{max,A}$ or $P_{max,B}$. Section 5.1 provides an empirical proxy test for this conditional non-redundancy claim.

The final routing feature vector is

$$\begin{bmatrix} P_{max,A}, \text{Entropy}_A, P_{max,B}, \text{Entropy}_B, \\ \text{agree}, V_{prob,B\to A}, \text{Disagreement} \end{bmatrix} \quad (5)$$

which is obtained from only two forward passes and requires no hidden-state access.

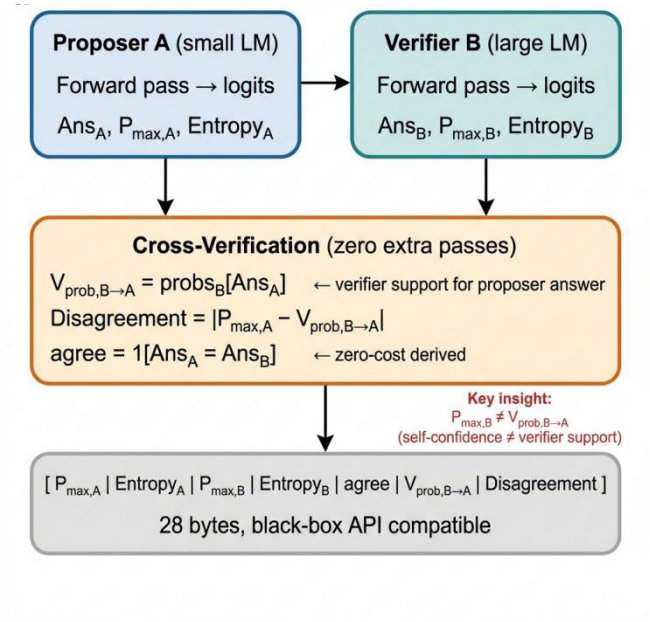

Figure 1. CrossVerify framework and signal construction

The CrossVerify pipeline extracts routing signals from two forward passes without hidden-state access. The proposer A (small LM, blue) generates a candidate answer together with self-confidence features $P_{max,A}$ and Entropy_A. The verifier B (large LM, green) independently produces its own answer and confidence features $P_{max,B}$ and Entropy_B. The cross-verification module (orange) then extracts three external signals — $\text{agree} = 1[\hat{y}_A = \hat{y}_B]$, $V_{prob,B\to A}$, and Disagreement — from B's probability distribution evaluated at A's answer, requiring zero additional forward passes. The final 7-element feature vector (lower panel, grey) occupies only 28 bytes and is fully compatible with black-box commercial APIs. The key insight (highlighted in red) is that $P_{max,B}$ (verifier self-confidence) and $V_{prob,B\to A}$ (verifier support for the proposer) are logically distinct quantities that capture complementary aspects of cross-model uncertainty.

3.3. Datasets

CrossVerify is evaluated on six datasets: FEVER, FEVEROUS, HoVer, GPQA, C-Eval, and MMLU-Pro. These datasets cover both factual verification and multiple-choice reasoning, and are intentionally heterogeneous in answer-space structure and ambiguity.

For the later task-dependence analysis, FEVER, FEVEROUS, and HoVer are grouped as claim-heavy tasks, while GPQA, C-Eval, and MMLU-Pro are grouped as MC-heavy tasks. The final sample counts are listed in Table 1.

Table 1. Dataset and Samples

Dataset	Samples
FEVER	1000
FEVEROUS	1000
HoVer	1000
GPQA	546
C-Eval	1000
MMLU-Pro	1000

3.4. Model configuration

The main proposer-verifier pair is Qwen3.5-2B and Phi-4-mini-instruct. A separate stronger model, Qwen3.5-27B-MLX-4bit, is used only to instantiate budgeted escalation outcomes in the fixed-fallback analyses; it is treated as a stronger deployment option, not as an oracle. All inference is performed through MLX-based model wrappers on Apple Silicon. The proposer and verifier are queried with the same task prompt and a fixed assistant prefix "" to reduce token-level frequency bias.

As a sanity check for this fixed-fallback interpretation, Table 2 reports ground-truth accuracy for the main proposer A, the verifier B, and the stronger L2 model on the same sampled items. The key property required for the deployment analyses is that L2 be empirically stronger than the proposer, not that it be perfect. This condition is satisfied on all six datasets: L2 exceeds A by +0.063 on FEVER, +0.195 on FEVEROUS, +0.095 on HoVer, +0.181 on GPQA, +0.313 on C-Eval, and +0.314 on MMLU-Pro. The threshold-routing results in Section 5.3 and the deployment-cost discussion in Section 4.6 should therefore be read as evidence for routing

with a fixed stronger fallback, not as evidence for oracle escalation.

Table 2. Ground-truth accuracy of the main proposer-verifier pair and the stronger fixed fallback

Dataset	accuracy	accuracy	L2 accuracy	L2 minus
FEVER	0.715	0.718	0.778	+0.063
FEVEROUS	0.430	0.549	0.625	+0.195
HoVer	0.497	0.489	0.592	+0.095
GPQA	0.326	0.297	0.507	+0.181
C-Eval	0.508	0.354	0.821	+0.313
MMLU-Pro	0.298	0.316	0.612	+0.314

3.5. Experimental pipeline

The study follows a two-stage design. In the first stage, the proposer and verifier are profiled to extract CrossVerify features together with proposer/verifier correctness indicators against ground truth, producing the L1 tables. The stronger L2 model is run under an identical sampling setup to establish the fixed-fallback outcomes.

These outputs then feed into a signal-analysis pipeline. Sections 4.1-4.5 analyze the extent to which the proposed signals explain proposer correctness and evaluate how well they transfer across different tasks and model pairs. For budget-sensitive routing, the recorded L2 output is treated as a concrete fixed standby upgrade setting rather than an oracle controller. Section 4 reports the results in an order designed to build understanding step by step: first the task-dependent signal landscape (Section 4.1), followed by signal-combination value and Shapley attribution (Section 4.2), single-feature baselines (Section 4.3), cross-pair and cross-task generalization (Section 4.4), multi-seed robustness (Section 4.5), and system overhead (Section 4.6). Section 5 then provides a theoretically informed empirical test consistent with these findings.

3.6. Calibration, alignment, and evaluation metrics

An obvious challenge in this context is that different model families produce confidence scores at completely mismatched scales. A strong alignment strategy is therefore required. A non-parametric method is adopted to avoid the normality assumption. Each feature is mapped to a uniform quantile scale via an empirical CDF. Merging across datasets is then performed using a mixture of centroids weighted by sample size, thereby establishing a common normative scale. Although conceptually simple, this calibration effectively places all features in a unified space.

All subsequent evaluations are performed in this specification framework. For single-feature evaluation, a simple one-dimensional calibrator is trained to map each signal to a posterior probability. We report AUC and AP to measure ranking quality, and BS, BSS and ECE to measure calibration quality. BSS is used as the main indicator. By comparing performance to previous training, it essentially performs a fairer cross-task evaluation. For budget routing, final path accuracy, average path cost, L2 routing frequency, and budget gap with respect to the offline threshold reference are tracked.

4. Empirical results

Evaluations include task-specific signal behavior (4.1), signal combination with single-feature baselines (4.2,4.3), cross-domain generalization (4.4), seed robustness (4.5), and system overhead (4.6).

4.1. The task-dependent signal landscape

Before evaluating the signal combinations, the task panorama is first described. This analysis establishes a comparison of two series, Table 3:Claim-heavy and MC-heavy tasks. On the more demanding task, $V_{prob,B\to A}$ is the strongest cross-validated signal, with an average blind source separation (BSS) of around 0.0290, while cue-centric features remain weak. In the MC-heavy task, the proposer's confidence is dominant: $P_{max,A}$ reaches about 0.1048 and $Entropy_A$ reaches about 0.0972, both of which are much higher than the prover's support.

Figure 2 illustrates a clear task-dependent shift in routing-signal behavior across datasets. In Figure 2(a), the zero-centered red-white-blue (RdYlBu) Brier Skill Score heat map shows that signal utility varies substantially across task families separated by the dashed boundary between claim-heavy tasks (FEVER, FEVEROUS, and HoVer) and multiple-choice-heavy tasks (GPQA, C-Eval, and MMLU-Pro). Among these patterns, $V_{prob,B\to A}$ is especially strong on MMLU-Pro, where it reaches a BSS of 0.199, while $P_{max,A}$ and $Entropy_A$ dominate on C-Eval, with BSS values of approximately 0.18 and 0.16, respectively. By contrast, Disagreement remains close to zero across datasets, suggesting limited standalone routing value. Figure 2(b) further confirms this family-level transition by showing the average BSS change from claim-heavy to multiple-choice-heavy tasks: confidence-related indicators, especially$P_{max,A}$ and $Entropy_A$, exhibit a marked upward shift in the multiple-choice setting, whereas the divergence-based indicator remains near zero, indicating that confidence signals become substantially more discriminative as the answer space grows.

Table 3. Task-family dependence of the main routing signals (mean BSS)

Feature	Claim-heavy	MC-heavy	MC minus Claim
Disagreement	0.0025	0.0037	0.0012
Entropy_A	0.0108	0.0972	0.0864
Entropy_B	0.0190	0.0387	0.0197
P_max_A	0.0117	0.1048	0.0931
P_max_B	0.0126	0.0316	0.0190
V_prob_B_on_A	0.0290	0.0657	0.0367

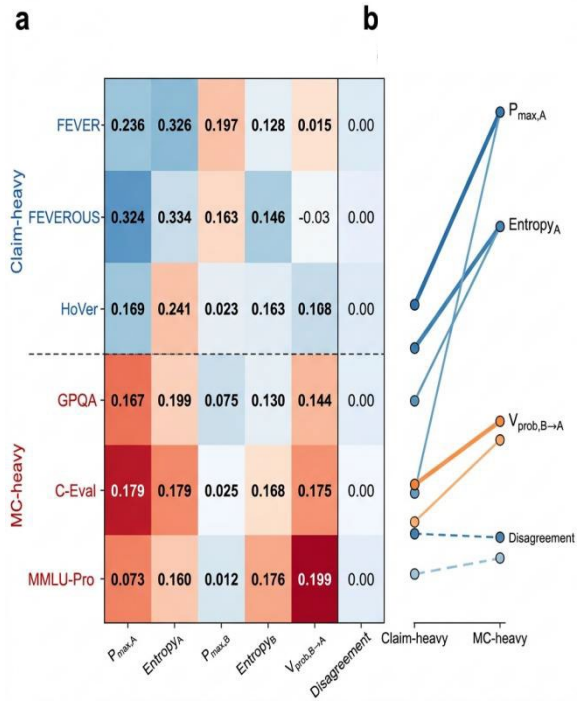

Figure 2. Task-dependent signal landscape. (a) Signal-dataset heat map. (b) Claim-to-MC signal shift

4.2. Signal combination value

Research now moves from the task level to the value of signal combinations. Averaging over all datasets, the best configuration is A+B+agree+V with an average AUC of 0.6507, an average AP of 0.6319, and an average BSS of 0.0834. This exceeds A+B+agree, A+B, and A_only, directly supporting the claim that the main value of cross-validation lies in the combination of signals rather than any single variable. Adding inconsistency on top of A+B+agree+V did not improve the average AUC (Table 4).

Figure 3 decomposes the signal combination results into contribution and gain attributes for each dataset. Figure 3 (a) reveals three empirical patterns: (i) The consistency measure yields the largest single-step AUC improvement (+0.0247 on average between +B and +B+agree across datasets, Table 4), confirming that binary answer consistency is a surprisingly powerful routing heuristic, despite not requiring additional probability computations; (ii) Further additions of V_prob, B→A produce modest additional gains (+ 0.0028 on average), mainly concentrated in MMLU-Pro(+0.014), and almost zero (+ 0.001); (iii) Fold level variance (represented by jitter points) shows little variation across configurations, suggesting that the average improvement is systematic instead of reflecting outlier folding. Figure 3 (b) provides a structural explanation: in the claim-heavy task (FEVER, FEVEROUS), agree dominates the gain decomposition, reflecting the binary nature of the label space, where inter-model agreements carry high information content; On MMLU-Pro (10 options), V_prob, B→A contributes the largest component (+0.014), consistent with the hypothesis that verifier support becomes more discriminative as the answer space expands.

Table 4. Mean performance of progressive signal groups across datasets

Feature group	Mean AUC	Mean AP	Mean BSS
A+B+agree+V	0.6507	0.6319	0.0834
A+B+agree	0.6479	0.6304	0.0814
A+B+agree+V+D	0.6478	0.6285	0.0819
A+B	0.6232	0.6136	0.0673
A_only	0.6139	0.5995	0.0601
B_only	0.5928	0.5762	0.0298

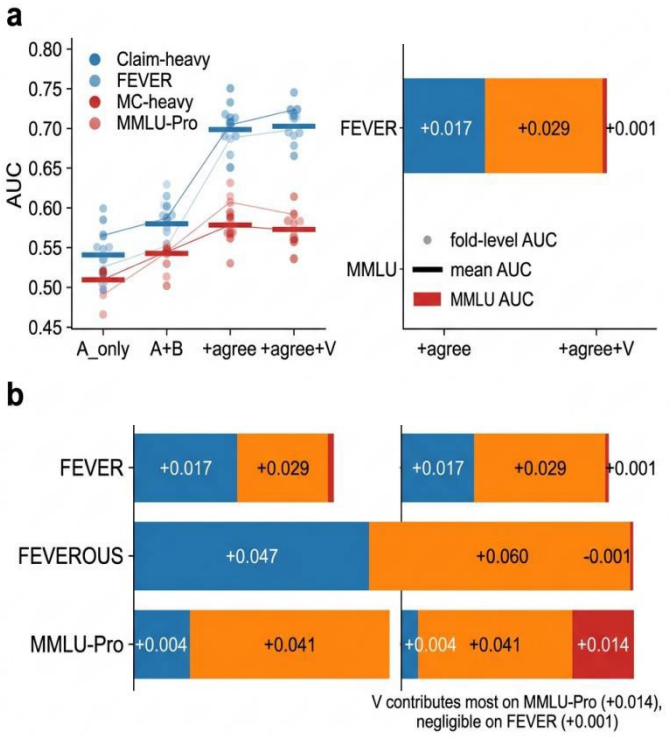

Figure 3. Signal combination value and grouped Shapley attribution. (a) Progressive AUC gains across signal combinations by dataset. (b) Grouped Shapley contribution of self-confidence and cross-verification features across datasets

Shapley attribution grouping

Grouped Shapley analysis asks how many explanatory values should be assigned to the confidence, validated confidence, and cross-validation groups when interactions and collinearity are taken seriously. At the group level, confidence and cross-validation emerged as two dominant groups with mean absolute values of grouped Shapley values of approximately 0.0587 and 0.0540, respectively. The verifier has a much smaller confidence set of about 0.0233 (Table 5).

This result is important because it avoids the simple reading of ranking individual features. Although cue confidence is the strongest mean scalar feature, the grouped attribution analysis shows that the cross-validated

contribution has almost as much explanatory value as the full confidence block.

Per-dataset nuances

The average advantage of A+B+agree+V is not consistent across all six datasets. On GPQA, the simplest configuration A_only, with applicant confidence and entropy, achieves an AUC of 0.572, slightly better than the full combination of 0.565. In C-Eval, A+B (0.736) is slightly higher than A+B+agree+V (0.734). These cases are consistent with the task dependency paper established in Section 4.1: in the MC-heavy setting with limited samples per dataset (GPQA, 546 items) or when questioner confidence already provides a strong routing signal (C-Eval), additional cross-validated features do not improve routing performance. The average advantage of the full combination was mainly driven by claim-heavy task-FEVER (0.666 vs. 0.619 a_ only) and FEVEROUS (0.620 vs. 0.514) and MMLU-Pro (0.748 vs. 0.690). Cross-validation provides a lot of complementary information. This pattern supports the central claim of the paper: external validation remains task dependent and is most evident in combinations of features rather than isolated scalars.

Table 5. Mean grouped Shapley importance across datasets

Feature group	Mean absolute Shapley	Mean signed Shapley
Self-confidence	0.0587	0.0002
Cross-verification	0.0540	0.0021
Verifier-confidence	0.0233	-0.0012

4.3. Single-feature baselines

Once the task landscape was established, single-feature performance was used as a baseline for cross-validated complementarity. The strongest mean univariate baseline is the proposer's own confidence. $P_{max,A}$ has the best weighted average Brier skill score with a platt map of 0.0647, whereas the logistic version achieves 0.0614 (Table 6). The best external validation signal is V_{prob, B\ tarrow A}, whose platt scaled version achieves a weighted average BSS of 0.0592. In contrast, disagreement as an independent feature is weak, with a weighted average BSS of only 0.0036.

This ordering is important for the final narrative of the paper. Cross-validation is not a substitute for the confidence signal. In contrast, proposer confidence remains the strongest mean scalar signal, while verifier support for proposer answers provides the strongest external complement.

Table 6. Weighted cross-dataset summary of single-feature performance after posterior mapping

Feature	Mapping	Weighted mean AUC	Weighted mean AP	Weighted mean BSS	Weighted mean ECE
$P_{max,A}$	Platt	0.6218	0.6144	0.0647	0.0837
$P_{max,A}$	Logistic	0.6218	0.6144	0.0614	0.0871
$V_{prob,B\rightarrow A}$	Platt	0.6227	0.6159	0.0592	0.0902
$Entropy_A$	Logistic	0.6159	0.6077	0.0567	0.0884
$V_{prob,B\rightarrow A}$	Logistic	0.6036	0.6006	0.0508	0.0949
Disagreement	Logistic	0.5051	0.4969	0.0036	0.0899

Next, we study the union semantics of $V_{prob,B\rightarrow A}$ and. The aim is not to argue that disagreement itself is strong, but to determine whether disagreement helps the verifier to support the common interpretation.

The heatmap analysis is consistent with this interpretation. Some datasets exhibit interpretable regions in the joint plane, where high verifier support and moderate disagreement are associated with higher answer reliability than suggested by any marginal viewpoint. Fever is the most obvious example, while GPQA remains difficult in most joint Spaces. Thus, disagreement is not a primary signal, but an interaction sensitive word whose meaning depends on the support of the verifier. Its 0.0036 independent BSS are too small to serve as an independent routing feature, but it is retained in the full 7-feature vector because its joint space structure with $V_{prob,B\rightarrow A}$ is highly informative within structured datasets such as FEVER. Table 4 shows that adding disagreement to A+B+agree+V does not reduce the average AUC, so there is no measurable cost at the aggregate level.

4.4. Generalization across pairs and across tasks

We evaluate generalization along two orthogonal axes (Figure 4). Cross-pair generalization tests whether a router trained on one proposor-prover pair will transfer to the other pair while keeping the task constant. Cross-task generalization tests whether a router fitted on one task family transfers to another task family while keeping pairings constant.

Over-pair generalization. Here, the training pair is Qwen3.5-2B × Phi-4-mini-instruct. The two pairs of tests retained were Llama-3.2-1B-Instruct × Qwen3.5-2B and gemma-3-1b-it × phi-4 mini-instruct. The transfer performance is assessed using a pair of alignment strategies, ECDF and z-score, and the z-score results are excluded when the distribution overlap check fails.

Table 7. Cross-pair generalization AUC (mean over valid test pairs per dataset; ECDF vs z-score alignment)

Dataset	ECDF AUC	z-score AUC	(ECDF z-score)
FEVER	0.753	0.715	+0.038
MMLU-Pro	0.728	0.701	+0.027
GPQA	0.552	0.534	+0.017
FEVEROUS	0.527	0.487	+0.040
HoVer	0.540	—†	—
C-Eval	0.499	0.455	+0.044

† *z-score excluded for HoVer: distribution overlap check failed for both test pairs.*

ECDF alignments strictly dominate the z-score on each effectively compared dataset. The interval ranges from +0.017 (GPQA) to +0.044 (C-Eval) (Table 7). However, the absolute results must be read carefully. Cross-pair transmission achieves AUC above 0.72 on FEVER and MMRU-Pro, indicating that the routing signal structure across model families for these tasks is largely preserved. The transfer is only slightly higher than random in HoVer, FEVEROUS, and GPQA states. The results for C-Eval are particularly worrying: while ECDF has the largest advantage over z-score (+0.044), the absolute value of 0.499 for ECDF AUC is essentially random (Table 8).

Table 8. Per-test-pair cross-pair transfer AUC (ECDF alignment)

Dataset	Test pair	AUC
FEVER	Llama-3.2-1B × Qwen3.5-2B	0.755
FEVER	gemma-3-1b × Phi-4-mini	0.750
MMLU-Pro	gemma-3-1b × Phi-4-mini	0.730
MMLU-Pro	Llama-3.2-1B × Qwen3.5-2B	0.726
GPQA	Llama-3.2-1B × Qwen3.5-2B	0.551
GPQA	gemma-3-1b × Phi-4-mini	0.552
FEVEROUS	gemma-3-1b × Phi-4-mini	0.609
FEVEROUS	Llama-3.2-1B × Qwen3.5-2B	0.445
HoVer	gemma-3-1b × Phi-4-mini	0.557
HoVer	Llama-3.2-1B × Qwen3.5-2B	0.523
C-Eval	gemma-3-1b × Phi-4-mini	0.518
C-Eval	Llama-3.2-1B × Qwen3.5-2B	0.480

Cross-task generalization. We pool Claim-heavy tasks (FEVER, FEVEROUS, HoVer) as one domain and MC-heavy tasks (GPQA, C-Eval, MMLU-Pro) as the other, training on one domain and testing on the other. z-score alignment fails the KL divergence gate in both directions, confirming that cross-task feature distributions are incompatible without nonparametric re-scaling. Under ECDF alignment, the Claim→MC direction reaches AUC 0.643 (AP 0.580) and the MC→Claim direction reaches AUC 0.610 (AP 0.675) (Table 9). Both are meaningfully above random.

Table 9. Cross-task generalization (ECDF alignment; z-score excluded due to KL gate failure)

Direction	Train tasks	Test tasks	AUC	AP
Claim → MC	FEVER + FEVEROUS + HoVer	GPQA + C-Eval + MMLU-Pro	0.643	0.580
MC → Claim	GPQA + C-Eval + MMLU-Pro	FEVER + FEVEROUS + HoVer	0.610	0.675

The asymmetry between the two directions is of reference value. From the AUC metric, the transfer ability from Claim to MC is slightly stronger (0.643 vs 0.610), which is consistent with the research results: in tasks dominated by MC, the confidence signals proposed by the authors hold a dominant position and are relatively stable across different task types. While the transfer accuracy from MC to Claim is higher (0.675 vs 0.580), indicating that the Claim direction has higher accuracy at high confidence thresholds. Overall, neither direction has degenerated into random performance, supporting the use of globally trained routers when task labels cannot be obtained during deployment; this also confirms that when the task identity is known, task-based calibration is still better.

Leave-one-dataset-out cross-task generalization. To complement the rough two-group division, we adopt the single dataset exclusion (LOO) method for assessment: each time, one dataset is alternately used as the test set, while the remaining five datasets are used for training the router, and the test is conducted on this test set. This method decomposes cross-task transfer into the independent contributions of each dataset, avoiding confusion caused by intra-group differences.

Table 10. Three-way generalization comparison

Dataset	In-dist AUC	LOO cross-task	Δ (LOO)	Cross-pair ECDF	Δ (cross-pair)
MMLU-Pro	0.748	0.717	−0.031	0.728	−0.020
C-Eval	0.734	0.616	−0.118	0.499	−0.235
FEVER	0.666	0.664	−0.002	0.753	+0.087
FEVEROUS	0.620	0.591	−0.029	0.527	−0.093
HoVer	0.570	0.558	−0.012	0.540	−0.030
GPQA	0.565	0.577	+0.012	0.552	−0.014

The LOO comparison shows the separation between the two migration axes. For the FEVER task, both axes exhibited stability: the cross-task gap was negligible (−0.002), and cross-aligned migration was actually higher than the upper limit of the same linear distribution by +0.087, indicating that in the fact verification task, the logarithmic probability signal centered on the proposer was largely unaffected by the model identity. MMLU-Pro also showed similar stable migration on both axes (differences of −0.031 and −0.020 respectively), clearly classified as the transferable MC category (Table 10). GPQA presented a different pattern: LOO only slightly exceeded the internal distribution range (+0.012), indicating that training on five different datasets provided a more robust signal basis compared to fitting with only a single dataset for this low-sample benchmark.

C-Eval is a notable exception. The performance of LOO on cross-task tasks has declined (−0.118), indicating that the behavior of the multilingual MC signal is sufficiently different from the other five datasets, even across different task categories, and can still produce detectable distribution shifts. Cross-aligned transfer completely failed (−0.235,

ECDF AUC = 0.499), confirming that the failure of C-Eval is due to architecture-driven rather than task-driven factors: the Chinese-centered word segmentation method generated a log-probability distribution, while the ECDF quantile mapping cannot be aligned between different model families. Overall, these results indicate that in the CrossVerify signal interface, the migration barriers between task families are smaller compared to the obstacles caused by mismatch in architecture and label space.

Figure 4 presents a single-view comparison of the three evaluation modes through parallel coordinates. The visualization results indicate that the generalization behaviors among the different datasets are not consistent. FEVER (dark blue) and MMLU-Pro (light red) are located on the high stability trajectory - their curves remain above AUCUCUC ≈ 0.72 on all three axes, indicating that the routing signal structure of these tasks has strong robustness to changes in the task family and model alignment replacement. However, C-Eval (medium red) shows significant differences between the two transfer dimensions: the LOO cross-task degradation degree is relatively small (-0.118, from 0.734 to 0.616), but cross-alignment transfer is severely catastrophic (-0.235, reaching the level of 0.499). This difference provides empirical evidence for identifying architectural features - spatial mismatch (rather than task domain differences) as the main obstacle to signal transmission: in C-Eval, the Chinese-centered word segmentation generation produced a log-probability distribution, whose quantile structure cannot be aligned between different model families only through ECDF mapping. FEVEROUS also shows a significant inter-data set decline (-0.093), but still above the accidental level, which is consistent with its intermediate position between the fully transferable mode (FEVER) and the non-transferable mode (C-Eval).

Deployment Description: No annotated routes. One of the practical advantages of CrossVerify that deserves special emphasis is that its routing signal vector is entirely generated from the restricted decoding probability distribution during inference, without relying on the correctness labels of specific routes. Routing methods that rely on learning preference supervision or labels based on benchmark data face annotation bottleneck issues during deployment [15, 16]; in contrast, the output space interface of CrossVerify completely circumvents this limitation while retaining the aforementioned cross-alignment and cross-task transferability.

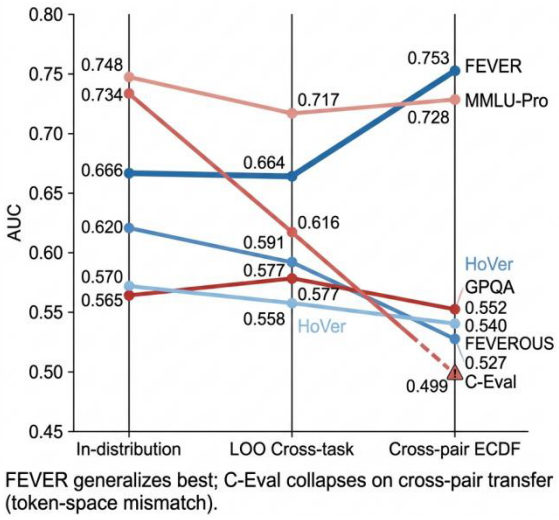

Figure 4. Cross-pair and cross-task generalization

4.5. Multiseed robustness

After conducting tests using multiple random seeds on CEVAL, FEVER, and GPQA, the most stable mode was the proposer-centric mode. On CEVAL, $P_{max,A}$ and $Entropy_A$ both performed strongly and were highly stable, with average BSS values of approximately 0.1125 and 0.1094 respectively. On GPQA, the proposer-centric feature once again dominated. The most important robustness conclusion is that, regardless of the different seeds, the degree of inconsistency remains weak.

4.6. System overhead

In the FEVER benchmark test using the main model, the logprob-based process takes an average of approximately 381.7 milliseconds per sample, while the direct hidden state extraction requires about 327.1 milliseconds. Therefore, in terms of the original request delay, the logprob process is slightly slower; this is expected because the hidden state measurement only generates dense embeddings without marking selection for the probability distribution.

Compared with the single-pass method that solely relies on confidence levels, CrossVerify needs to call a fixed validator once for each sample. Therefore, its end-to-end system cost is not necessarily lower than the baseline scheme that solely relies on confidence levels. If the expected cost of the confidence-based cascade method is $C_{conf} = c_A + p_{conf}c_{L2}$, and the expected cost of CrossVerify is $C_{cv} = c_A + c_B + p_{cv}c_{L2}$, then CrossVerify has a cost advantage only when $c_B < (p_{conf} - p_{cv})c_{L2}$. Therefore, the claim in this paper should be understood as a statement about the low-load external validation at the interface level, rather than a general claim about the total cost of all single-pass baselines.

To illustrate the break-even condition specifically, three deployment modes are considered, each adopting the API pricing in mid-2026. Mode 1 (Homogeneous Small Model):

The proposer and the verifier both use low-cost models (e.g., GPT-4o-mini, $0.15 per million input tokens), and employ a stronger fallback mechanism (GPT-4o, $2.50 per million input tokens). For a typical 500-token query, $c_A \approx c_B \approx \$0.000075$, $c_{L2} \approx \$0.00125$. To achieve break-even, $p_{conf} - p_{cv} > 0.06$ is required. If CrossVerify reduces the fallback rate by at least 6 percentage points compared to the cascading mechanism based solely on confidence levels, it can be achieved.

Mode 2 (Heterogeneous Model): The small proposer (GPT-4o-mini) is paired with the large verifier (GPT-4o, $2.50 per million input tokens). At this point,$c_B \approx \$0.00125$. The break-even requirement is $p_{conf} - p_{cv} > 1.0$, but it cannot be met - the verification cost exceeds any potential savings from reducing the fallback usage.

Mode 3 (API with Token-logprob Access): If the verifier exposes only the logprob values of the first 5 tokens through restricted decoding prompts (as most commercial APIs do), the cost of the verification call is the same as the standard generation call. At this point, the analysis result depends on the model level and reduces to Mode 1 or Mode 2. This cost analysis shows that when the cost of the verifier is equal to that of the proposer, the deployment advantage of CrossVerify is most significant; the value of the interface lies in its black-box compatibility and the 28-byte payload, rather than unconditional cost reduction.

The decisive difference lies in the deployment load. The logprob-based router requires only 7 scalar values per sample ($P_{max,A}$, $Entropy_A$, $P_{max,B}$, $Entropy_B$, $V_{prob,B\to A}$, $Disagreement$, $agree$), each occupying 4 bytes, so each routing decision requires only 28 bytes. This value is a fixed constant and is not affected by the dimension of the hidden state of the model: it does not increase with the model size and does not require embedding transmission.

The more fundamental limitation is that hidden state routing requires white-box access to the internal representations of the two models, which is not available in black-box commercial APIs. The logprob-based design can be applied to any model that exposes token-level probabilities through restricted decoding prompts. As a reference, the top-level embedding of the main pair (2,048 + 3,072 floating-point 32-dimensional vectors) requires a total of 20,480 bytes on each sample. However, in a deployment with restricted access, when only the output space probability can be accessed, this benchmark scheme is not applicable (Table 11). Therefore, CrossVerify's contribution in this aspect lies in its fixed 28-byte memory footprint, independence from model size, and complete compatibility with black-box APIs [9, 13].

Table 11. System-level comparison of logprob routing and hidden-state routing (FEVER, main model pair)

Router type	Description	Mean latency (ms)	Payload / sample (bytes)
Logprob-based	7 scalar features (float32)	381.74	28
Hidden-state	Last-layer embeddings, both models (2,048 + 3,072 dims, fp32)	327.10	20,480

Note: The hidden state baseline requires white-box access to the model internals and is only for reference; under the constraints of black-box deployment, it is not an effective comparison benchmark.

5. Consistency of experience and decision analysis

Sections 4.1 to 4.6 elaborate on the main empirical claims of this paper: CrossVerify provides a set of compact and black-box-compatible external validation signals, whose value depends on the specific task and can serve as a supplement to confidence. The role of this section is relatively limited; it does not provide theorem-level guarantees nor attempts to replace empirical research with an abstract proof stack. Instead, it explores whether the observed empirical behavior conforms to a series of theoretical expectations: the non-redundancy of cross-verification conditions, the stability of classical alignment on continuous feature subspaces, and threshold routing as a budget-sensitive decision principle. Each subsection directly corresponds to an empirical result: Section 5.1 explains the complementary gain shown in Section 4.2 (Tables 4-5); Section 5.2 argues that the ECDF alignment method supports the viewpoint in Section 4.4 (Tables 7-8); Section 5.3 verifies the threshold routing strategy applicable to fixed fallback deployment scenarios. 5.4 Section links the task-dependent gain to the task fuzzy pattern established in Section 4.1 (Table 3).

5.1. Information gain proxy

Section 4.2 shows that adding $V_{prob,B\to A}$ to the A+B configuration can increase the average AUC from 0.6232 to 0.6507 (Table 4), and the grouped Shapley analysis also confirms that the explanatory power of cross-verification is almost equivalent to that of the complete confidence block (Table 5). Now we explore whether this complementarity is based on a formal non-redundancy argument for the correctness of the proposer. We start from the following non-redundancy condition proposition:

$$I(Z_A; S_{cv} \mid S_{self}, X) > 0 \tag{6}$$

This proposition should be understood as the additional information provided by cross-verification under a fixed confidence level. Since directly estimating this conditional mutual information from a small heterogeneous dataset requires a more complex density estimation process, we adopt a restricted empirical proxy strategy. Appropriate scoring rules and information functions act on probability measures, while AUC is a ranking-based statistic and thus cannot be regarded as an affine proxy of information gain.

The resulting evidence has informational value, but is deliberately controlled at a moderate level (Table 12). The relationship between the overall degree of difference among different datasets and the downstream performance improvement is not consistent. FEVER achieved the highest

average Jensen-Shannon proxy index across six datasets, but the absolute performance improvement obtained after adding cross-verification was smaller. In contrast, the average degree of difference for MMLU-Pro was only at a medium level, but when the support of the validator features was added, its group Shapley contribution was the largest and the positive improvement effect was the strongest.

Table 12. Dataset-level summary for information-gain proxies and cross-verification gains

Dataset	JS proxy mean	CV help rate	CV net gain rate	BSS mean	AUC mean	Cross-verification Shapley
CEVAL	0.0396	0.1020	-0.1540	-0.0021	-0.0019	0.0273
FEVER	0.0706	0.0680	0.0030	0.0014	0.0006	0.0415
FEVEROUS	0.0353	0.1910	0.1190	-0.0003	-0.0010	0.0902
GPQA	0.0352	0.1465	-0.0293	0.0001	0.0041	0.0210
HOVER	0.0801	0.0100	-0.0080	0.0000	0.0020	0.0181
MMLU-Pro	0.0447	0.1430	0.0180	0.0129	0.0132	0.1258

Note: The CV assistance rate and CV net increase rate are sampling frequency statistics (i.e., which samples are helpful or harmful to the individual predictions of the cross-verification). ΔBSS and ΔAUC are weighted summary values based on the magnitude of squared error. When the improvement of beneficial samples is less than the loss of damaged samples, the positive net increase rate frequency is compatible with the close-to-zero ΔBSS - this is a typical situation where the base predictor has been adequately calibrated.

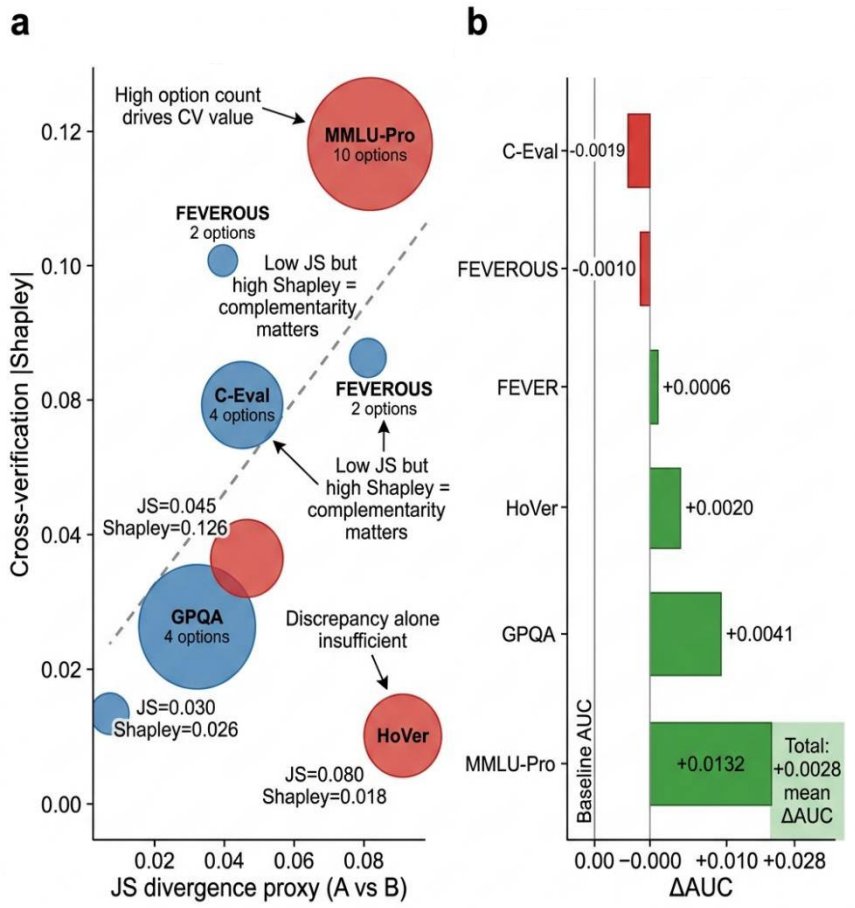

Figure 5. Information gain and task topology. (a) Information gain versus cross-validation contribution. (b) Net gain from adding verifier support

Figure 5 explores the empirical relationship between the differences between the proposer and the verifier, the task structure, and the value of cross-verification routing. Figure 5 (a) shows that the Shapley contribution of cross-verification is not monotonically related to the JS difference proxy variable: the average JS difference of HoVer is the largest (0.080), but its Shapley contribution is the smallest (0.018); while FEVEROUS has a smaller difference (0.035), but its

Shapley contribution is the second highest (0.090). This non-monotonicity directly indicates that the value of cross-verification is determined by complementarity rather than the degree of original inconsistency. The bubble area (proportional to the number of options) provides a topological explanation: MMLU-Pro (10 options, the largest bubble, with a Shapley value of 0.126) shows that as the answer space increases, the value of cross-verification also increases, which is consistent with the theoretical view - a larger option space provides more support opportunities for the verifier to handle the proposer's ambiguity. Figure 5 (b) presents the waterfall decomposition of the net ΔAUC: MMLU-Pro dominates the cumulative gain (+0.0132), while FEVEROUS and C-Eval contribute slightly negative changes (−0.0010 and −0.0019, respectively). The cumulative average ΔAUC (+0.0028) is positive but small, which is consistent with the paper's view: $V_{prob,B\to A}$ is a complementary signal, whose value is obtained by combination rather than appearing as an isolated additive term.

5.2. Stability of canonical alignment

Section 4.4 demonstrated that ECDF alignment enables cross-pair and cross-task transfer with meaningful AUC on FEVER (0.753) and MMLU-Pro (0.728), while C-Eval collapses to 0.499 (Tables 7-8). The research now verifies whether the canonical alignment map preserves the routing-relevant rank structure that such transfer requires. The empirical alignment map preserves the ranking structure needed by routing and transfer. Spearman rank correlations remain high for all variables, reaching 0.9826 for Disagreement, 0.8643 for $V_{prob,B\to A}$, and 0.8392 for $P_{max,B}$. Even the weakest case, $P_{max,A}$, remains at 0.7155. The top-10% and top-20% overlap rates show the same pattern: although the alignment map does not preserve absolute distances, it preserves the high-value regions that matter most for threshold routing (Table 13). We additionally report the one-dimensional Wasserstein distance between raw and aligned values as a transport-based distortion measure.

Table 13. Pooled alignment stability on the continuous feature subspace

Feature	Spearman	Kendall	Top-10% overlap	Top-20% overlap	Bin consistency	W_1(raw,aligned)
$P_{max,A}$	0.7155	0.5717	0.7856	0.7782	0.2409	0.8065
$Entropy_A$	0.5772	0.4544	0.2378	0.4743	0.2908	1.0991
$P_{max,B}$	0.8392	0.6601	0.6523	0.7412	0.3354	0.7738
$Entropy_B$	0.7446	0.5702	0.2396	0.5239	0.3018	0.8290
$V_{prob,B\to A}$	0.8643	0.6938	0.6523	0.7493	0.3210	0.6374
Disagreement	0.9826	0.8892	0.8541	0.9179	0.6230	0.6379

5.3. Threshold routing on a smoothed risk surface

Section 4.4 showed cross-pair AUC up to 0.753 on FEVER (Table 7), establishing that quantile-space routing signals are transferable between model families on factual verification tasks. We now ask whether quantile-space threshold routing also recovers the offline budget-constrained optimum in a concrete fixed-fallback deployment setting. Using the real FEVER proposer outputs and the empirical correctness labels of the stronger non-oracle L2 model, we compare an offline brute-force threshold search, a threshold restricted to the canonical quantile space, and a dual-guided adaptive threshold update defined on a kernel-smoothed objective. The quantile-space threshold nearly matches the offline optimum. The brute-force strategy reaches an accuracy of 0.761 at an average cost of 2.488, while the quantile-space strategy reaches 0.760 at a lower average cost of 2.468. The accuracy gap is only 0.001 (Table 14). This is sufficient to justify the threshold policy as a stable engineering approximation under the fixed-fallback instantiation studied here.

The dual-guided adaptive update should be interpreted more cautiously. In the present implementation, it converges toward a high-cost regime and overshoots the target budget. This does not invalidate the threshold-routing theory itself. It shows that the current online update rule remains a preliminary realization of the theoretical policy class, not yet a deployment-grade controller.

Table 14. Threshold-routing summary under the budget constraint.

Method	Best threshold	Accuracy	Average cost	Route-to-L2 rate	Accuracy gap to brute-force	Cost gap to brute-force
Offline brute-force	0.5239	0.761	2.488	0.372	0.000	0.000
Quantile space	0.5264	0.760	2.468	0.367	-0.001	-0.020
Dual guided	0.1463	0.778	4.984	0.996	0.017	2.496

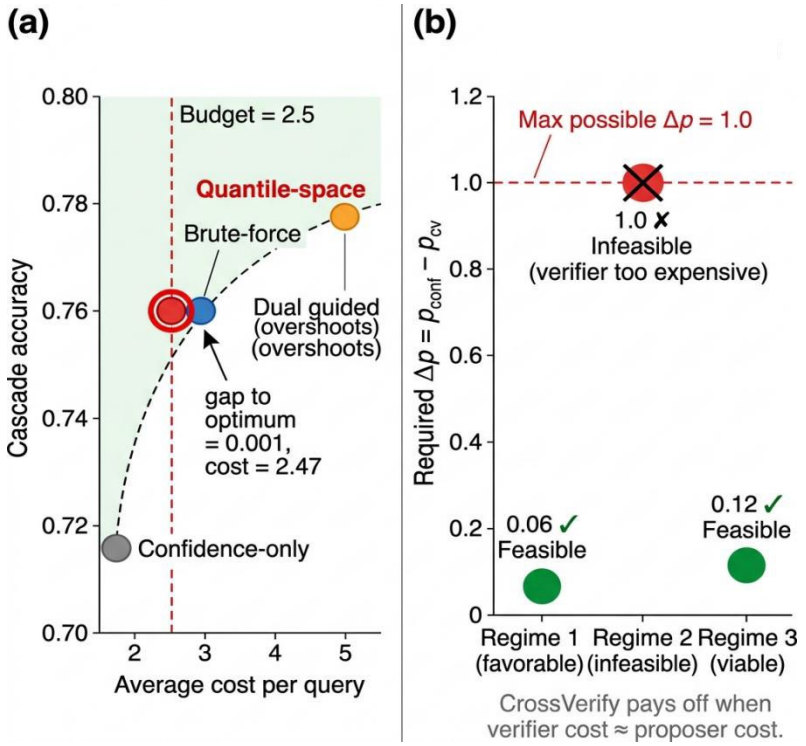

Figure 6. Threshold routing and deployment cost (a) Pareto frontier of accuracy versus cost (b) Break-even condition

Figure 6 translates the threshold-routing results from Table 14 into visual form. Figure 6(a) demonstrates that the quantile-space policy (red marker with highlight ring) lies almost exactly on the Pareto frontier: it achieves accuracy comparable to the offline brute-force optimum (0.760 vs. 0.761) while operating within the budget constraint of 2.5. The dual-guided adaptive point (orange, far right) illustrates the cost of unconstrained dual-ascent optimization: it achieves the highest accuracy (0.778) by routing nearly all samples to the stronger fallback, at nearly double the budget. Figure 6(b) translates the theoretical break-even condition (c_B < (p_conf − p_cv) c_L2) into three representative deployment patterns for commercial API pricing. This visual structure clearly shows that CrossVerify's cost advantage is conditional rather than universal: the strategy works only when the inference costs of the verifier and proposer are equal (Mode 1, $\Delta p \geq 0.06$). When the verifier is an order of magnitude more expensive than the proposer, break-even cannot be achieved (Mode 2, $\Delta p > 1.0$).

5.4. Gamma agents and task dependencies

Section 4.1 shows that in intensive task declarations, $V_{prob,B\to A}$ BSS in Mc-intensive tasks is 0.0290, while the value in Table 3 is 0.0657, establishing a clear task family dependency. Next, we explore whether this variation can be correlated to measurable task characteristics: ambiguity, complementarity, and the topology of the answer space. Finally, we consider a constant dependent on the task

$$\gamma = g(\text{ambiguity},\text{complementarity},\text{topology}),$$

a constant that appears in the weak lower-bound interpretation of cross-verification gains. The goal here is not to estimate γ as a formal parameter, because the number of datasets is too small for that purpose. Instead, an exploratory dataset-level proxy metric is constructed by combining the empirical option count, empirical label entropy, Shapley complementarity between groups, and the measured gain obtained by cross-verification in the routing feature set. This descriptive pattern is reasonable. MMLU-Pro shows the highest gamma-style surrogate values and is also the dataset with the strongest cross-verification in terms of grouped Shapley importance and ablation gain. FEVEROUS exhibits a large Shapley contribution, but the gain in the current routing proxy is close to zero, indicating that complementarity alone is not sufficient unless it can be translated into actual routing-level improvements. See Table 15 for details.

Table 15. Exploratory gamma-proxy summary by dataset

Dataset	Empirical options	Label entropy	Task family	Cross-verification Shapley	BSS mean	AUC mean	Gamma proxy
FEVER	2	0.6377	Claim-heavy	0.0415	0.0014	0.0006	0.000057
FEVEROUS	2	0.6746	Claim-heavy	0.0902	-0.0003	-0.0010	0.000000
HOVER	2	0.6927	Claim-heavy	0.0181	0.0000	0.0020	0.000001
GPQA	4	1.3840	MC-heavy	0.0210	0.0001	0.0041	0.000003

CEVAL	4	1.3855	MC-heavy	0.0273	-0.0021	-0.0019	0.000000
MMLU-Pro	10	2.2946	MC-heavy	0.1258	0.0129	0.0132	0.001617

5.5. Integrated interpretation

In summary, the theory-based tests support a coherent but bounded interpretation of the main empirical studies. The information gain analysis supports the idea that cross-verification is not redundant once the confidence is fixed, but does not prove the general divergence gain law. Alignment experiments show that the canonical map preserves the ordering structure required for routing in continuous subspaces. Threshold experiments show that quantile space threshold routing can almost recover the best offline budget routing policy on real data with fixed fallback Settings. Gamma-style analysis supports a weaker view that task ambiguity and model complementarity help explain when cross-verification becomes more useful.

This is the intended role of this section. It does not replace the primary evidence of earlier empirical chapters, nor does it require formal guarantees. It explains why the observed complementarity, transferability, and threshold routing behaviors are operationally consistent and theoretically sound.

Based on this, a practical routing scheme can be derived. When the task family is known at deployment time, the analysis in Sections 4.1 (Table 3) and 5.4 (Table 15) supports heuristics in two cases: For demanding tasks (binary label space), $V_{prob,B\to A}$ and binary protocol features have a disproportionate predictive weight, reflecting the verifier support advantage documented in Table 3; For MC-heavy tasks (large option Spaces), $P_{max,A}$ is the most stable dominant signal, and $V_{prob,B\to A}$ is a secondary supplementary contribution. This ordering is consistent with the topological regression coefficients from Section 5.4, where $V_{prob,B\to A}$ indicates the largest positive response to option count (+0.028 per unit). When task identity is not available at deployment time, the full seven feature combinations serve as a reliable fallback: Table 10 shows that LOO has an AUC of 0.558 to 0.717 across the six benchmarks. Learned topology-aware signal selection that automatically recognizes such structures -without task family labels -remains a direction for future work.

6. Conclusion

This paper studies a practical system problem that fuses LLM collaboration, confidence estimation, and budget-aware reasoning: how to design routing signals for heterogeneous black-box cascades when hidden state access is not available or desired. Instead of assuming that the second model's own confidence is sufficient, we study cross-verification as a framework for signal design, which explicitly measures the strength of the verifier's support for the proposer's answer. This yields a series of output derived signals that can be computed with minimum interface requirements and integrated into a calibrated routing policy.

The empirical results support a clear and internally consistent interpretation. First, proposer confidence remains the strongest average univariate baseline. Second, the most valuable external verification signal is not the verifier's confidence, but the verifier's support for the proposer's answer, which is V_{prob, B\rightarrow A}. Third, the utility of cross-verification is in feature combinations, not isolated scalars. Ablation and grouped Shapley analyses show that cross-verification is structurally important, even if it is not the primary univariate. Fourth, the usefulness of these signals is strongly task dependent: proposition tasks benefit more from verifier support signals, while proposer confidence remains dominant in multiple-choice tasks. Fifth, Table 1 shows that the budget routing setup is instantiated with an empirically stronger fallback model instead of an oracle, which is the correct interpretation of the escalation analysis in this paper. Finally, our system-level analysis shows that the main deployment advantages of cross-verification are black-box compatibility and very low overhead. Any end-to-end cost gain over the single-pass baseline depends on the verifier cost being offset by the reduced alternate usage.

Empirical consistency tests further shed light on these findings. Cross-verification is conditionally nonredundant over confidence, canonical alignment preserves the sequential structure associated with routing on continuous feature subspaces, and quantile space threshold routing is a principled approximation of the budget-constrained cascading decision problem when upgraded by a fixed stronger fallback instantiation. At the same time, this section also places important restrictions on the claims that can be made. This paper does not show that cross-verification is generally dominant, that disagreement itself is a strong independent routing feature, or that the analysis achieves theorem-level guarantees. Rather, it shows that externally verified supporting signals under the fixed fallback mechanism studied here provide a theoretically plausible and empirically useful complement to confidence under heterogeneous black-box collaboration.

Taken together, these results suggest a revised view of cascading design for LLM systems. The key challenge is not to search for a single universally strongest scalar routing statistic. It is to build a small, calibrated, and transitive signaling interface that can draw on complementary information across heterogeneous models without breaking deployment constraints. Cross-validation provides a concrete signal design perspective for future black-box expert systems, especially when deployments include specified stronger fallbacks rather than oracle upgrades.

7. Limitations and future work

This evaluation examines three proposer-verifier pairs out of the four model backbones, with the master pair used for the main result and the reserved pair used as a transfer check. In budget routing analysis, escalation is instantiated with a fixed stronger alternative model instead of oracle or model-agnotic utility objectives. This setting is appropriate for two-phase deployments with specified stronger fallback alternatives, but it also means that the results should be interpreted as fixed fallback path discovery rather than generalized upgrade theory. The task dependency of the prover support signal, one of the core findings, further limits the portability of any fixed heuristic. Near-term practical implications of both cases are discussed in Section 5.5, but learned topology-aware signal selection remains an unsolved direction, as later systems may need to estimate appropriate signal weights using query statistics. Subsequent work could extend the composition to multiple alternative models, closed-source commercial apis, and a wider range of tokenizers.

The system-level advantage of cross-validation is conditional rather than universal. Since each query still calls the proposer and the verifier, the cost of cross-validation is not automatically lower than the cost of single-pass confidence concatenation; Any end-to-end savings will only occur if the additional cost to the verifier is offset by a lower use of strong alternatives. A low-load, black-box compatible verification interface and a break-even deployment framework are built, but all relevant cost mechanisms are not explicitly benchmarked. Further work should evaluate this trade-off under different verifier/fallback scheme cost ratios and deployment Settings, and directly compare to single-pass trust cascades.

Information gain and gamma analysis use empirical proxies rather than direct estimates; The results should be viewed as consistency checks rather than formal proofs, which is the appropriate scope for signal design contributions. The dual-bootstrapped adaptive threshold strategy in Section 5.3 is a proof-of-concept of a budget dual-architecture rather than a production-ready controller, whereas the evaluation framework remains completely offline. Future work should develop richer nonparametric CMI estimators, clearer norm-aligned finite-sample transmission bounds, more stable online optimization of adaptive routes, and more extensive deployment studies under realistic computational constraints, confidence induction, and benchmark changes [17,20,26].

References

[1] Varangot-Reille C, Bouvard C, Ciancone M, et al. Doing more with less: A survey on routing strategies for resource optimisation in large language model-based systems. J. Artif. Intell. Res. 2026;86:865-893. https://doi.org/10.1613/jair.1.19801.

[2] Dong Y, Ito T. From consensus theory to LLM agents: Practical consensus-building for multi-issue negotiation. Expert Syst. Appl. 2026;322:132250. https://doi.org/10.1016/j.eswa.2026.132250.

[3] Huang J, Han T, Yu N, Yi X. Socratic elenchus-inspired multi-agent debate for mitigating hallucinations in large language models. Expert Syst. Appl. 2026;320:132208. https://doi.org/10.1016/j.eswa.2026.132208.

[4] Valkanas A, Pal S, Rumiantsev P, Zhang Y, Coates M. C3PO: Optimized large language model cascades with probabilistic cost constraints for reasoning. In: Advances in Neural Information Processing Systems. 2025;38:84481-84522.

[5] Luo H, Liu Y, Zhang R, et al. Toward edge general intelligence with multiple-large language model (Multi-LLM): Architecture, trust, and orchestration. IEEE Trans. Cogn. Commun. Netw. 2025;11(6):3563-3585. https://doi.org/10.1109/TCCN.2025.3612760.

[6] Hendrickx K, Perini L, Van der Plas D, Meert W, Davis J. Machine learning with a reject option: A survey. Mach. Learn. 2024;113(5):3073–3110.

[7] Silva Filho TM, Song H, Perello-Nieto M, Santos-Rodriguez R, Kull M, Flach P. Classifier calibration: A survey on how to assess and improve predicted class probabilities. Mach. Learn. 2023;112(9):3211–3260.

[8] Hullermeier E, Waegeman W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. Mach. Learn. 2021;110(3):457–506.

[9] La Malfa E, et al. Language-models-as-a-service: Overview of a new paradigm and its challenges. J. Artif. Intell. Res. 2024;80:1497–1523.

[10] Dong Y, et al. Safeguarding large language models: A survey. Artif. Intell. Rev. 2025;58(12):382. https://doi.org/10.1007/s10462-025-11389-2.

[11] Coussement K, Abedin MZ, Kraus M, Maldonado S, Topuz K. Explainable AI for enhanced decision-making. Decis. Support Syst. 2024;184:114276.

[12] Chen J, Mueller J. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024. p. 5186–5200.

[13] Lin Z, Trivedi S, Sun J. Generating with confidence: Uncertainty quantification for black-box large language models. Trans. Mach. Learn. Res. 2024. https://openreview.net/forum?id=DWkJCSxKU5.

[14] Kapoor S, et al. Large language models must be taught to know what they don't know. In: Advances in Neural Information Processing Systems. 2024;37:85932-85972.

[15] Wang X, et al. MixLLM: Dynamic routing in mixed large language models. In: Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2025. p. 10001–10018.

[16] Shnitzer T, Ou A, Silva M, et al. Large language model routing with benchmark datasets. In: Proceedings of the First Conference on Language Modeling (COLM). 2024.

[17] Shen Y, Liu Y, Huang Z, Yin R, Zheng X, Huang X. SATER: A self-aware and token-efficient approach to routing and cascading. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. 2025. p. 10515–10529.

[18] Chen B-W, Chen C-C, Yen A-Z. Confidence-driven multi-scale model selection for cost-efficient inference. In: Findings of the Association for Computational Linguistics: EACL 2026. 2026. p. 1760–1770.

[19] Shirkavand R, Gao S, Yu P, Huang H. Cost-aware contrastive routing for LLMs. In: Advances in Neural Information Processing Systems. 2025;38. https://arxiv.org/abs/2508.12491.

[20] Felicioni N, Maystre L, Ghiassian S, Ciosek K. On the importance of uncertainty in decision-making with large language models. Trans. Mach. Learn. Res. 2024. https://openreview.net/forum?id=YfPzUX6DdO.
[21] Bhattacharjya D, et al. SIMBA UQ: Similarity-based aggregation for uncertainty quantification in large language models. In: Findings of the Association for Computational Linguistics: EMNLP 2025. 2025. p. 15880–15894.
[22] Mahaut M, Aina L, Czarnowska P, Hardalov M, Müller T, Marquez L. Factual confidence of LLMs: On reliability and robustness of current estimators. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024. p. 4554–4570.
[23] Khanmohammadi R, et al. Calibrating LLM confidence by probing perturbed representation stability. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. 2025. p. 10448–10514.
[24] Hong R, Zhang H, Pang X, Yu D, Zhang C. A closer look at the self-verification abilities of large language models in logical reasoning. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). 2024. p. 900–925.
[25] Zhang Y, et al. Small language models need strong verifiers to self-correct reasoning. In: Findings of the Association for Computational Linguistics: ACL 2024. 2024. p. 15637–15653.
[26] Xiong M, et al. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In: The Twelfth International Conference on Learning Representations. 2024. https://openreview.net/forum?id=gjeQKFxFpZ.
[27] Trapeznikov K, Saligrama V, Castanon DA. Multi-stage classifier design. Mach. Learn. 2013;92(2–3):479–502.
[28] Wang Z, Wang Q, Zhang Y, Chen T, Zhu X, Shi X, Xu K. SConU: Selective Conformal Uncertainty in Large Language Models. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 19052–19075). Association for Computational Linguistics. 2025. https://doi.org/10.18653/v1/2025.acl-long.934.
[29] Zoppi T, Popov P. Confidence ensembles: Tabular data classifiers on steroids. Inf. Fusion. 2025;120:103126.
[30] Haas S, Hüllermeier E. Conformalized prescriptive machine learning for uncertainty-aware automated decision making: The case of goodwill requests. Int. J. Data Sci. Anal. 2024;20(3):2061–2077.
[31] Longo L, et al. Explainable artificial intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. Inf. Fusion. 2024;106:102301.
[32] Lyu Q, et al. Calibrating large language models with sample consistency. Proc. AAAI Conf. Artif. Intell. 2025;39(18):19260–19268.