# Decentralized Aggregation of LLM Predictions via Wagering Mechanisms

**Anonymous Authors**[1]

## Abstract

We propose WALLA, a decentralized mechanism for aggregating probabilistic predictions from multiple LLMs using wagering mechanisms. Each model reports a prediction and a learned wager reflecting its expected score advantage; predictions are then aggregated using wagers as weights. Our mechanism introduces a leave-one-out baseline that yields four key properties: dominant-strategy incentive compatibility under general belief structures, a best-response wager proportional to expected score advantage, decoupling of prediction and wager optimization and bounded worst-case deficit independent of the number of participants. Experiments on QA benchmarks and a forecasting benchmark show that WALLA matches centralized baselines while simultaneously achieving advantage-weighted aggregation, uncertainty-awareness, fully decentralized learning, and incentive-compatibility guarantees.[1]

## 1. Introduction

Having different large language models—each with distinct capabilities, domain expertise, or access to private tools and data—collaborate on complex tasks is becoming increasingly common. In this work, we focus on forecasting settings, where probabilistic predictions from diverse LLMs are elicited and aggregated to improve overall accuracy. We propose designing a decentralized mechanism that, for each domain-specific question, incentivizes models expecting to outperform the pool to participate and contribute with high weight, while those anticipating being outperformed are naturally excluded.[2]

This goal has a clean realization in *wagering mechanisms* where each participant reports a prediction and places a wager—a stake reflecting confidence in that prediction. The mechanism then rewards those whose predictions are accurate relative to others, scaled by the amount wagered. Interpreting wagers as aggregation weights allows the mechanism to simultaneously incentivize truthful predictions and elicit which models should meaningfully contribute.

The canonical weighted-score wagering mechanism (WSWM) (Lambert et al., 2015) satisfies key incentive properties but has two features that limit its applicability to learning-based agents. First, its incentive compatibility relies on *immutable beliefs*—the assumption that agents would not revise their posteriors upon observing others' reports. Second, as the net payout of WSWM is linear in agent wager, a rational agent's optimal wager is always at the boundary, yielding a binary signal that becomes less informative for weighted aggregation or for learning to an agent's comparative advantage.

**Our contribution.** We propose a family of *advantage-aligned wagering mechanisms* that resolves both limitations through a single structural modification to the payout function. This modification yields four main properties that facilitate the participation of learning-based agents: (1) dominant-strategy incentive compatibility under general belief structures: truthful prediction remains optimal regardless of opponents' strategies and how agents' beliefs are formed, (2) score–wager alignment: the best-response wager is proportional to the agent's expected score advantage, (3) the decoupling of prediction and wager learning, and (4) the worst-case deficit of the mechanism is bounded by a constant independent of the number of participants. We refer to our method as the *advantage-aligned wagering mechanism for LLM aggregation* (WALLA). Unlike heuristic confidence measures, the equilibrium wager has a precise interpretation (i.e., expected score advantage) and is elicited through a payout function that makes honest self-assessment the dominant strategy.

Empirically, we conduct experiments on multi-agent prediction aggregation where LLM agents use their model hidden states to learn a wager network. We evaluate WALLA across three scenarios: homogeneous models with private contexts, heterogeneous models, and heterogeneous models with private contexts, demonstrating that the wager network learns to identify comparative advantage from diverse sources:

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]The source code is provided in https://anonymous.4open.science/r/MultiLLMs-B48C/README.md.

[2]We leave discussions on related work on LLM aggregation, wagering mechanisms, and LLM for forecasting to Appendix A.

context assignment, domain expertise, and their combination. While our primary motivation is event forecasting, we heavily utilize QA benchmarks for our evaluation, as they are effective proxies for real-world forecasting that requires specialized knowledge. WALLA matches centralized aggregation baselines (stacked generalization, learned routers) on four QA benchmarks and a forecasting benchmark, while being the only method that simultaneously achieves weighted aggregation, uncertainty-awareness, and fully decentralized learning with incentive-compatibility guarantees.

## 2. Preliminaries

We consider a probabilistic prediction setting in which a center elicits predictions on uncertain outcomes from multiple agents and evaluates them based on the quality of their predictions. Let $\mathcal{M} = \{1, 2, \ldots, M\}$ denote a set of $M$ agents. A *wagering mechanism* operates in two stages. In the first stage, given a question $x \in \mathcal{X}$ with unknown outcome $y \in \mathcal{Y}$, each agent $i$ submits a prediction $p_i \in \Delta(\mathcal{Y})$ and wagers some non-negative amount $w_i \in \mathbb{R}_{\geq 0}$ on her prediction. We denote the resulting *prediction* and *wager profiles* by $\mathbf{p} \coloneqq (p_1, \ldots, p_M)$ and $\mathbf{w} \coloneqq (w_1, \ldots, w_M)$, and write $\mathbf{p}_{-i}$ and $\mathbf{w}_{-i}$ for the corresponding profiles excluding agent $i$. In the second stage, after the outcome is revealed, the mechanism determines the *net payoff* to each agent, denoted $\pi_i(\mathbf{p}, \mathbf{w}, y) \in \mathbb{R}$, based on the quality of her prediction and the amount wagered.

The canonical weighted-score wagering mechanism (WSWM) (Lambert et al., 2015) adopts the following net payout function for agent $i$ under a strictly proper scoring rule $s$, such as the Brier score (Brier, 1950).

$$\pi_i^{\mathrm{WSWM}}(\mathbf{p}, \mathbf{w}, y) = w_i \left( s(p_i, y) - \frac{\sum_{j=1}^{M} w_j s(p_j, y)}{\sum_{j=1}^{M} w_j} \right). \quad (1)$$

WSWM satisfies desirable properties, including incentive compatibility (IC) under immutable beliefs, individual rationality (IR), budget balance, and so on. The extended preliminary can be found in Appendix B.

## 3. Design Wagering Mechanisms for LLM Aggregation

This section presents our proposed *advantage-aligned wagering mechanism*. We start by analyzing our mechanism under the *one-shot, single-question* setting (Sections 3.1–3.2), where each agent holds a belief and acts to maximize her expected payoff. Section 3.3 then extends to the *multiple-question* setting, formalizing how LLM agents can learn wager policies from observed payouts as a function of their hidden states, which encode the question and any context. Two standard aggregation functions, *linear pooling* and *logarithmic pooling*, which will be discussed in Appendix E.8.

### 3.1. Setting, Information Model, Strategies

We now formalize the setting, information model, and agent strategy. Let $(\Omega, \mathcal{F})$ be a measurable space, where $\Omega$ is the set of possible states of the world and $\mathcal{F}$ is the $\sigma$-algebra of measurable events. Each state $\omega \in \Omega$ encodes the question $X$, the outcome $Y \in \mathcal{Y}$, and all agents' private signals $(S_1, \ldots, S_M)$. The private signal $S_i$ represents additional information available to agent $i$ beyond the question itself, such as outputs from tool calls or extra context. The information available to agent $i$ is captured by $\mathcal{F}_i = \sigma(X, S_i) \subseteq \mathcal{F}$: all agents observe the question $X$, but may differ in $S_i$.Each agent $i$ holds a belief $\mathcal{Q}_i$, a probability measure on $(\Omega, \mathcal{F})$ that may differ from the data-generating distribution $\mathcal{P}$.

A strategy for agent $i$ is a pair $(p_i, w_i)$, where $p_i \colon \Omega \to \Delta(\mathcal{Y})$ is a probabilistic prediction and $w_i \colon \Omega \to \mathbb{R}_{\geq 0}$ is a wager. Both are required to be $\mathcal{F}_i$-measurable.[3] All agents are assumed *risk-neutral*, optimizing expected net payout on each individual question. For notational simplicity, we suppress the explicit dependence on $\omega \in \Omega$ using $p_i$, $w_i$, and $y$ to denote $p_i(\omega)$, $w_i(\omega)$, and $Y(\omega)$ respectively.

### 3.2. An Advantage-Aligned Wagering Mechanism

**Definition 3.1** (Net Payout Function)**.** Given a strictly proper scoring rule $s$ and a constant $c_3 > 0$, the *net* payout of agent $i$ upon the realization of outcome $y \in \mathcal{Y}$ is

$$\pi_i(\mathbf{p}, \mathbf{w}, y) := w_i(s(p_i, y) - s_{-i}(\mathbf{p}_{-i}, \mathbf{w}_{-i}, y) - c_3 w_i)\,, \quad (2)$$

where $s_{-i} \colon \Delta(\mathcal{Y})^{M-1} \times \mathbb{R}_{\geq 0}^{M-1} \times \mathcal{Y} \to \mathbb{R}$ is the *leave-one-out baseline* score: it depends only on other agents' predictions and wagers, not on $(p_i, w_i)$. The specific form of $s_{-i}(\cdot)$ is left free; Appendix E.6 instantiates two concrete choices that trade off normality with no-arbitrage. Both variants admit a constant bound on the worst-case deficit.

We show two main properties of this net payout function: the first establishes the optimal prediction for *any* wager; the second establishes that the best-response wager elicits the agent's *expected score advantage*. We defer proofs of this section to Appendix E.

**Theorem 3.2** (DSIC of Prediction)**.** *Under the payout function (2), truthful prediction is a dominant strategy: each agent maximizes her expected payoff by reporting her own belief about $Y$, regardless of opponents' strategies and her own wager. Formally, for any belief $\mathcal{Q}_i$ held by agent i, any wager profile $\mathbf{w} \in \mathbb{R}_{\geq 0}^{M}$, and any $\mathbf{p}_{-i} \in \Delta(\mathcal{Y})^{M-1}$,*

$$p_i^* := \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i) \quad (3)$$

$$\in \arg\max_{p_i \in \Delta(\mathcal{Y})} \mathbb{E}_{\mathcal{Q}_i}[\pi_i((p_i, \mathbf{p}_{-i}), \mathbf{w}, Y) \mid \mathcal{F}_i]\,, \quad (4)$$

*and the maximizer is unique when $w_i > 0$.*

[3]For LLMs, $\mathcal{F}_i$-measurability is automatic: for fixed model weights (e.g., during inference), the model's output depends only on input $(X, S_i)$.

*Table 1.* Categorization of baselines.

| Methods | Uncertainty-Aware | Decentralized | Learning | IC / Manipulation-Resistant |
|---|---|---|---|---|
| Pre-inference Router (Lu et al., 2023; Chen et al., 2024; Ong et al., 2025; Song et al., 2025) | ✗ | ✗ | ✓ | – |
| Stacked Generalization (Wolpert, 1992) | ✓ | ✗ | ✓ | – |
| Uniform Averaging (Wang et al., 2022; Li et al., 2024; Yu et al., 2024) | ✗ | ✓ | ✗ | ✗ |
| Weight by Model Confidence (Kang et al., 2025) | ✓ | ✓ | ✗ | ✗ |
| Weight by Context Perplexity (Mavromatis et al., 2024) | ✓ | ✓ | ✗ | ✗ |
| WALLA (Ours) | ✓ | ✓ | ✓ | ✓ |

Two features deserve emphasis. *First*, our DSIC holds under any belief structure, including mutable beliefs: if an agent revises $\mathcal{Q}_i$ after observing others' reports, the updated posterior remains optimal. *Second*, the optimal prediction does not depend on the agent's own wager $w_i > 0$, decoupling prediction from wagering and enabling the learning pipeline of Section 3.3.

**Theorem 3.3** (Score-Wager Alignment)**.** *Under the payout function* (2)*, the best-response wager is proportional to the agent's expected score advantage. Formally, for any belief $\mathcal{Q}_i$ held by agent $i$, any $\mathbf{p} \in \Delta(\mathcal{Y})^M$, and any $\mathbf{w}_{-i} \in \mathbb{R}^{M-1}_{\geq 0}$, the best-response wager for agent $i$ is*

$$w_i^* = \left(\frac{A_i}{2c_3}\right)^+, \qquad A_i := \mathbb{E}_{\mathcal{Q}_i}[s(p_i, Y) - s_{-i} \mid \mathcal{F}_i], \tag{5}$$

*where $(x)^+ := \max(x, 0)$ and $A_i$ is the agent $i$'s* expected score advantage *over the baseline.*

Three features follow from Theorem 3.3. *First*, agents with $A_i \leq 0$ wager zero and are automatically excluded from aggregation. *Second*, the best-response wager is well defined for any prediction $p_i$, not just the optimal one, allowing the wager to be learned separately without altering the base model's predictions. *Third*, under the Brier score and truthful prediction, $\mathbb{E}_{\mathcal{Q}_i}[s(p_i, Y) \mid \mathcal{F}_i] = c_1 - c_2 \operatorname{Var}_{\mathcal{Q}_i}(\delta_Y \mid \mathcal{F}_i)$, so higher predictive uncertainty reduces the expected self-score and hence the wager.[4]

Heterogeneous predictions can stem from two sources: differences in information or differences in beliefs. Under common prior belief, a BNE can be characterized, while with heterogeneous beliefs, NE can be characterized. See examples and the equilibrium analysis in Appendix E.3.

### 3.3. Learning a Wager Policy from Payouts

In repeated predictions, agents can learn wager policies from realized outcomes. Let $h_i = f_i(x, s_i)$ denote the hidden representation of agent $i$'s LLM on question $x$ and private signal $s_i$. The base LLM produces a probabilistic forecast $p_i = \phi_i(h_i)$, while a small wager network $w_i = g_{\theta_i}(h_i)$ outputs a nonnegative scalar wager. Only $g_{\theta_i}$ is trained while LLM parameters are frozen. For each resolved question, agent $i$ observes its own wager and payout. Since $\pi_i = w_i(s_i - s_{-i} - c_3 w_i)$, the realized score differential $s_i - s_{-i}$ can be recovered from the payout whenever $w_i > 0$. The wager network is then trained to maximize the empirical net payout, $\mathcal{L}(\theta_i) = -\frac{1}{T}\sum_{t=1}^{T}\left[g_{\theta_i}(h_{it})(s_{it} - s_{-i,t}) - c_3 g_{\theta_i}(h_{it})^2\right]$. At optimum, the learned wager approximates the positive expected score advantage conditioned on the model's hidden state.

This learning problem is substantially simpler than improving the prediction itself. The wager network does not need to learn the full conditional distribution of the outcome. It only needs to learn when the existing model is likely to outperform the pool. This distinction is especially valuable for aggregation of proprietary or frozen LLMs, where fine-tuning the base model may be unavailable.

## 4. Empirical Evaluation

### 4.1. Experiment Setup

All participants are LLMs. We draw from a pool of four models spanning general-purpose and domain-specialized capabilities: Gemma-2-9B (Gemma2) (et al., 2024b), Llama3.1-8B (Llama3.1) (et al., 2024a), Llama3-Aloe-8B (Aloe) (Gururajan et al., 2024), a medical domain fine-tuned model, and BioMistral-7B (BioMistral) (Labrak et al., 2024), pre-trained on biomedical corpora. Each model produces a probability distribution over the outcome space from its output token logits, following standard practice for multiple-choice and forecasting evaluations.

We evaluate on six datasets covering question answering and event forecasting: PubMedQA (Jin et al., 2019), MedMCQA (Pal et al., 2022), MMLU (Hendrycks et al., 2021), ARC-Challenge (Clark et al., 2018), and BayesX, a synthetic Bayesian forecasting dataset we construct. Existing QA datasets mostly test whether additional context improves prediction quality. In forecasting, however, additional information can reduce *epistemic uncertainty* while increasing *aleatoric uncertainty* about the realized outcome. BayesX captures this setting: prior probabilities can be extreme, and the private context moves the posterior closer to uniform, making it a challenging test of whether an aggregation method can identify the informed model even when its prediction appears less confident. PubMedQA and BayesX include question-related contexts that may be selectively revealed to a subset of participants, enabling experiments with heterogeneous information. Details about datasets are provided in Appendix F.6.

[4] Overconfident models underestimate this variance and may over-wager; calibration on realized payouts can help (Sec. F.9).

*Table 2.* Four heterogeneous models with private contexts on the PubMedQA dataset. For each question, we randomly assign one model with a relevant context (left columns) or two models, each with a relevant and an irrelevant context (right columns). All reported metrics in tables are scaled by $\times 100$, and all reported intervals ($\pm$) are 95% confidence intervals over five independent runs.

| | One Relevant Context | | | | | | One Relevant Context & One Irrelevant Context | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ |
| UniformAvg | 73.43 | 76.46 | 11.51 | - | 0.00 | 24.28 | 72.76 | 74.60 | 11.32 | - | 0.00 | 27.66 |
| SelfCertainty | 69.84 | 77.25 | 3.13 | 86.11 | 46.30 | 21.60 | 66.58 | 71.40 | 10.89 | 83.61 | 39.68 | 31.20 |
| PackLLM | 77.85 | 83.60 | 15.38 | 66.42 | 12.32 | 18.05 | 71.95 | 77.02 | 12.86 | 58.23 | 5.39 | 26.41 |
| RouterDC w/ context | $73.33_{\pm 0.63}$ | $76.55_{\pm 1.26}$ | $11.58_{\pm 1.91}$ | $54.66_{\pm 37.92}$ | $10.64_{\pm 64.71}$ | $24.23_{\pm 0.34}$ | $72.53_{\pm 1.44}$ | $74.90_{\pm 2.94}$ | $11.74_{\pm 3.55}$ | $48.77_{\pm 6.51}$ | $-2.19_{\pm 11.98}$ | $27.65_{\pm 0.52}$ |
| NIRTRouter w/ context | $73.20_{\pm 0.58}$ | $76.92_{\pm 0.84}$ | $12.04_{\pm 7.80}$ | $43.90_{\pm 50.11}$ | $-7.29_{\pm 152.52}$ | $24.15_{\pm 3.73}$ | $72.53_{\pm 0.34}$ | $75.40_{\pm 0.00}$ | $12.27_{\pm 4.23}$ | $43.15_{\pm 69.64}$ | $-7.18_{\pm 130.40}$ | $27.48_{\pm 2.74}$ |
| RouteLLM w/o context | $72.15_{\pm 5.90}$ | $76.27_{\pm 0.21}$ | $8.26_{\pm 1.51}$ | $55.46_{\pm 56.78}$ | $11.11_{\pm 36.00}$ | $23.09_{\pm 3.05}$ | $68.97_{\pm 12.13}$ | $75.81_{\pm 3.57}$ | $5.91_{\pm 3.73}$ | $62.59_{\pm 7.84}$ | $15.84_{\pm 1.12}$ | $25.20_{\pm 5.89}$ |
| RouteLLM w/ context | $74.46_{\pm 0.57}$ | $79.11_{\pm 0.59}$ | $5.66_{\pm 1.40}$ | $60.84_{\pm 2.83}$ | $14.24_{\pm 2.72}$ | $19.18_{\pm 0.32}$ | $73.19_{\pm 0.72}$ | $77.89_{\pm 0.64}$ | $5.97_{\pm 1.35}$ | $62.18_{\pm 2.94}$ | $15.00_{\pm 2.44}$ | $22.28_{\pm 0.32}$ |
| StackedGen | $79.28_{\pm 0.36}$ | $81.02_{\pm 0.20}$ | $5.79_{\pm 0.24}$ | $68.41_{\pm 0.10}$ | $23.03_{\pm 0.11}$ | $15.72_{\pm 0.12}$ | $73.65_{\pm 0.48}$ | $78.60_{\pm 0.42}$ | $8.52_{\pm 0.55}$ | $67.92_{\pm 1.11}$ | $18.55_{\pm 0.75}$ | $22.12_{\pm 0.53}$ |
| WALLA I | $79.76_{\pm 0.65}$ | $82.11_{\pm 0.18}$ | $11.58_{\pm 0.52}$ | $72.59_{\pm 0.75}$ | $28.06_{\pm 0.59}$ | $16.95_{\pm 0.05}$ | $77.98_{\pm 0.37}$ | $81.15_{\pm 0.18}$ | $11.88_{\pm 0.35}$ | $74.98_{\pm 0.81}$ | $28.38_{\pm 0.32}$ | $20.09_{\pm 0.09}$ |
| WALLA II | $79.73_{\pm 0.23}$ | $82.19_{\pm 0.31}$ | $11.30_{\pm 0.58}$ | $72.31_{\pm 0.47}$ | $27.02_{\pm 0.59}$ | $16.72_{\pm 0.09}$ | $77.60_{\pm 0.05}$ | $80.84_{\pm 0.24}$ | $10.81_{\pm 0.43}$ | $75.36_{\pm 0.74}$ | $26.29_{\pm 0.78}$ | $19.93_{\pm 0.05}$ |
| WALLA I (log-pooling) | $78.28_{\pm 0.65}$ | $81.60_{\pm 0.16}$ | $7.51_{\pm 0.64}$ | $72.51_{\pm 0.60}$ | $28.10_{\pm 0.33}$ | $16.07_{\pm 0.20}$ | $75.02_{\pm 0.35}$ | $78.83_{\pm 0.18}$ | $4.07_{\pm 0.21}$ | $75.18_{\pm 0.62}$ | $28.43_{\pm 0.72}$ | $20.26_{\pm 0.22}$ |
| WALLA I (calibrated) | $87.17_{\pm 0.25}$ | $86.18_{\pm 0.14}$ | $11.06_{\pm 0.18}$ | $78.31_{\pm 0.09}$ | $35.19_{\pm 0.41}$ | $12.11_{\pm 0.06}$ | $85.93_{\pm 0.23}$ | $87.19_{\pm 0.05}$ | $11.86_{\pm 0.13}$ | $81.41_{\pm 0.17}$ | $37.21_{\pm 0.44}$ | $11.82_{\pm 0.09}$ |

**Metrics.** We evaluate the area under the ROC curve (AUC) and accuracy (ACC)—and expected calibration error (ECE), and additional metrics to evaluate whether the aggregation weights align with per-question Brier-score performance: **Kendall's Tau**, the rank correlation between the ordering of models by their wagers and by their per-question Brier-score performance. **Mean reciprocal rank (MRR)**: Average inverse rank of the best-performing model under the ordering induced by aggregation weight. **Dynamic Regret (D-Regret)**: Per-question Brier-score gap between the aggregated prediction and the best individual model. More details can be found in Appendix F.4.

**Baselines.** We consider seven baselines spanning pre-inference routers (Chen et al., 2024; Song et al., 2025; Ong et al., 2025), learning-free ensembling (Mavromatis et al., 2024; Kang et al., 2025; Wang et al., 2022), and learning-based ensembling with LLM hidden states (Wolpert, 1992). Table 1 categorizes all methods along four axes: uncertainty-awareness, decentralized learning, learned weights, and incentive compatibility. WALLA simultaneously achieves all four properties. Detailed descriptions of each baseline are in Appendix F.5.

We provide the learning procedure of WALLA and other experiment setup details in Appendix F.1 and F.2, respectively.

### 4.2. Results and Discussions

We defer the results and discussions for scenarios I (homogeneous models with private contexts) and II (heterogeneous models) in Appendix F.7 and F.8, respectively. Discussion for robustness to variants, score–wager alignment, and model calibration is provided in Appendix F.9.

**Scenario III: heterogeneous models with private contexts.** We use all four LLMs for evaluation on PubMedQA. We consider two settings: (1) one randomly selected model receives the relevant context, as in Scenario I; and (2) one randomly selected model receives the relevant context while another randomly selected model receives an irrelevant context (an abstract randomly drawn from the test set). The results are provided in Table 2 and 13.

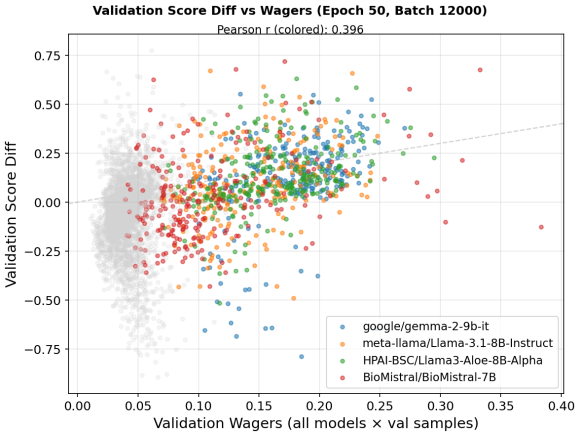


*Figure 1.* Score-wager alignment for WALLA I on Scenario III with heterogeneous models and private contexts. Colored dots highlight models with context. Pearson correlations are reported at the top of each panel. The plots for other settings are in Fig. 3.

StackedGen and WALLA stand out as the best-performing methods. RouteLLM (with context) and PackLLM are competitive when only one relevant context is present—context availability is easy to detect from input tokens—but their advantage diminishes when an irrelevant context is also introduced, as the method must now distinguish relevant from misleading context.

## 5. Conclusion

We propose a family of *advantage-aligned wagering mechanisms for LLM aggregation* (WALLA) that yields three properties (1) dominant-strategy incentive compatibility under general belief structures (2) score-wager alignment and (3) the decoupling of prediction and wager learning. To our knowledge, this is the first mechanism designed for aggregating LLM predictions that simultaneously achieves advantage-weighted aggregation, uncertainty-awareness, fully decentralized learning, and incentive-compatibility. Empirical results suggest that WALLA matches with centralized baselines in aggregated prediction performance.

## References


Abernethy, J. and Frongillo, R. M. A collaborative mechanism for crowdsourcing prediction problems. In *Advances in Neural Information Processing Systems*, 2011.

Abernethy, J., Chen, Y., and Vaughan, J. W. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation*, 1(2), 2013.

Arora, S., Hazan, E., and Kale, S. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012. doi: 10.4086/toc.2012.v008a006.

Azaria, A. and Mitchell, T. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Bergemann, D., Bojko, M., Duetting, P., Paes Leme, R., Xu, H., and Zuo, S. Data-driven mechanism design: Jointly eliciting preferences and information. In *Proceedings of the 26th ACM Conference on Economics and Computation*, pp. 507, 2025.

Beygelzimer, A., Langford, J., and Pennock, D. M. Learning performance of prediction markets with kelly bettors. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, pp. 1317–1318, 2012.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

Chen, S., Jiang, W., Lin, B., Kwok, J., and Zhang, Y. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Chen, Y., Devanur, N. R., Pennock, D. M., and Vaughan, J. W. Removing arbitrage from wagering mechanisms. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, pp. 377–394, 2014.

Chen, Z., Li, J., Chen, P., Li, Z., Sun, K., Luo, Y., Mao, Q., Li, M., Xiao, L., Yang, D., Ban, Y., Sun, H., and Yu, P. S. Harnessing multiple large language models: A survey on LLM ensemble. *arXiv preprint arXiv:2502.18036*, 2025.

Choi, H. K., Zhu, J., and Li, S. Debate or vote: Which yields better decisions in multi-agent large language models? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Dai, H., Teehan, R., and Ren, M. Are LLMs prescient? a continuous evaluation using daily news as the oracle. In *Forty-second International Conference on Machine Learning*, 2025.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Dubey, A., Feng, Z., Kidambi, R., Mehta, A., and Wang, D. Auctions with llm summaries. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 713–722, 2024.

Dütting, P., Mirrokni, V., Paes Leme, R., Xu, H., and Zuo, S. Mechanism design for large language models. In *Proceedings of the ACM Web Conference 2024*, pp. 144–155, 2024.

et al., A. G. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024a.

et al., G. T. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024b.

et al., Z. Z. Futurex: An advanced live benchmark for LLM agents in future prediction. *arXiv preprint arXiv:2508.11987*, 2025.

Freeman, R., Pennock, D. M., Podimata, C., and Vaughan, J. W. No-regret and incentive-compatible online learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Frongillo, R., Della Penna, N., and Reid, M. D. Interpreting prediction markets: a stochastic approach. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Genest, C. A Characterization Theorem for Externally Bayesian Groups. *The Annals of Statistics*, 12(3):1100 – 1105, 1984.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Guan, Y., Peng, H., Wang, X., Hou, L., and Li, J. OpenEP: Open-ended future event prediction. *ACM Trans. Inf. Syst.*, 2026.

Gururajan, A. K., Lopez-Cuena, E., Bayarri-Planas, J., Tormos, A., Hinjos, D., Bernabeu-Perez, P., Arias-Duart, A., Martin-Torres, P. A., Urcelay-Ganzabal, L., Gonzalez-Mallo, M., Alvarez-Napagao, S., Ayguadé-Parra, E., and Garcia-Gasulla, U. C. D. Aloe: A family of fine-tuned open healthcare LLMs. *arXiv preprint arXiv:2405.01886*, 2024.

Halawi, D., Zhang, F., Yueh-Han, C., and Steinhardt, J. Approaching human-level forecasting with language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

He, P., Gao, J., and Chen, W. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. Bayesian model averaging: a tutorial. *Statistical science*, 14(4):382–417, 1999.

Hu, J. and Storkey, A. Multi-period trading prediction markets with connections to machine learning. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1773–1781, 2014.

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

Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Jin, W., Khanna, R., Kim, S., Lee, D.-H., Morstatter, F., Galstyan, A., and Ren, X. ForecastQA: A question answering challenge for event forecasting with temporal text data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4636–4650, 2021.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Dodds, Z. N., DasSarma, N., Tran-Johnson, E., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Kang, Z., Zhao, X., and Song, D. Scalable best-of-n selection for large language models via self-certainty. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Karger, E., Bastani, H., Yueh-Han, C., Jacobs, Z., Halawi, D., Zhang, F., and Tetlock, P. Forecastbench: A dynamic benchmark of AI forecasting capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025.

Labrak, Y., Bazoge, A., Morin, E., Gourraud, P.-A., Rouvier, M., and Dufour, R. BioMistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5848–5864, 2024.

Lambert, N. S., Pennock, D. M., and Shoham, Y. Eliciting truthful answers to multiple-choice questions. pp. 109–118, 2008.

Lambert, N. S., Langford, J., Wortman Vaughan, J., Chen, Y., Reeves, D. M., Shoham, Y., and Pennock, D. M. An axiomatic characterization of wagering mechanisms. *Journal of Economic Theory*, 156:389–416, 2015.

Li, J., Zhang, Q., Yu, Y., Fu, Q., and Ye, D. More agents is all you need. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Lu, K., Yuan, H., Zheng, R., Yin, J., Zhao, X., Huang, X., and Qiu, X. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*, 2023.

Mavromatis, C., Karypis, P., and Karypis, G. Pack of LLMs: Model fusion at test-time via perplexity optimization. In *First Conference on Language Modeling*, 2024.

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., and Stoica, I. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025.

Pal, A., Umapathi, L. K., and Sankarasubbu, M. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Flores, G., Chen, G. H., Pollard, T., Ho, J. C., and Naumann, T. (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174, pp. 248–260, 2022.

Raja, A. A., Pinson, P., Kazempour, J., and Grammatico, S. A market for trading forecasts: A wagering mechanism. *International Journal of Forecasting*, 40(1):142–159, 2024.

Shnitzer, T., Ou, A., Silva, M., Soule, K., Sun, Y., Solomon, J., Thompson, N., and Yurochkin, M. Large language model routing with benchmark datasets. In *First Conference on Language Modeling*, 2024.

Song, W., Huang, Z., Cheng, C., Gao, W., Xu, B., Zhao, G., Wang, F., and Wu, R. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Soumalias, E., Curry, M., and Seuken, S. Truthful aggregation of LLMs with an application to online advertising. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Wang, F., Liu, Q., Chen, E., Huang, Z., Chen, Y., Yin, Y., Huang, Z., and Wang, S. Neural cognitive diagnosis for intelligent education systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6153–6161, Apr. 2020.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Wang, Z., Zhou, X., Yang, Y., Ma, B., Wang, L., Dong, R., and Anwar, A. OpenForecast: A large-scale open-ended event forecasting dataset. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5273–5294, 2025.

Wolpert, D. H. Stacked generalization. *Neural Networks*, 5 (2):241–259, 1992.

Xiao, Z., Dou, D., Xiong, B., Chen, Y., and Chen, G. Enhancing uncertainty estimation in LLMs with expectation of aggregated internal belief. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(40):34043–34051, Mar. 2026.

Xie, J., Chen, A. S., Lee, Y., Mitchell, E., and Finn, C. Calibrating language models with adaptive temperature scaling. In *EMNLP*, 2024.

Yang, Q., Mahns, S., Li, S., Gu, A., Wu, J., and Xu, H. LLM-as-a-prophet: Understanding predictive intelligence with prophet arena. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yu, Y.-C., Kuo, C. C., Ziqi, Y., Yucheng, C., and Li, Y.-S. Breaking the ceiling of the LLM community by treating token generation as a classification for ensembling. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1826–1839, 2024.

Zhang, Z., Cao, Y., Ye, C., Ma, Y., Liao, L., and Chua, T.-S. Analyzing temporal complex events with large language models? a benchmark towards temporal, long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1588–1606, 2024.

Zhu, C., Chen, N., Gao, Y., Zhang, Y., Tiwari, P., and Wang, B. Is your LLM outdated? a deep look at temporal generalization. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7433–7457, 2025.

Zou, A., Xiao, T., Jia, R., Kwon, J., Mazeika, M., Li, R., Song, D., Steinhardt, J., Evans, O., and Hendrycks, D. Forecasting future world events with neural networks. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

## A. Related Work

A growing body of work studies LLM aggregation. Equal-weighting approaches (Wang et al., 2022; Du et al., 2023; Choi et al., 2025) implicitly assume similar capabilities across participants. Training-free methods weight models by self-reported confidence (Kang et al., 2025) or input perplexity (Mavromatis et al., 2024), but these heuristics can be unreliable when models are miscalibrated and are susceptible to manipulation. Learning-based methods, including pre-inference routers (Lu et al., 2023; Chen et al., 2024; Ong et al., 2025; Song et al., 2025) and stacked generalization (Wolpert, 1992), can learn more effective domain-dependent weights, but often require central access to models' outputs or hidden states, and may require retraining as the model pool evolves.

Overall, existing methods offer useful trade-offs but have yet to simultaneously account for heterogeneity in model competence, support decentralized deployment, and ensure robustness to strategic behavior.

**Aggregation of LLM predictions.** The aggregation or ensembling of LLMs has been studied extensively (Chen et al., 2025). Equal-weighted aggregation, such as self-consistency (Wang et al., 2022), majority vote (Li et al., 2024; Yu et al., 2024; Choi et al., 2025), or multi-agent debate (Du et al., 2023) is shown to be effective. However, the underlying assumption that models have similar capabilities for solving the task may be flawed when the task requires knowledge that is familiar to only specialized LLMs. As it has been shown that LLMs' internal states encode their uncertainty or confidence (Kadavath et al., 2022; Azaria & Mitchell, 2023; Xiao et al., 2026), several training-free methods evaluate model uncertainties to compute the aggregation weights (Mavromatis et al., 2024; Kang et al., 2025). While efficient, these methods assume the uncertainty estimation can be faithful for all models across diverse tasks that require specialized knowledge, which may not be realistic. More general approaches consider ensembling heterogeneous LLMs via supervised learning. A stream of work considers pre-inference ensembling, learning a neural network (usually a small language model such as BERT) to route a task to a specific model without inferring to the model weights (Lu et al., 2023; Shnitzer et al., 2024; Chen et al., 2024; Ong et al., 2025; Song et al., 2025). Alternatively, stacked generalization (Wolpert, 1992) is a traditional ensembling method that uses models' hidden state as input and optimizes for the aggregation weights. Finally, many methods ensemble the models based on the evaluation of the output tokens by external reward models (Uesato et al., 2022; Jiang et al., 2023; Lightman et al., 2023). However, the costs for generating and evaluating long responses and for learning the reward models can be much higher than those of other aggregation methods. A series of works proposes mechanism designs for LLM aggregation (Dütting et al., 2024; Dubey et al., 2024; Bergemann et al., 2025; Soumalias et al., 2026). Rather than improving the aggregated prediction performance, their mechanisms aim to elicit truthful reports of LLM preferences in the form of bids, with application to online advertising.

**Wagering mechanisms and prediction markets.** We draw on the literature on wagering mechanisms, which provide incentive-compatible methods for eliciting and aggregating probabilistic beliefs from multiple agents (Lambert et al., 2008; Chen et al., 2014; Lambert et al., 2015; Freeman et al., 2020; Raja et al., 2024). Although originally developed in the context of forecast elicitation with monetary incentives, these mechanisms offer a natural foundation for decentralized aggregation of multiple LLMs. A closely related line of work studies prediction markets (Abernethy & Frongillo, 2011; Beygelzimer et al., 2012; Frongillo et al., 2012; Abernethy et al., 2013; Hu & Storkey, 2014), which aggregate beliefs through market prices, typically in dynamic environments where participants can update their positions over time. In contrast, we consider a static information setting.

**LLM for forecasting.** Several works evaluate LLMs' temporal generalization capability via forecasting tasks (Zhu et al., 2025; Dai et al., 2025) or study the adoption of language models to enhance the forecasting capabilities of AI systems (Zou et al., 2022; Halawi et al., 2024). As there is a growing interest in evaluating LLMs on forecasting tasks, numerous benchmarks have been designed (Jin et al., 2021; Zhang et al., 2024; Karger et al., 2025; Wang et al., 2025; et al., 2025; Guan et al., 2026; Yang et al., 2026). The recent benchmark (Yang et al., 2026) shows that state-of-the-art proprietary LLMs are already capable forecasters as they achieve low calibration errors and promising market returns on events from real-world prediction markets.

## B. Extended Preliminaries

This section introduces preliminaries underlying our proposed mechanism and analysis.

**Scoring rules and wagering mechanisms.** We consider a probabilistic prediction setting in which a center elicits predictions on uncertain outcomes from multiple agents and evaluates them based on the quality of their predictions. Let $\mathcal{M} = \{1, 2, \ldots, M\}$ denote a set of $M$ agents. A *wagering mechanism* operates in two stages. In the first stage, given a question $x \in \mathcal{X}$ with unknown outcome $y \in \mathcal{Y}$, each agent $i$ submits a prediction $p_i \in \Delta(\mathcal{Y})$ and wagers some non-negative amount $w_i \in \mathbb{R}_{\geq 0}$ on her prediction. We denote the resulting *prediction* and *wager profiles* by $\mathbf{p} := (p_1, \ldots, p_M)$ and $\mathbf{w} := (w_1, \ldots, w_M)$, and write $\mathbf{p}_{-i}$ and $\mathbf{w}_{-i}$ for the corresponding profiles excluding agent $i$. In the second stage, after the outcome is revealed, the mechanism determines the *net payoff* to each agent, denoted $\pi_i(\mathbf{p}, \mathbf{w}, y) \in \mathbb{R}$, based on the quality of her prediction and the amount wagered.

The quality of a prediction is measured by *a scoring rule*: $s\colon \Delta(\mathcal{Y}) \times \mathcal{Y} \to \mathbb{R}$, which assigns a numerical score $s(p, y)$ to a probabilistic forecast $p$ upon realization of outcome $y$. A *strictly proper scoring rule*, as defined, incentivizes a *risk-neutral* agent to predict according to her true belief.

**Definition B.1** (Strictly Proper Scoring Rule)**.** A scoring rule $s$ is *strictly proper* if and only if for all $p, q \in \Delta(\mathcal{Y})$ and $p \neq q$,

$$\mathbb{E}_{Y\sim q}[s(q, Y)] > \mathbb{E}_{Y\sim q}[s(p, Y)].$$

A forecast of $p = q$ is *truthful*, as the agent simply reports her belief $q$.

*Example* B.1 (Brier Score (Brier, 1950; Gneiting & Raftery, 2007)). $s(p, y) = c_1 - c_2\|p - \delta_y\|_2^2$, where $c_1 \in \mathbb{R}$, $c_2 > 0$, and $\delta_y \in \Delta(\mathcal{Y})$ denotes the Dirac measure at $y$ (i.e., the indicator vector of the realized outcome $y$).

The Brier score, also known as the quadratic score, is a simple and widely used strictly proper scoring rule; it is bounded and has the property that expected score is monotonically decreasing in $\|p - q\|_2^2$.

A scoring rule is then used in the mechanism's payout function $\pi : (\Delta(\mathcal{Y}) \times \mathbb{R}_{\geq 0})^M \times \mathcal{Y} \to \mathbb{R}^M$.

**Definition B.2** (Weighted-Score Wagering Mechanism (WSWM) (Lambert et al., 2015))**.** Given a scoring rule $s$, the WSWM specifies the net payoff of agent $i$ as the difference between her score and the wager-weighted average of all agents' scores, scaled by her own wager, i.e.,

$$\pi_i^{\text{WSWM}}(\mathbf{p}, \mathbf{w}, y) = w_i \left( s(p_i, y) - \frac{\sum_{j=1}^{M} w_j s(p_j, y)}{\sum_{j=1}^{M} w_j} \right). \tag{6}$$

The WSWM based on a Brier score is also referred to as the *Brier betting* mechanism.

**Desirable properties of wagering mechanisms.** Following Lambert et al. (2015) and Chen et al. (2014), we introduce a set of properties that wagering mechanism designers typically aim to satisfy and that are particularly relevant in our setting: (a) **Incentive compatibility (IC)**: Every agent maximizes her expected net payoff by reporting her true belief. The mechanism is *dominant-strategy incentive compatible (DSIC)* if this holds regardless of other agents' reports. (b) **Individual rationality (IR)**: Every agent prefers participating in the mechanism to not participating (i.e., the expected payoff of participating is nonnegative). (c) **Normality**: If from agent $i$'s perspective the prediction of another agent $j$ improves, then agent $i$'s expected new payoff decreases. (d) **No arbitrage**: No agent can make a profit risklessly for all prediction and wager profiles. We leave the formal mathematical condition for each to Appendix C.

WSWM is well-suited for human forecasters, satisfying the above properties except for no-arbitrage, but can fall short for learning-based agents. First, its linear dependence on $w_i$ necessitates hard wager caps $[w^L, w^H]$, leading rational agents to concentrate at the boundaries depending on whether they outperform the pool average, which yields only coarse, binary feedback. Second, its IC relies on immutable beliefs (Lambert et al., 2015); when agents update beliefs after observing others' reports, truthfulness may no longer be optimal. We aim to close both gaps by designing a mechanism suitable for both human and LLM participants: the best-response wager reflects score advantage and DSIC holds under general belief structure.

**Warm-up: WSWM with non-strategic agents.** We analyze WSWM with a scaled linear scoring rule under non-strategic agents who reinvest their cumulative payouts as wagers. We show the resulting dynamics are equivalent to multiplicative weights updates and recover Bayesian model averaging, with tighter regret bounds than standard Brier betting. We defer details to Appendix D.

## C. Desirable Properties of Wagering Mechanisms

**Definition C.1** (Incentive Compatibility (IC))**.** A wagering mechanism is incentive compatible if all players maximize their expected payout when predicting their true beliefs. Formally, $\forall i \in \mathcal{M}$, $q_i \in \Delta(\mathcal{Y})$ which reflects agent $i$'s true belief, $p_i \in \Delta(\mathcal{Y})$ where $q_i \neq p_i$, $\mathbf{p}_{-i} \in \Delta(\mathcal{Y})^{M-1}$, and $\mathbf{w} \in \mathbb{R}^M_{\geq 0}$,

$$\mathbb{E}_{Y\sim q_i}[\pi_i((q_i, \mathbf{p}_{-i}), \mathbf{w}, Y)] > \mathbb{E}_{Y\sim q_i}[\pi_i((p_i, \mathbf{p}_{-i}), \mathbf{w}, Y)].$$

**Definition C.2** (Individual Rationality)**.** Every agent prefers participating to not participating in the mechanism. Formally, $\forall i \in \mathcal{M}$, $q_i \in \Delta(\mathcal{Y})$ which reflects agent $i$'s true belief, and $w_i \in \mathbb{R}_{\geq 0}$, there exists some $p_i \in \Delta(\mathcal{Y})$ such that for all $\mathbf{p}_{-i} \in \Delta(\mathcal{Y})^{M-1}$ and $\mathbf{w}_{-i} \in \mathbb{R}^{M-1}_{\geq 0}$,

$$\mathbb{E}_{Y\sim q_i}[\pi_i((p_i, \mathbf{p}_{-i}), (w_i, \mathbf{w}_{-i}), Y)] \geq 0.$$

**Definition C.3** (Normality (Chen et al., 2014))**.** A mechanism satisfies normality if, for every $i \neq j \in \mathcal{M}$ , any belief $\mathcal{Q}_i$ over the measurable space $(\Omega, \mathcal{F})$, $\mathcal{F}_i \in \mathcal{F}$, $\mathbf{w} \in \mathbb{R}^M_{\geq 0}$ and $\mathbf{p} \in \Delta(\mathcal{Y})^M$, defining $\mathbf{p}'$ by $p'_j = p^*_i := \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i)$ and $p'_k = p_k$ for $k \neq j$,

$$\mathbb{E}_{\mathcal{Q}_i}[\pi_i(\mathbf{p}, \mathbf{w}, Y) \mid \mathcal{F}_i] \;\geq\; \mathbb{E}_{\mathcal{Q}_i}[\pi_i(\mathbf{p}', \mathbf{w}, Y) \mid \mathcal{F}_i],$$

with strict inequality whenever $p_j \neq \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i), w_i > 0,$ and $w_j > 0$. That is, if from agent $i$'s perspective (i.e., $\mathcal{Q}_i$), the prediction of another agent $j$ improves, then agent $i$'s expected payoff does not increase.

**Definition C.4** (No-Arbitrage (Chen et al., 2014))**.** A mechanism is arbitrage-free if there exist no agent $i \in \mathcal{M}$, prediction profile $\mathbf{p} \in \Delta(\mathcal{Y})^M$, and $\mathbf{w} \in \mathbb{R}^M_{\geq 0}$ such that $\pi_i(\mathbf{p}, \mathbf{w}, y) > 0$ for all $y \in \mathcal{Y}$. No agent can guarantee a strictly positive payoff regardless of the realized outcome.

**Definition C.5** (Weak Budget Balance)**.** The mechanism does not lose money but can potentially make a profit. Formally, $\forall \mathbf{p} \in \Delta(\mathcal{Y})^M, \mathbf{w} \in \mathbb{R}^M_{\geq 0}$ and $y \in \mathcal{Y}$,

$$\sum_{i=1}^{M} \pi_i(\mathbf{p}, \mathbf{w}, y) \geq 0.$$

## D. Theoretical Properties of WSWM Using a Scaled Linear Score

Suppose all participants are non-strategic: reporting the predictions $p_i^{(t)}$ truthfully at round $t$ and wagering the cumulative payout from the previous round, or namely, their budget (denoted as $B_i^{(t-1)}$, and $w_i^{(t)} = B_i^{(t-1)}$). Consider a WSWM that adopts a scaled linear score $s_{\text{lin}} \in \mathcal{S} : (\Delta(\mathcal{Y}) \times \mathbb{R}_{\geq 0})^M$ such that

$$s_i^{(t)} = \frac{p_i^{(t)}}{\sum_{j=1}^{M} w_j^{(t)} p_j^{(t)}}$$

and the linear pooling (i.e., $p_{\text{agg}}^{(t)} = \frac{\sum_{i=1}^{M} w_i^{(t)} p_i^{(t)}}{\sum_{i=1}^{M} w_i^{(t)}}$). We first demonstrate that such a mechanism inherently executes an exponentiated gradient descent step on the cross-entropy loss. We then show the mechanism exactly recovers Bayesian Model Averaging (BMA) (Hoeting et al., 1999). Finally, we derive a tighter regret bound than the one shown in Freeman et al. (2020), which highlights a tradeoff between this tight bound and incentive compatibility.

For notation simplicity, we remove the timestamp and simplify $p_{\text{agg}}[y]$ as $p_{\text{agg}}$ and $p_i[y]$ as $p_i$ when the context is clear.

### D.1. WSWM Recovers the Scaled Negative Gradient of Cross-Entropy Loss

We connect this mechanism to centralized supervised learning by analyzing the cross-entropy loss of the aggregated prediction.

**Theorem D.1.** *Suppose WSWM utilizes the scoring function $s_{lin}$. The net payout function, defined as $\pi_i = w_i \left( s_i - \frac{\sum_{k=1}^{M} s_k w_k}{\sum_{k=1}^{M} w_k} \right)$, satisfies $\pi_i = -w_i \frac{\partial L}{\partial w_i}$ where $L = -\log p_{agg}$, the cross-entropy loss.*

*Proof.* Expanding the logarithm of the quotient yields the loss function:

$$L = -\log \left( \sum_{i=1}^{M} w_i p_i \right) + \log \left( \sum_{i=1}^{M} w_i \right).$$

Taking the partial derivative of $L$ with respect to a specific weight $w_i$ gives:

$$\frac{\partial L}{\partial w_i} = -\frac{p_i}{\sum_{k=1}^{M} w_k p_k} + \frac{1}{\sum_{k=1}^{M} w_k}.$$

Substituting the given scoring rule $s_k$ into $\sum_{k=1}^{M} s_k w_k$ gives:

$$\begin{aligned}\sum_{k=1}^{M} s_k w_k &= \sum_{k=1}^{M} \left( \frac{p_k}{\sum_{j=1}^{M} w_j p_j} \right) w_k \\ &= \frac{\sum_{k=1}^{M} w_k p_k}{\sum_{j=1}^{M} w_j p_j} \\ &= 1.\end{aligned}$$

We substitute this result back into the net payout function $\pi_i$:

$$\begin{aligned}\pi_i &= w_i \left( s_i - \frac{1}{\sum_{k=1}^{M} w_k} \right) \\ &= w_i \left( \frac{p_i}{\sum_{j=1}^{M} w_j p_j} - \frac{1}{\sum_{k=1}^{M} w_k} \right).\end{aligned}$$

Factoring out $-1$ from the terms inside the parentheses, we get:

$$\pi_i = -w_i \left( -\frac{p_i}{\sum_{j=1}^{M} w_j p_j} + \frac{1}{\sum_{k=1}^{M} w_k} \right).$$

We observe that the expression inside the parentheses is exactly the partial derivative $\frac{\partial L}{\partial w_i}$ derived previously. Therefore, by direct substitution:

$$\pi_i = -w_i \frac{\partial L}{\partial w_i}.$$

□

*Remark* D.1. This relationship demonstrates that the budget update $B_i^{(t)} = w_i^{(t)} + \pi_i^{(t)} = w_i^{(t)} \left( 1 - \frac{\partial L^{(t)}}{\partial w_i^{(t)}} \right) = w_i^{(t)} - w_i^{(t)} \frac{\partial L^{(t)}}{\partial w_i^{(t)}}$ has the form of a Multiplicative Weights Update (MWU) algorithm (Arora et al., 2012). It can be interpreted as executing decentralized gradient descent where the learning rate is scaled dynamically by the current weight. It also mirrors the exponentiated gradient step $B_i^{(t)} \approx w_i^{(t)} e^{-\frac{\partial L^{(t)}}{\partial w_i^{(t)}}}$ under a first-order Taylor approximation.

### D.2. Equivalence to Bayesian Model Averaging

We now show that the WSWM under this setting is equivalent to Bayesian model averaging (BMA) (Hoeting et al., 1999). We start with the review of BMA.

**Definition D.2** (Bayesian Model Averaging (BMA))**.** Let $\theta_1, \ldots, \theta_M$ be discrete parameters for the $M$ models. Given a history of observations $D^{(1:t-1)} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(t-1)}, y^{(t-1)})\}$, the posterior probability of parameter $\theta_i$ is $p(\theta_i | D^{(1:t-1)})$. Assuming $(x^{(t)}, y^{(t)})$ are drawn i.i.d., the BMA predictive probability for the true label $y^{(t)}$ is:

$$p_{\text{BMA}}(y^{(t)} | x^{(t)}, D^{(1:t-1)}) = \sum_{i=1}^{M} p(y^{(t)} | x^{(t)}, \theta_i) p(\theta_i | D^{(1:t-1)}).$$

To prove that the mechanism recovers the exact posterior, we assume that the BMA predictive distribution perfectly recovers the true marginal probability of the label (i.e., $p_{\text{BMA}}(y^{(t)} | x^{(t)}, D^{(1:t-1)}) = p(y^{(t)} | x^{(t)}, D^{(1:t-1)})$.

**Theorem D.3.** *Assume the BMA predictive distribution equals the true marginal data distribution: $p_{BMA}(y^{(t)}|x^{(t)}, D^{(1:t-1)}) = p(y^{(t)}|x^{(t)}, D^{(1:t-1)})$. If the initial budgets are set to the prior probabilities $B_i^{(0)} = p(\theta_i)$, the aggregated prediction satisfies $p_{agg}^{(t)} = p_{BMA}(y^{(t)}|x^{(t)}, D^{(1:t-1)})$ for all $t \geq 1$, and the budgets track the exact posterior $B_i^{(t)} = p(\theta_i|D^{(1:t)})$.*

*Proof.* We proceed by induction. The base case holds by initialization $B_i^{(0)} = p(\theta_i)$. Assume the inductive hypothesis $B_i^{(t-1)} = p(\theta_i|D^{(1:t-1)})$. Because $w_i^{(t)} = B_i^{(t-1)}$, substituting the hypothesis into the aggregated prediction yields $p_{\text{agg}}^{(t)} = \sum_{j=1}^{M} p(\theta_j|D^{(1:t-1)})p(y^{(t)}|x^{(t)}, \theta_j)$, which exactly matches Definition D.2.

Evaluating the budget update step established previously:

$$
\begin{aligned}
B_i^{(t)} &= w_i^{(t)} s_i^{(t)} \\
&= w_i^{(t)} \frac{p_i^{(t)}}{\sum_{j=1}^{M} w_j^{(t)} p_j^{(t)}} \\
&= B_i^{(t-1)} \frac{p_i^{(t)}}{p_{\text{agg}}^{(t)}} \\
&= \frac{p(\theta_i|D^{(1:t-1)})p(y^{(t)}|x^{(t)}, \theta_i)}{p_{\text{BMA}}(y^{(t)}|x^{(t)}, D^{(1:t-1)})} \\
&= \frac{p(\theta_i|x^{(t)}, D^{(1:t-1)})p(y^{(t)}|x^{(t)}, D^{(1:t-1)}, \theta_i)}{p_{\text{BMA}}(y^{(t)}|x^{(t)}, D^{(1:t-1)})} \quad \text{(conditional independence)} \\
&= \frac{p(\theta_i|x^{(t)}, D^{(1:t-1)})p(y^{(t)}|x^{(t)}, D^{(1:t-1)}, \theta_i)}{p(y^{(t)}|x^{(t)}, D^{(1:t-1)})} \quad \text{(by assumption)} \\
&= p(\theta_i|y^{(t)}, x^{(t)}, D^{(1:t-1)}) \quad \text{(Bayes' theorem)} \\
&= p(\theta_i|D^{(1:t)}).
\end{aligned}
$$

This completes the proof. □

As $p_{\text{BMA}}$ cannot recover the true marginal exactly, its effectiveness is typically quantified with the notion of regret. We provide the regret bound in the next section.

### D.3. Regret Bound

We continue to assume the initial budgets sum to one and derive a regret bound for the WSWM with non-strategic agents under $s_{lin}$ (i.e., $s_i^{(t)} = \frac{p_i^{(t)}}{\sum_{j=1}^{M} w_j^{(t)} p_j^{(t)}}$).

**Theorem D.4** (Regret Bound)**.** *Assume $p_i^{(t)} > 0$ for all $i, t$. The regret of the WSWM mechanism over $T$ rounds with respect to any optimal fixed distribution of models $\mathbf{u} \in \Delta(\mathcal{M})$ is bounded by:*

$$
\sum_{t=1}^{T} L^{(t)}(\mathbf{w}^{(t)}) - \sum_{t=1}^{T} L^{(t)}(\mathbf{u}) \leq D_{KL}(\mathbf{u} \parallel \mathbf{w}^{(0)}) + H(\mathbf{u}),
$$

*where $D_{KL}$ is the Kullback-Leibler divergence and $H(\mathbf{u})$ is the Shannon entropy.*

*Proof.* Because $\sum_{k=1}^{M} w_k^{(t)} = 1$ and $\sum_{k=1}^{M} s_k^{(t)} w_k^{(t)} = \frac{\sum_{k=1}^{M} w_k^{(t)} p_k^{(t)}}{\sum_{j=1}^{M} w_j^{(t)} p_j^{(t)}} = 1$, the net payout simplifies to:

$$
\pi_i^{(t)} = w_i^{(t)}(s_i^{(t)} - 1).
$$

Let $Z_t = \sum_{j=1}^{M} w_j^{(t)} p_j^{(t)}$. A player's updated budget for the next round is their wager plus their net payout: $w_i^{(t+1)} = B_i^{(t)} = w_i^{(t)} + \pi_i^{(t)} = w_i^{(t)} s_i^{(t)} = \frac{w_i^{(t)} p_i^{(t)}}{Z_t}$.

The instantaneous loss is $L^{(t)}(\mathbf{w}^{(t)}) = -\log Z_t$. We expand the terminal weight $w_i^{(T+1)}$ recursively:

$$w_i^{(T+1)} = w_i^{(0)} \prod_{t=1}^{T} \frac{p_i^{(t)}}{Z_t}.$$

Taking the natural logarithm of both sides:

$$\log w_i^{(T+1)} = \log w_i^{(0)} + \sum_{t=1}^{T} \log p_i^{(t)} - \sum_{t=1}^{T} \log Z_t.$$

Define the loss for a fixed model $i$ as $L^{(t)}(\mathbf{e}_i) = -\log p_i^{(t)}$. Substituting this and $-\log Z_t = L^{(t)}(\mathbf{w}^{(t)})$ yields:

$$\log w_i^{(T+1)} = \log w_i^{(0)} - \sum_{t=1}^{T} L^{(t)}(\mathbf{e}_i) + \sum_{t=1}^{T} L^{(t)}(\mathbf{w}^{(t)}).$$

Because $\mathbf{w}^{(T+1)} \in \Delta(\mathcal{M})$, we know $\log w_i^{(T+1)} \leq 0$. Rearranging gives:

$$\sum_{t=1}^{T} L^{(t)}(\mathbf{w}^{(t)}) - \sum_{t=1}^{T} L^{(t)}(\mathbf{e}_i) \leq -\log w_i^{(0)}.$$

Taking the expectation over $i \sim \mathbf{u}$:

$$\sum_{t=1}^{T} L^{(t)}(\mathbf{w}^{(t)}) - \sum_{i=1}^{M} u_i \sum_{t=1}^{T} L^{(t)}(\mathbf{e}_i) \leq -\sum_{i=1}^{M} u_i \log w_i^{(0)}.$$

By Jensen's inequality, $-\sum_{i=1}^{M} u_i \log p_i^{(t)} \geq -\log\left(\sum_{i=1}^{M} u_i p_i^{(t)}\right) = L^{(t)}(\mathbf{u})$. Substituting this lower bound and isolating the information-theoretic quantities produces:

$$\sum_{t=1}^{T} L^{(t)}(\mathbf{w}^{(t)}) - \sum_{t=1}^{T} L^{(t)}(\mathbf{u}) \leq \sum_{i=1}^{M} u_i \log \frac{u_i}{w_i^{(0)}} - \sum_{i=1}^{M} u_i \log u_i,$$

which simplifies to $D_{\mathrm{KL}}(\mathbf{u} \parallel \mathbf{w}^{(0)}) + H(\mathbf{u})$. □

*Remark* D.2. The constant regret bound $D_{\mathrm{KL}}(\mathbf{u} \parallel \mathbf{w}^{(0)}) + H(\mathbf{u})$ achieved by WSWM is tighter than the $\mathcal{O}(\sqrt{T \ln M})$ sublinear bound established by Freeman et al. (2020). This difference highlights a tradeoff between a tight regret bound and IC.[5]

# E. Deferred Content from Sec. 3

## E.1. Proof of DSIC of Prediction

**Theorem E.1** (DSIC of Prediction)**.** *Under the payout function (2), truthful prediction is a dominant strategy: each agent maximizes her expected payoff by reporting her own belief about $Y$, regardless of opponents' strategies and her own wager. Formally, for any belief $\mathcal{Q}_i$ held by agent $i$, any wager profile $\mathbf{w} \in \mathbb{R}^M_{\geq 0}$, and any $\mathbf{p}_{-i} \in \Delta(\mathcal{Y})^{M-1}$,*

$$p_i^* := \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i) \;\in\; \arg\max_{p_i \in \Delta(\mathcal{Y})} \mathbb{E}_{\mathcal{Q}_i}[\pi_i((p_i, \mathbf{p}_{-i}), \mathbf{w}, Y) \mid \mathcal{F}_i],$$

*and the maximizer is unique when $w_i > 0$.*

*Proof.* From the decomposition (**??**), since $s_{-i}$ does not depend on $p_i$, the optimization reduces to

$$p_i^* \in \arg\max_{p_i \in \Delta(\mathcal{Y})} \mathbb{E}_{\mathcal{Q}_i}[s(p_i, Y) \mid \mathcal{F}_i].$$

By strict properness of $s$, this is uniquely attained at $p_i^* := \mathbb{E}_{\mathcal{Q}_i}[\delta_Y \mid \mathcal{F}_i] = \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i)$, the agent's belief about $Y$ given her observation of the question $X$ and private signal $S_i$. When $w_i = 0$, the payout is identically zero regardless of the prediction. □

[5] Although Freeman et al. (2020) is incentive compatible to agents who strategically report predictions, they assume a fixed wagering strategy that always wagers a fraction $\eta$ of the cumulative payout.

### E.2. Proof of Score-Wager Alignment

**Theorem E.2** (Score-Wager Alignment)**.** *Under the payout function (2), the best-response wager is proportional to the agent's expected score advantage. Formally, for any belief $\mathcal{Q}_i$ held by agent $i$, any $\mathbf{p} \in \Delta(\mathcal{Y})^M$, and any $\mathbf{w}_{-i} \in \mathbb{R}^{M-1}_{\geq 0}$, the best-response wager for agent $i$ is*

$$w_i^* = \left(\frac{A_i}{2c_3}\right)^+, \qquad A_i := \mathbb{E}_{\mathcal{Q}_i}[s(p_i, Y) - s_{-i} \mid \mathcal{F}_i], \tag{7}$$

*where $(x)^+ := \max(x, 0)$ and $A_i$ is the agent $i$'s* expected score advantage *over the baseline.*

*Proof.* For any fixed $p_i$, $\mathbf{p}_{-i}$, and $\mathbf{w}_{-i}$, the expected payout (**??**) as a function of $w_i \geq 0$ is $\mathbb{E}[\pi_i \mid \mathcal{F}_i] = w_i\, A_i - c_3\, w_i^2$. Since $\frac{\partial^2}{\partial w_i^2}\mathbb{E}[\pi_i \mid \mathcal{F}_i] = -2c_3 < 0$, this is strictly concave. The unique unconstrained maximizer is $w_i = A_i/(2c_3)$, and the constrained maximizer over $w_i \geq 0$ is $w_i^* = (A_i/(2c_3))^+$. □

### E.3. Examples of Heterogeneous Predictions and Equilibrium Analyses

We use Theorems 3.2 and 3.3 to characterize equilibrium in each extreme case. In practice, both sources often appear together. See example

*Example* E.1 (Homogeneous agents with private signals). Consider $M$ homogeneous agents sharing a common prior $\mathcal{Q}$ over $(\Omega, \mathcal{F})$ (i.e., $\mathcal{Q}_i = \mathcal{Q}$ for all $i$.). Each agent $i$ observes a private signal $S_i$, giving rise to distinct private information $\mathcal{F}_i = \sigma(X, S_i) \neq \mathcal{F}_j$ when $S_i \neq S_j$. Agents differ not in their priors but in what they observe: heterogeneous predictions arise from heterogeneous information. In LLM settings, this corresponds to $M$ copies of the same base model, each receiving a different context for the same question.

**Corollary E.3** (BNE under common prior)**.** *Suppose agents share a common prior $\mathcal{Q}$ (Example E.1). Let $p_i^* = \mathcal{Q}(Y = \cdot \mid \mathcal{F}_i)$ (Theorem 3.2) and $w_i^* = (A_i/2c_3)^+$ with $A_i = \mathbb{E}[s(p_i^*, Y) - s_{-i} \mid \mathcal{F}_i]$ (Theorem 3.3). The strategy profile $\{(p_i^*, w_i^*)\}_{i=1}^M$ is a Bayes–Nash equilibrium: for every agent $i$ and every alternative strategy $(\tilde{p}_i, \tilde{w}_i)$,*

$$\mathbb{E}\big[\pi_i\big((p_i^*, \mathbf{p}_{-i}^*), (w_i^*, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\big] \;\geq\; \mathbb{E}\big[\pi_i\big((\tilde{p}_i, \mathbf{p}_{-i}^*), (\tilde{w}_i, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\big]\,.$$

*Example* E.2 (Heterogeneous agents). Consider $M$ agents holding beliefs $\mathcal{Q}_1, \ldots, \mathcal{Q}_M$ that may differ. No assumption is made on how the beliefs relate; they may be independent, correlated, or partially overlapping. Agent $i$ does not know agent $j$'s belief $\mathcal{Q}_j$. Heterogeneous predictions arise not from different information but from different perspectives on the same input. In the LLM setting, this corresponds to $M$ distinct models—differing in architecture, training data, or fine-tuning—each observing the same question with no additional context ($S_i = \emptyset$, so $\mathcal{F}_i = \sigma(X) = \mathcal{F}_j$).

**Corollary E.4** (NE under heterogeneous beliefs)**.** *Suppose each agent $i$ holds her own belief $\mathcal{Q}_i$ (Example E.2). Let $p_i^* = \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i)$ (Theorem 3.2) and $w_i^* = (A_i/2c_3)^+$ with $A_i = \mathbb{E}_{\mathcal{Q}_i}[s(p_i^*, Y) - s_{-i} \mid \mathcal{F}_i]$ (Theorem 3.3). The strategy profile $\{(p_i^*, w_i^*)\}_{i=1}^M$ is a Nash equilibrium: for every agent $i$ and every alternative strategy $(\tilde{p}_i, \tilde{w}_i)$,*

$$\mathbb{E}_{\mathcal{Q}_i}\big[\pi_i\big((p_i^*, \mathbf{p}_{-i}^*), (w_i^*, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\big] \;\geq\; \mathbb{E}_{\mathcal{Q}_i}\big[\pi_i\big((\tilde{p}_i, \mathbf{p}_{-i}^*), (\tilde{w}_i, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\big]\,.$$

### E.4. Proof of BNE under Common Prior

**Corollary E.5** (BNE under common prior)**.** *Suppose agents share a common prior $\mathcal{Q}$ (Example E.1). Let $p_i^* = \mathcal{Q}(Y = \cdot \mid \mathcal{F}_i)$ (Theorem 3.2) and $w_i^* = (A_i/2c_3)^+$ with $A_i = \mathbb{E}[s(p_i^*, Y) - s_{-i} \mid \mathcal{F}_i]$ (Theorem 3.3). The strategy profile $\{(p_i^*, w_i^*)\}_{i=1}^M$ is a Bayes–Nash equilibrium: for every agent $i$ and every alternative strategy $(\tilde{p}_i, \tilde{w}_i)$,*

$$\mathbb{E}\big[\pi_i\big((p_i^*, \mathbf{p}_{-i}^*), (w_i^*, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\big] \;\geq\; \mathbb{E}\big[\pi_i\big((\tilde{p}_i, \mathbf{p}_{-i}^*), (\tilde{w}_i, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\big]\,.$$

*Proof.* Fix opponents' equilibrium strategies $(\mathbf{p}_{-i}^*, \mathbf{w}_{-i}^*)$. By Theorem 3.2, $p_i^* = \mathcal{Q}(Y = \cdot \mid \mathcal{F}_i)$ uniquely maximizes $\mathbb{E}[s(p_i, Y) \mid \mathcal{F}_i]$ over all $p_i$, and hence maximizes $A_i = \mathbb{E}[s(p_i, Y) - s_{-i} \mid \mathcal{F}_i]$. Given $p_i^*$, Theorem 3.3 gives the best-response wager $w_i^* = (A_i/2c_3)^+$. When $A_i > 0$, the pair $(p_i^*, w_i^*)$ is the unique maximizer of the expected net payout. When $A_i \leq 0$, $w_i^* = 0$ and the net payout is zero regardless of $p_i$; the agent is excluded from aggregation. A fixed point of the wager best-response mapping exists by Brouwer's theorem for bounded $s$, as the mapping is continuous on $[0, (\bar{s} - \underline{s})/(2c_3)]^M$. □

### E.5. Proof of NE under Heterogenenous Beliefs

**Corollary E.6** (NE under heterogeneous beliefs)**.** *Suppose each agent $i$ holds her own belief $\mathcal{Q}_i$ (Example E.2). Let $p_i^* = \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i)$ (Theorem 3.2) and $w_i^* = (A_i/2c_3)^+$ with $A_i = \mathbb{E}_{\mathcal{Q}_i}[s(p_i^*, Y) - s_{-i} \mid \mathcal{F}_i]$ (Theorem 3.3). The strategy profile $\{(p_i^*, w_i^*)\}_{i=1}^M$ is a Nash equilibrium: for every agent $i$ and every alternative strategy $(\tilde{p}_i, \tilde{w}_i)$,*

$$\mathbb{E}_{\mathcal{Q}_i}\left[\pi_i\big((p_i^*, \mathbf{p}_{-i}^*), (w_i^*, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\right] \;\geq\; \mathbb{E}_{\mathcal{Q}_i}\left[\pi_i\big((\tilde{p}_i, \mathbf{p}_{-i}^*), (\tilde{w}_i, \mathbf{w}_{-i}^*), Y\big) \mid \mathcal{F}_i\right].$$

*Proof.* Fix opponents' equilibrium strategies $(\mathbf{p}_{-i}^*, \mathbf{w}_{-i}^*)$. By Theorem 3.2, $p_i^*$ maximizes $\mathbb{E}_{\mathcal{Q}_i}[s(p_i, Y) \mid \mathcal{F}_i]$ and hence maximizes $A_i(p_i) = \mathbb{E}_{\mathcal{Q}_i}[s(p_i, Y) - s_{-i} \mid \mathcal{F}_i]$ (since $s_{-i}$ depends on opponents' strategies, not on $p_i$). By Theorem 3.3, for any fixed $p_i$, the payoff is maximized at $w_i = (A_i(p_i)/2c_3)^+$. Since $p_i^*$ maximizes $A_i$, the pair $(p_i^*, w_i^*)$ maximizes the expected payoff over all $(\tilde{p}_i, \tilde{w}_i)$ jointly. A fixed point exists by Brouwer's theorem for bounded $s$. □

### E.6. Two Variants of Leave-One-Out Score Baseline

Theorems 3.2 and 3.3 hold for any leave-one-out baseline $s_{-i}$ that does not depend on $(p_i, w_i)$. We now instantiate two concrete choices that make different design tradeoffs.

**Definition E.7** (Variant I: Weighted-Score Baseline)**.** The baseline is the wager-weighted average of other agents' scores:

$$s_{-i}^{\mathrm{I}}(\mathbf{p}_{-i}, \mathbf{w}_{-i}, y) := \frac{\sum_{j \neq i} w_j \, s(p_j, y)}{\sum_{j \neq i} w_j}. \tag{8}$$

**Definition E.8** (Variant II: Weighted-Prediction Baseline)**.** The baseline evaluates the scoring rule at the wager-weighted pool of other agents' predictions:

$$s_{-i}^{\mathrm{II}}(\mathbf{p}_{-i}, \mathbf{w}_{-i}, y) := s(q_{-i}, y), \qquad q_{-i} := \frac{\sum_{j \neq i} w_j \, p_j}{\sum_{j \neq i} w_j} \in \Delta(\mathcal{Y}). \tag{9}$$

In short, Variant I averages scores, whereas Variant II scores an average prediction. The two variants trade off normality (Def. C.3) and no-arbitrage (Def. C.4). These two properties are in tension under known wagering mechanism designs. Lambert et al. (2015) showed that WSWM satisfies normality, while Chen et al. (2014) showed it admits arbitrage and proposed the no-arbitrage wagering mechanism (NAWM), which eliminates arbitrage but sacrifices normality. Our two variants exhibit the same tradeoff.

**Proposition E.9** (Normality of Variant I)**.** *Variant I satisfies normality.*

*Proof.* By strict properness, $p_i^* := \mathcal{Q}_i(Y = \cdot \mid \mathcal{F}_i)$ uniquely maximizes $\mathbb{E}[s(p, Y) \mid \mathcal{F}_i]$. Replacing $p_j$ with a prediction closer to $p_i^*$ increases $\mathbb{E}[s(p_j, Y) \mid \mathcal{F}_i]$, which raises the baseline $\mathbb{E}[s_{-i}^{\mathrm{I}} \mid \mathcal{F}_i]$ and hence reduces $A_i$ and the payoff. □

**Proposition E.10** (Arbitrage in Variant I)**.** *Variant I does not satisfy no-arbitrage. When predictions from other agents are not all identical, there exists a report $\hat{p}_i$ and a sufficiently small wager $w_i$ such that $\pi_i > 0$ for all outcomes $y \in \mathcal{Y}$.*

*Proof.* By Theorem 3.3 of Chen et al. (2014), when at least two predictions differ ($p_j \neq p_k$ for some $j, k \neq i$), there exists $\hat{p}_i$ with $s(\hat{p}_i, y) > s_{-i}^{\mathrm{I}}$ for all $y$. Let $\delta(y) := s(\hat{p}_i, y) - s_{-i}^{\mathrm{I}} > 0$ and $\delta_{\min} := \min_y \delta(y) > 0$. Then $\pi_i = w_i(\delta(y) - c_3 w_i)$. Choosing any $w_i \in (0, \, \delta_{\min}/c_3)$ ensures $c_3 w_i < \delta(y)$ for all $y$, hence $\pi_i > 0$ for all $y$. □

**Proposition E.11** (No-Arbitrage of Variant II)**.** *Variant II satisfies no-arbitrage.*

*Proof.* By Theorem 4.1 of Chen et al. (2014), the base payoff $\tilde{\pi}_i := w_i(s(p_i, y) - s(q_{-i}, y))$ has the form of a no-arbitrage wagering mechanism: there exists $y^*$ with $\tilde{\pi}_i(y^*) \leq 0$. Since $\pi_i = \tilde{\pi}_i - c_3 w_i^2 \leq \tilde{\pi}_i$, it follows that $\pi_i(y^*) \leq 0$. □

**Proposition E.12** (Normality Failure of Variant II)**.** *Variant II does not satisfy normality: by adopting $p_i^*$, agent $j$ can change the pooled baseline $q_{-i}$ in a direction that increases player $i$'s expected payoff.* ***Proof.*** *See our constructed counterexample in Appendix E.7.*

Both variants satisfy DSIC (Theorem 3.2) and score-wager alignment (Theorem 3.3), and the choice between variants is a design decision, depending on the presence of strategic behavior. The leave-one-out baseline in our payout function enables DSIC under any belief structure, score-wager alignment at best response, and prediction-wager decoupling, however, at the cost of budget balance. We show below that the resulting worst-case deficit of our mechanism is bounded and is independent of the number of participants $M$.

**Proposition E.13** (Bounded Worst-Case Deficit)**.** *Under the payout function (2) with a bounded scoring rule ($s \in [\underline{s}, \overline{s}]$) for Variant I, or a bounded and concave scoring rule for Variant II (e.g., Brier), the total payout satisfies*

$$\sum_{i=1}^{M} \pi_i \;\le\; \left(\frac{\overline{s}-\underline{s}}{W} - c_3\right) \sum_{i=1}^{M} w_i^2 \;\le\; \frac{(\overline{s}-\underline{s})^2}{4c_3} \tag{10}$$

*where $W = \sum_i w_i$, for any prediction profile $\mathbf{p} \in \Delta(\mathcal{Y})^M$, any wager profile $\mathbf{w} \in \mathbb{R}^M_{\ge 0}$, and realized outcome $y \in \mathcal{Y}$.*

*The worst-case deficit is $(\overline{s}-\underline{s})^2/(4c_3)$, independent of the number of agents $M$. For our payout function using the Brier score, the worst-case deficit is $c_2^2/c_3$.*

*Proof.* We first consider *Variant I*. Write $s_i := s(p_i, y)$, $\bar{s}_w := \frac{\sum_j w_j s_j}{W}$ and $W_{-i} = \sum_{j \ne i} w_j$. From $W\bar{s}_w = w_i s_i + W_{-i} s^{\mathrm{I}}_{-i}$, we have:

$$s_i - s^{\mathrm{I}}_{-i} = \frac{W(s_i - \bar{s}_w)}{W_{-i}}, \qquad s_i - \bar{s}_w = \frac{W_{-i}}{W}(s_i - s^{\mathrm{I}}_{-i}). \tag{11}$$

Writing $\frac{W}{W_{-i}} = 1 + \frac{w_i}{W_{-i}}$ and using $\sum_i w_i(s_i - \bar{s}_w) = 0$, we write our payout function (2) as the difference of the leave-one-out (LOO) inflation term and the regularization term:

$$\sum_{i=1}^{M} \pi_i = \underbrace{\sum_{i=1}^{M} \frac{w_i^2(s_i - \bar{s}_w)}{W_{-i}}}_{\text{LOO inflation}} - \underbrace{c_3 \sum_{i=1}^{M} w_i^2}_{\text{regularization}}. \tag{12}$$

Using (11) and $s_i - s^{\mathrm{I}}_{-i} \le \overline{s} - \underline{s}$, we have

$$\frac{w_i^2(s_i - \bar{s}_w)}{W_{-i}} = \frac{w_i^2(s_i - s^{\mathrm{I}}_{-i})}{W} \le \frac{w_i^2(\overline{s}-\underline{s})}{W}.$$

Therefore, the LOO inflation is at most $\frac{\overline{s}-\underline{s}}{W}\sum_i w_i^2$, giving the first inequality in (10).

For the second: since $\sum_i w_i^2/W \le w_{\max}$ and $\sum_i w_i^2 \ge w_{\max}^2$,

$$\frac{(\overline{s}-\underline{s})\sum_i w_i^2}{W} - c_3 \sum_i w_i^2 \;\le\; (\overline{s}-\underline{s})\, w_{\max} - c_3\, w_{\max}^2 \;\le\; \frac{(\overline{s}-\underline{s})^2}{4c_3},$$

where the last step maximized at $(\overline{s}-\underline{s})/(2c_3)$ with value $(\overline{s}-\underline{s})^2/(4c_3)$.

*Variant II.* When $s(\cdot, y)$ is concave in $p$, Jensen's inequality gives $s^{\mathrm{II}}_{-i} \ge s^{\mathrm{I}}_{-i}$ and thus $\pi^{\mathrm{II}}_{-i} \le \pi^{\mathrm{I}}_{-i}$ for all prediction and wager profiles, so the Variant I bound applies. □

Compare to WSWM with global average baseline (Lambert et al., 2015) which achieves exact budget balance, our leave-one-out baseline inflates each agent's deviation by $W/W_{-i}$, producing the LOO inflation term in (12). The regularization $c_3 \sum_i w_i^2$ serves to offset this inflation (e.g., by taxing wagers). The regularization constant $c_3$ is the designer's lever: larger $c_3$ reduces the worst-case deficit at the cost of suppressing wagers.

### E.7. Proof of Normality Failure of Variant II

**Proposition E.14** (Normality Failure of Variant II)**.** *Variant II does not satisfy normality in general (Def. C.3).*

*Proof.* Recall that normality is asserted for every *fixed* prediction profile $\mathbf{p} \in \Delta(\mathcal{Y})^M$ and wager profile $\mathbf{w} \in \mathbb{R}^M_{\geq 0}$; these are treated as deterministic parameters, and the conditional expectation $\mathbb{E}[\cdot \mid \mathcal{F}_i]$ is taken solely over the randomness in $Y$. Consequently, for any fixed $\mathbf{p}$ and $\mathbf{w}$, the pooled baseline $q_{-i} = \sum_{j \neq i} w_j p_j / W_{-i}$ is a fixed vector in $\Delta(\mathcal{Y})$, and it is valid to use it as a constant inside conditional expectations.

We construct a concrete counterexample with the Brier score $s(p, y) = c_1 - c_2 \|p - \delta_y\|_2^2$ (Def. B.1).

Let $\mathcal{Y} = \{1, 2, 3\}$, $M = 3$, and consider normality for player $i = 1$. Fix the following deterministic profiles:

$$p_1 = (0.5,\, 0.5,\, 0), \quad p_2 = (1,\, 0,\, 0), \quad p_3 = (0,\, 1,\, 0), \quad w_2 = 1, \quad w_3 = 2,$$

so $W_{-1} = 3$. The Variant II baseline for player 1 is

$$q_{-1} = \frac{w_2 p_2 + w_3 p_3}{W_{-1}} = \tfrac{1}{3}(1, 0, 0) + \tfrac{2}{3}(0, 1, 0) = \left(\tfrac{1}{3}, \tfrac{2}{3}, 0\right).$$

Since $q_{-1}$ is a fixed vector and $p_1 = \mathbb{E}[\delta_Y \mid \mathcal{F}_1]$ is $\mathcal{F}_1$-measurable, the bias–variance decomposition yields

$$\mathbb{E}\left[\|q_{-1} - \delta_Y\|_2^2 \mid \mathcal{F}_1\right] = \|q_{-1} - p_1\|_2^2 + \mathbb{E}\left[\|p_1 - \delta_Y\|_2^2 \mid \mathcal{F}_1\right],$$

where the cross-term $2\langle q_{-1} - p_1,\, \mathbb{E}[p_1 - \delta_Y \mid \mathcal{F}_1]\rangle = 2\langle q_{-1} - p_1,\, p_1 - p_1\rangle = 0$ vanishes because $q_{-1}$ is a constant and $\mathbb{E}[\delta_Y \mid \mathcal{F}_1] = p_1$. Hence

$$\mathbb{E}[s(q_{-1}, Y) \mid \mathcal{F}_1] = c_1 - c_2 \|q_{-1} - p_1\|_2^2 - c_2 \mathbb{E}\left[\|p_1 - \delta_Y\|_2^2 \mid \mathcal{F}_1\right]. \tag{13}$$

The squared distance is

$$\|q_{-1} - p_1\|_2^2 = \left\|\left(-\tfrac{1}{6}, \tfrac{1}{6}, 0\right)\right\|_2^2 = \frac{1}{36} + \frac{1}{36} = \frac{1}{18}.$$

Now let $\mathbf{p}'$ be the profile obtained by setting $p'_2 = p_1 = (0.5, 0.5, 0)$ (player 2's prediction improves to match player 1's true belief) while keeping $p'_3 = p_3$ and all wagers unchanged. Under Definition C.3, this constitutes an improvement in player 2's prediction from player 1's perspective, since $p_1$ is the maximizer of $\mathbb{E}[s(\cdot, Y) \mid \mathcal{F}_1]$. The new pooled baseline is

$$q'_{-1} = \tfrac{1}{3}(0.5, 0.5, 0) + \tfrac{2}{3}(0, 1, 0) = \left(\tfrac{1}{6}, \tfrac{5}{6}, 0\right),$$

and the new squared distance to $p_1$ is

$$\|q'_{-1} - p_1\|_2^2 = \left\|\left(-\tfrac{1}{3}, \tfrac{1}{3}, 0\right)\right\|_2^2 = \frac{1}{9} + \frac{1}{9} = \frac{2}{9} > \frac{1}{18} = \|q_{-1} - p_1\|_2^2.$$

*Comparison of expected payoffs.* By the same calculation as (13) applied to $q'_{-1}$,

$$\mathbb{E}[s(q'_{-1}, Y) \mid \mathcal{F}_1] < \mathbb{E}[s(q_{-1}, Y) \mid \mathcal{F}_1].$$

Because the payout subtracts the baseline score, and $w_1 > 0$, the expected payoff of player 1 strictly increases:

$$\mathbb{E}[\pi_1^{\mathrm{II}}(\mathbf{w}, \mathbf{p}', Y) \mid \mathcal{F}_1] = w_1\big(\mathbb{E}[s(p_1, Y) \mid \mathcal{F}_1] - \mathbb{E}[s(q'_{-1}, Y) \mid \mathcal{F}_1] - c_1 w_1\big) > \mathbb{E}[\pi_1^{\mathrm{II}}(\mathbf{w}, \mathbf{p}, Y) \mid \mathcal{F}_1].$$

This is a contradiction to Def. C.3 and hence Variant II fails normality. □

### E.8. Aggregation

After eliciting predictions $\{p_i\}_{i=1}^M$ and wagers $\{w_i\}_{i=1}^M$, the platform aggregates individual reports into a single probabilistic forecast $\hat{p} \in \Delta(\mathcal{Y})$ using wagers as weights.

**Linear pooling.** The wager-weighted arithmetic mean

$$\hat{p}_{\mathrm{lin}} := \frac{\sum_{i=1}^M w_i\, p_i}{W} \in \Delta(\mathcal{Y}) \tag{14}$$

is the unique minimizer of the wager-weighted sum of squared Euclidean distances $\sum_i w_i \|q - p_i\|_2^2$ over $q \in \Delta(\mathcal{Y})$, or equivalently, the minimizer of the wager-weighted reverse KL divergence $\sum_i w_i\, \mathrm{KL}(p_i \| q)$. Linear pooling is computationally simple and preserves the extremes of individual predictions.

**Logarithmic pooling.** The wager-weighted geometric mean

$$\hat{p}_{\log}[y] \propto \prod_{i=1}^{M} p_i[y]^{w_i/W}, \qquad y \in \mathcal{Y}, \tag{15}$$

minimizes the wager-weighted forward KL divergence $\sum_i w_i \, \mathrm{KL}(q \| p_i)$. It is externally Bayesian and satisfies marginalization consistency (Genest, 1984), but concentrates mass on outcomes where all agents agree, and is sensitive to predictions near zero.

# F. Deferred Content from Sec. 4

*Table 3.* The mechanism parameters of WALLA and learning parameters of all learning-based methods.

| Mechanism Parameters | | | | | | Learning Parameters | | | |
|---|---|---|---|---|---|---|---|---|---|
| $c_1$ | $c_2$ | $c_3$ | Aggregation | Scoring rule | Variant | Batch size | Dataset split | MLP hidden sizes | Optimizer |
| 1 | 1 | 1 | Linear pooling | Brier score | WALLA I | 100 | 8:1:1 | 512, 256 | Adam |

*Figure 2.* Evaluated under the setting of homogeneous models with heterogeneous context on the PubMedQA dataset. The three plots show dynamic regret, the average reciprocal rank of the best expert, and the accuracy of aggregated predictions across training batches over time. The performance of WALLA gradually improves over time. A similar trend can be observed in other settings.

## F.1. Learning Wager Policies for LLMs

Each agent is an LLM that produces predictions via its forward pass and learns a wager policy as described in Section 3.3. The base model parameters are frozen; only the wager network is trained. Specifically, we extract the last-layer hidden state after processing input tokens (i.e., question $X$ and context $S$) and pass it to a two-layer perceptron (MLP) to output a non-negative scalar wager. The goal is for each agent to learn its comparative advantage: *on which or what type of questions it is likely to outperform the pool, and by how much.*

This procedure simulates how in a multi-agent system, each LLM participant would independently learn to wager in a deployed wagering mechanism, using only its own hidden states and observed payouts. A key property of the payout function (2) is that each agent can recover the hindsight optimal wager from its own payout alone: since $\pi_i = w_i(s_i - s_{-i} - c_3 w_i)$, the realized score differential $s_i - s_{-i} = \pi_i / w_i + c_3 w_i$ is computable from the agent's observed payout and wager, without access to other agents' predictions, wagers, or scores. The wager network is trained to approximate the resulting target $w^* = ((s_i - s_{-i})/2c_3)^+$, so that at convergence the learned wager approximates $g_i^*(h_i) = (\mathbb{E}_P[s_i - s_{-i} \mid h_i]/2c_3)^+$, the expected score advantage under the true data-generating distribution $P$, conditioned on the model's hidden state.

We defer mechanism parameters $(c_1, c_2, c_3)$ and functions, wager network architecture, and the detailed training procedure to Appendix F.3.

## F.2. Additional Experiment Setup Details

The learning parameters provided in Table 3 are common to all methods. For every dataset, we use an 8:1:1 train-validation-test split and fix the batch size to 100. For learning-based methods, we use the Adam optimizer, tune the learning rate with grid search and adopt early stopping to prevent overfitting. For baselines, we stop at the epoch that achieves the best validation performance. For WALLA, because the training procedure is decentralized, we use the Brier score on the most recent training batches as the early stopping criterion, ensuring that the number of training-batch samples used for early stopping matches the validation-set size. Unless specified otherwise, all methods use linear pooling for aggregation. In

all experiments, the aggregated predictions are evaluated on a held-out test set, where no further updates to the wagering strategy are allowed.

Experiments are run on 4× NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs (96 GiB VRAM each) and a dual-socket AMD EPYC 9354 setup (64 CPU cores total). It provides about 1.1 TiB system RAM (plus 33GiB swap).

### F.3. Implementation details of the wager network

The detailed parameters for the mechanism and the learning procedure are shown in Table 3. Since $w^*$ is bounded in $[0, 1]$, we apply a sigmoid transformation to the wager-head output to preserve the feasible range while improving numerical stability. All participants' wager networks are trained simultaneously without assuming that opponents' strategies are fixed. The wager networks are trained using the mean squared error (MSE) between the predicted wager $w$ and the hindsight target $w^*$. This optimization is equivalent to minimizing the regret of playing optimally in each round when $w^* > 0$. When $w^* = 0$, the MSE update toward zero is less aggressive than directly minimizing regret, since the penalty is quadratic rather than linear.

### F.4. Metrics

Beyond standard predictive performance metrics—area under the ROC curve (AUC) and accuracy (ACC)—and expected calibration error (ECE), we evaluate whether the aggregation weights identify high-performing experts on each question. Since WALLA uses the Brier score as its scoring rule, we additionally evaluate whether the aggregation weights align with per-question Brier-score performance using three metrics: **Kendall's Tau (K-Tau):** Rank correlation between the ordering of models by their wagers and by their per-question Brier-score performance; higher values indicate the wagers correctly rank models by quality. **Mean Reciprocal Rank (MRR):** Average inverse rank of the best-performing model under the ordering induced by aggregation weights; MRR $= 1$ means the best model always receives the highest weight. **Dynamic Regret (D-Regret):** Per-question Brier-score gap between the aggregated prediction and the best individual model; lower values indicate the aggregation is closer to oracle per-question model selection.

We formally define Kendall's Tau, MRR, and dynamic regret in terms of the Brier score as follows.

**Definition F.1** (Kendall's Tau (K-Tau))**.** We use K-Tau to measure the ordinal association between two orderings: (1) the ordering of models by their per-question Brier-score performance and (2) the ordering of models by their wagers. Formally,

$$\tau = \frac{\text{(number of concordant pairs) - (number of discordant pairs)}}{\text{(number of pairs)}}.$$

**Definition F.2** (Mean Reciprocal Rank (MRR) of the Best Experts)**.** The per-sample reciprocal rank is the inverse rank of the best expert in terms of Brier score. Formally,

$$\text{MRR} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{\text{rank}(\arg\min_j \|\delta_y^{(n)} - p_j^{(n)}\|_2^2)},$$

where $N$ is the number of samples and the rank is induced by the aggregation weights.

**Definition F.3** (Dynamic Regret (D-Regret) in Brier Score)**.** The D-Regret is defined as

$$\frac{1}{N} \sum_{n=1}^{N} \|\delta_y^{(n)} - p_{\text{agg}}^{(n)}\|_2^2 - \min_j \|\delta_y^{(n)} - p_j^{(n)}\|_2^2.$$

It measures the Brier-score gap between the aggregated prediction and the best individual model on each question.

### F.5. Baselines

Among *learning-free methods*, UniformAvg performs equally weighted aggregation. SelfCertainty (Kang et al., 2025) uses the predicted distribution to estimate confidence and treats it as the aggregation weight. PackLLM (Mavromatis et al., 2024) uses the inverse perplexity on input tokens as weights, requiring access to model parameters. These methods can be considered decentralized and some are uncertainty-aware, but they do not learn from data, may be sensitive to miscalibrated confidence, and are not incentive compatible—participants can misreport confidence or perplexity to gain disproportionate weight.

*Learning-based methods*, including pre-inference routers (Chen et al., 2024; Ong et al., 2025; Song et al., 2025) and StackedGen (Wolpert, 1992), require centralized control to train routing or aggregation networks. Pre-inference routers learn weights conditional on input features but do not capture each model's internal uncertainty on the specific question. StackedGen trains on concatenated hidden states and is therefore uncertainty-aware, but requires centralized access to all models' internals. Since StackedGen has access to all models' hidden states, it can be considered as a natural performance upper bound for decentralized methods.

RouterDC (Chen et al., 2024) is a pre-inference router that finetunes a small language model DeBERTaV3-base (He et al., 2023) via dual-contrastive learning that aligns a query's embedding with the top-K most capable LLM for that specific task while distancing it from the last-K underperforming models.[6]

Song et al. (2025) proposes a pre-inference router inspired by Item Response Theory (IRT) called IRTRouter. It treats user queries as "test items" and LLMs as "respondents," and explicitly models the relationship between a specific query's difficulty and an individual model's latent capabilities. It uses a frozen BERT model to retrieve the query embedding. We adopt the variant named NIRTRouter that adopts neural networks for the IRT implementation, following the NCDM model (Wang et al., 2020).[7]

RouteLLM (Ong et al., 2025) was designed to choose between a more capable but expensive model and a weaker and cheaper model using human pairwise preference as feedback. Its base router is a BERT model that is finetuned using preference data. We extend this framework for routing on more LLMs using cross-entropy loss and pairwise ranking loss. [8]

Stacked generalization (StackedGen) (Wolpert, 1992) is a classical ensembling technique that learns a neural network to combine diverse predictive models to achieve greater overall accuracy than any single model could produce alone. We train a two-layer perceptron with cross-entropy loss. We use the same hidden states of LLMs as input to the perceptron as WALLA to be consistent.

Uniform averaging (UniformAvg) is a simple method that averages the predicted probability distributions with equal weights (i.e., $p_{\text{agg}} = \frac{\sum_{i=1}^{M} p_i}{M}$).

PackLLM (Mavromatis et al., 2024) is a learning-free ensembling method. Its main idea is to minimize the perplexity of the ensemble over the input prompt. Due to PackLLM's reliance on the perplexity of the input tokens for each model, it requires open-weight models. In a decentralized setting where each model is controlled by a different provider, the provider can manipulate the report of the perplexity to gain more weight in aggregation.[9]

SelfCertainty (Kang et al., 2025) is also a learning-free ensembling that uses model confidence as weight for averaging. We use the variant that quantifies confidence through the Kullback-Leibler (KL) divergence between the model's predicted token probability distribution and a uniform distribution, as it performs best.[10]

### F.6. Datasets

Table 4 shows basic information about each dataset.

*Table 4.* Dataset attributes.

| | PubMedQA (Jin et al., 2019) | BayeX | MedMCQA (Pal et al., 2022) | MMLU (Hendrycks et al., 2021) | Arc-Challenge (Clark et al., 2018) |
|---|---|---|---|---|---|
| Size | 37.82K | 10K | 183K | 116K | 1.17K |
| Type | Binary Classification | Matching Distribution | Multiple Choice | Multiple Choice | Multiple Choice |
| Private Context | ✓ | ✓ | ✗ | ✗ | ✗ |

PubMedQA (Jin et al., 2019)[11] is a specialized biomedical research dataset designed for answering questions based on

[6]We use the public implementation of RouterDC from https://github.com/shuhao02/RouterDC. We use the pretrained DeBERTaV3 from https://huggingface.co/microsoft/deberta-v3-base.

[7]We use the public implementation of NIRTRouter from https://github.com/Mercidaiha/IRT-Router. We adopt the pretrained BERT from https://huggingface.co/google-bert/bert-base-uncased.

[8]We adapt the public implementation of RouteLLM from https://github.com/lm-sys/RouteLLM.git. We adopt the pretrained BERT from https://huggingface.co/google-bert/bert-base-uncased.

[9]We use the public implementation of PackLLM from https://github.com/cmavro/PackLLM.

[10]We use the public implementation of SelfCertainty from https://github.com/backprop07/Self-Certainty.

[11]We use the publicly available PubMedQA dataset on Hugging Face: https://huggingface.co/datasets/qiaojin/

medical abstracts. The primary task is to provide a "yes," "no," or "maybe" response to a clinical question by synthesizing information from an associated PubMed abstract. For our experiment, we convert this into a binary classification by focusing on "yes" and "no" questions. We also balance the dataset such that there is a similar number of "yes" and "no" questions. We extract the abstract per question and randomly assign relevant or irrelevant abstracts to a subset of participating models.

MedMCQA (Pal et al., 2022)[12] MedMCQA is a large-scale, multiple-choice question-answering dataset specifically curated from Indian medical entrance exams (AIIMS and NEET-PG). It covers 21 diverse medical subjects, such as anatomy, pathology, and pharmacology. Because these questions are designed to test human medical students, they require deep domain knowledge and multi-step reasoning, making MedMCQA a standard tool for measuring the clinical competency of general and medical-purpose LLMs.

MMLU (Hendrycks et al., 2021)[13] is one of the most comprehensive benchmarks for evaluating the general knowledge and problem-solving capabilities of language models. It spans 57 subjects across STEM, the humanities, social sciences, and more, testing both world knowledge and analytical skill. The questions are multiple-choice and range in difficulty from elementary levels to advanced professional standards (such as law or medicine). MMLU has become the industry standard for reporting a model's foundational "intelligence" and its ability to handle a vast breadth of human knowledge.

ARC-Challenge (Clark et al., 2018)[14] The ARC-Challenge is a high-difficulty subset of the AI2 Reasoning Challenge, consisting of grade-school science questions that are particularly difficult for automated systems to solve. The Challenge set contains questions that cannot be answered by simple keyword matching or statistical co-occurrence alone; they often require a degree of logical reasoning, common sense, or a basic understanding of scientific processes. It remains a key metric for determining how well a model can move beyond rote memorization toward actual conceptual understanding.

BayesX is a new dataset we constructed to capture a setting in which additional information reduces epistemic uncertainty about the data-generating process while increasing aleatoric uncertainty about the realized outcome. Each example describes a binary cluster-saturation forecasting problem. The prior probability is either $0.1$ or $0.9$, and the context contains a binary warning signal with sampled true-positive and false-positive rates. The evidence is selected so that receiving additional information can move the posterior closer to uncertainty.The ground truth is therefore the posterior probability rather than only a hard binary label.[15]

### F.7. Scenario I: Homogeneous Models with Private Contexts

All participants are copies of the same LLM; heterogeneous predictions arise from private information (mirroring Example E.1). For each question, one randomly selected model receives relevant context; the rest receive none. We use Aloe on PubMedQA and Gemma2 on BayesX, scaling participants from 4 to 12.

On PubMedQA (Table 5 (appendix)), WALLA achieves AUC and ACC comparable to StackedGen and RouteLLM (with context)—both of which require centralized access to models' hidden states or private inputs—while outperforming all other baselines. The wager network learns from payout signals and hidden states to identify when the model has received relevant context, wagering high when it holds a comparative advantage and near zero otherwise (Fig. 2). As the pool grows from 4 to 12, learning-free methods degrade substantially (e.g., UniformAvg AUC drops from 70.3 to 63.1), while WALLA remains robust. On BayesX (Table 6 (appendix)), WALLA outperforms all baselines except RouteLLM (with context) and StackedGen, that easily recognize context in inputs. Without access to contexts, RouteLLM drops to uniform averaging. SelfCertainty performs poorly because observing the signal makes the posterior more faithful but closer to uniform, inverting its confidence estimate.

### F.8. Scenario II: Heterogeneous Models without Context

In this setting, we use four heterogeneous LLMs without additional context, so that predictions differ due to distinct model weights (mirroring Example E.2). We train a wager network for each LLM on the MMLU dataset, and evaluate in-distribution on MedMCQA and MMLU, and out-of-distribution on ARC-Challenge. Results with four participants are in

PubMedQA.

[12]We use the publicly available MedMCQA dataset on Hugging Face: https://huggingface.co/datasets/openlifescienceai/medmcqa.

[13]We use the publicly available MMLU dataset on Hugging Face: https://huggingface.co/datasets/cais/mmlu.

[14]We use the publicly available ARC-Challenge dataset on Hugging Face: https://huggingface.co/datasets/allenai/ai2_arc.

[15]We include the BayesX dataset in the anonymous repo https://anonymous.4open.science/r/MultiLLMs-B48C/README.md.

Table 9; two-participant results are in Tables 8–10 (appendix).

Table 9 shows that WALLA performs comparably to the centralized StackedGen across all four datasets. Aggregation can improve over individual models—especially on MedMCQA—and on MMLU and ARC-Challenge the aggregated prediction matches the performance of the strongest individual model (Gemma2). The wager network here learns a different mapping than in Scenario I: rather than learning context assignment, it must learn each model's domain-dependent comparative advantage from the hidden state and payout signals.

Learning-free methods are notably less robust to pool composition. On MedMCQA, weaker models (Gemma2, BioMistral) causes performance drops for UniformAvg, PackLLM, and SelfCertainty, because miscalibrated confidence in weaker models can inflate their aggregation weight. This illustrates another limitation of confidence-based aggregation that holds even without strategic manipulation.

### F.9. Robustness to Variants, Score–Wager Alignment, and Model Calibration

**Robustness to mechanism variants.** We evaluate WALLA II and log pooling as alternative variants. Both lead to results similar to WALLA I with linear pooling. This indicates that performance of WALLA is not driven by a particular choice of baseline or aggregation rule. The similarity between WALLA I and WALLA II is also consistent with theory: both variants preserve score–wager alignment, differing mainly in the normality/no-arbitrage tradeoff (Section E.6). Likewise, the similarity between linear pooling and log pooling suggests that the learned wagers, rather than the aggregation rule, are the primary driver of the aggregation quality.

**Score–wager alignment.** Fig. 3 shows that learned wagers are positively correlated with realized Brier-score advantages, providing direct evidence that models learn to wager in proportion to their expected advantage. We conduct further experiments to verify that the hidden states contain sufficient information to separately predict both the model's own score and the baseline score; we find that the difference between these predictions aligns closely with the learned wager (see Figs. 4–6 in the appendix).

**Calibration.** The mechanism also incentivizes participants to calibrate their predictions: even when a model's belief $Q_i \neq P$, calibration aligns expected scores better with realized scores, so that the learned wager better reflects *actual* comparative advantage. We conduct calibration via temperature scaling (Xie et al., 2024), which predicts an appropriate temperature conditioned on the last-layer hidden state. In Table 11, we show that when one model is calibrated while the others remain unchanged, the calibrated model receives improved net payoffs because its expected score advantage more accurately reflects its realized advantage. This suggests that rational participants have an incentive to calibrate—discouraging both overconfidence and underconfidence. We also report WALLA I results when all participants are calibrated for Scenarios II and III. The results show that calibrated aggregation can substantially improve accuracy on in-distribution tasks while maintaining similar performance on out-of-distribution tasks.

*Table 5.* Homogeneous models (Aloe) on PubMedQA with private contexts. For each question, one of $M \in \{4, 8, 12\}$ models is randomly selected to receive a relevant abstract; the rest receive none.[16]

| | 4 | | | | | 8 | | | | | 12 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | DRegret ↓ | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | DRegret ↓ | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | DRegret ↓ |
| UniformAvg | 70.29 | 71.66 | 2.27 | - | 20.80 | 64.85 | 67.20 | 3.12 | - | 25.35 | 63.10 | 66.17 | 4.49 | - | 27.30 |
| SelfCertainty | 71.78 | 79.23 | 2.67 | 86.76 | 13.62 | 69.31 | 76.92 | 3.33 | 84.73 | 16.50 | 68.05 | 75.66 | 3.65 | 84.07 | 18.12 |
| PackLLM | 78.23 | 85.22 | 10.23 | 88.64 | 8.01 | 75.65 | 81.71 | 9.38 | 87.85 | 11.99 | 73.65 | 78.97 | 7.44 | 87.67 | 14.64 |
| RouterDC w/ context | $69.36_{\pm 13.24}$ | $72.12_{\pm 25.21}$ | $2.58_{\pm 20.64}$ | $61.24_{\pm 351.39}$ | $20.82_{\pm 20.95}$ | $65.17_{\pm 3.02}$ | $67.51_{\pm 1.47}$ | $3.10_{\pm 1.97}$ | $47.36_{\pm 125.47}$ | $25.46_{\pm 2.19}$ | $63.35_{\pm 4.16}$ | $66.17_{\pm 0.42}$ | $4.75_{\pm 2.33}$ | $41.58_{\pm 174.74}$ | $27.20_{\pm 1.31}$ |
| NIRTRouter w/ context | $70.19_{\pm 1.32}$ | $71.71_{\pm 0.63}$ | $2.15_{\pm 1.17}$ | $52.26_{\pm 14.97}$ | $20.80_{\pm 0.01}$ | $64.85_{\pm 0.00}$ | $67.53_{\pm 0.00}$ | $3.18_{\pm 0.00}$ | $33.38_{\pm 24.19}$ | $25.57_{\pm 0.00}$ | $63.10_{\pm 0.01}$ | $66.17_{\pm 0.00}$ | $4.91_{\pm 0.19}$ | $24.29_{\pm 1.11}$ | $27.30_{\pm 0.00}$ |
| RouteLLM w/ context | $82.32_{\pm 0.07}$ | $87.48_{\pm 0.06}$ | $7.24_{\pm 0.07}$ | $87.81_{\pm 0.83}$ | $3.51_{\pm 0.08}$ | $82.22_{\pm 0.04}$ | $87.50_{\pm 0.03}$ | $7.21_{\pm 0.02}$ | $85.77_{\pm 1.05}$ | $3.54_{\pm 0.02}$ | $82.52_{\pm 0.05}$ | $87.39_{\pm 0.03}$ | $7.17_{\pm 0.02}$ | $84.15_{\pm 0.99}$ | $3.55_{\pm 0.02}$ |
| StackedGen | $82.60_{\pm 0.38}$ | $87.55_{\pm 0.10}$ | $7.58_{\pm 0.09}$ | $87.15_{\pm 0.25}$ | $3.42_{\pm 0.04}$ | $82.63_{\pm 0.21}$ | $87.72_{\pm 0.05}$ | $7.48_{\pm 0.07}$ | $86.06_{\pm 0.22}$ | $3.26_{\pm 0.03}$ | $82.69_{\pm 0.24}$ | $87.75_{\pm 0.17}$ | $7.50_{\pm 0.16}$ | $85.03_{\pm 0.63}$ | $3.27_{\pm 0.11}$ |
| WALLA I | $81.07_{\pm 0.73}$ | $87.01_{\pm 0.11}$ | $8.57_{\pm 0.15}$ | $88.15_{\pm 0.33}$ | $5.34_{\pm 0.15}$ | $80.42_{\pm 1.78}$ | $86.31_{\pm 0.19}$ | $10.57_{\pm 0.18}$ | $85.29_{\pm 0.45}$ | $6.69_{\pm 0.34}$ | $78.85_{\pm 0.81}$ | $85.60_{\pm 0.27}$ | $11.37_{\pm 0.28}$ | $85.02_{\pm 0.14}$ | $8.34_{\pm 0.04}$ |
| WALLA II | $81.16_{\pm 0.22}$ | $87.04_{\pm 0.21}$ | $8.16_{\pm 1.26}$ | $88.17_{\pm 0.15}$ | $5.33_{\pm 0.25}$ | $81.08_{\pm 0.41}$ | $86.21_{\pm 0.29}$ | $10.53_{\pm 0.50}$ | $85.55_{\pm 0.57}$ | $6.76_{\pm 0.73}$ | $79.04_{\pm 0.52}$ | $85.80_{\pm 0.28}$ | $11.30_{\pm 0.34}$ | $85.12_{\pm 0.17}$ | $7.99_{\pm 0.10}$ |

[16] Aloe achieves AUC=82.39, ACC=87.57, ECE=7.38 with the abstract and AUC=60.65, ACC=63.10, ECE=7.83 without.

[17] Gemma2 achieves KLD=3.24 and TVD=10.27 with evidence, and KLD=78.30 and TVD=45.46 without.

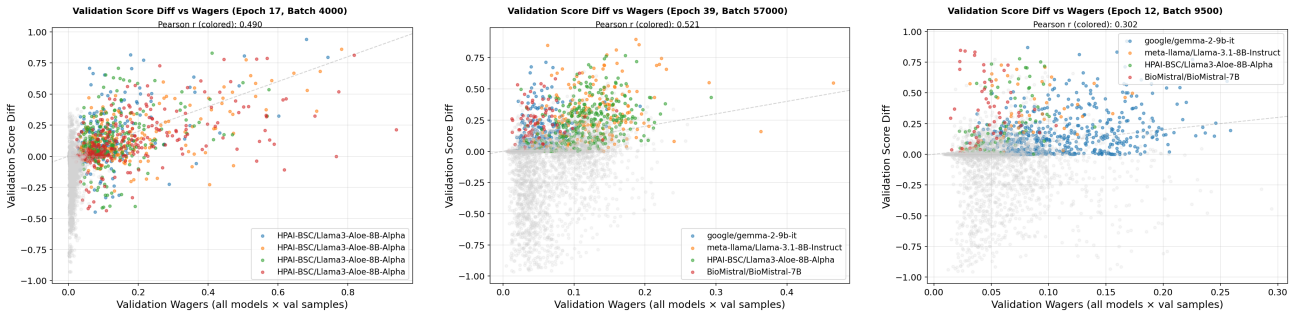


*Figure 3.* Score-wager alignment for WALLA I. Learned wager ($x$-axis) vs. realized score differentials $s(p_i, y) - s_{-i}$ ($y$-axis), which equal the hindsight optimal wagers under our parameter choice ($c_1 = 1, c_2 = 0.5, c_3 = 1$). Each panel corresponds to one scenario: homogeneous models on PubMedQA (left), heterogeneous models on MedMCQA (middle) and MMLU (right). Colored dots highlight models with context (PubMedQA) or the best-performing model (other datasets). Pearson correlations are reported at the top of each panel. The positive correlations confirm that LLMs learn to wager in proportion to their realized advantage. On PubMedQA, models successfully differentiate whether they received relevant context, wagering near zero when they did not.

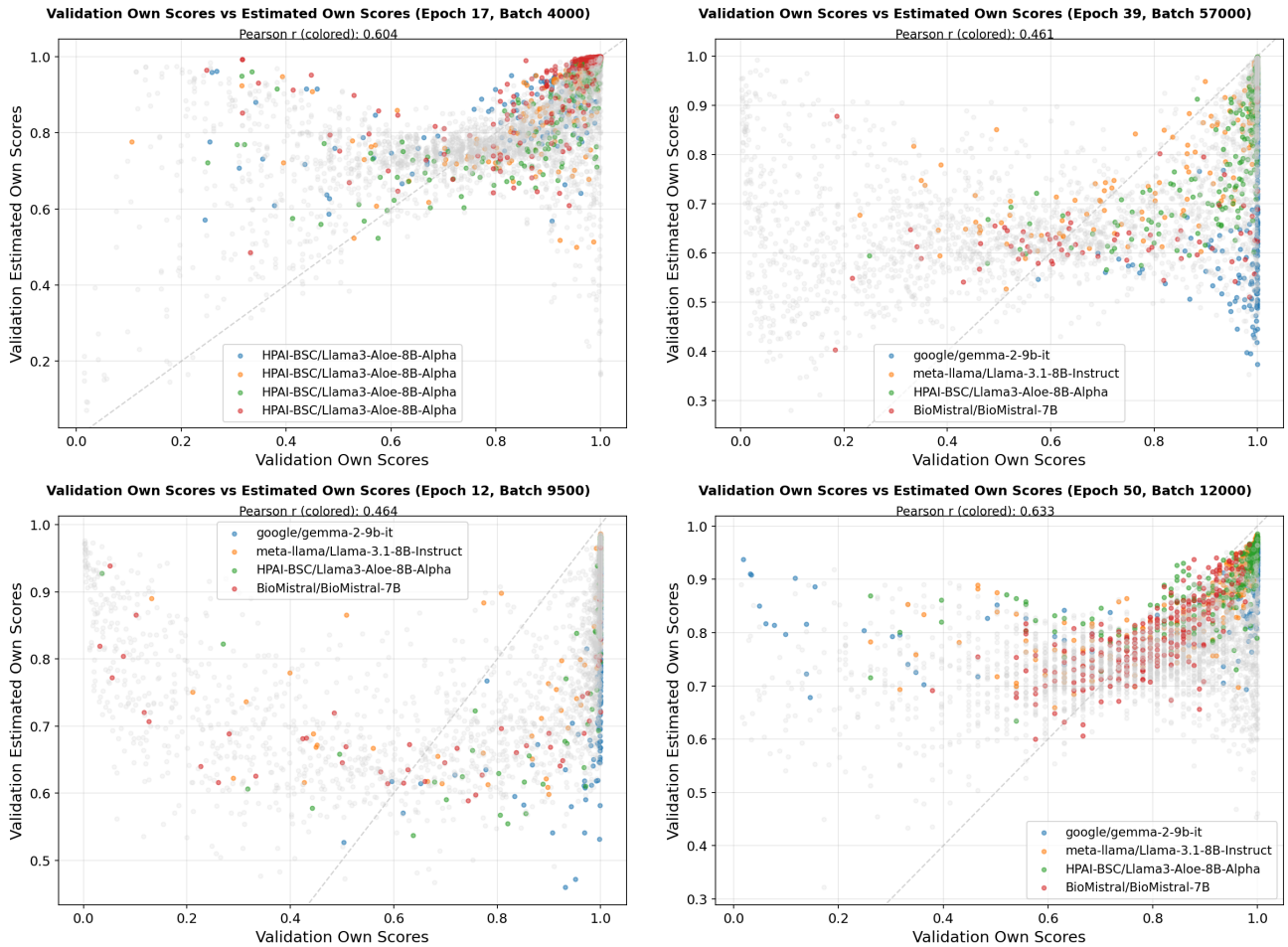


*Figure 4.* Demonstration of predictive power of LLM hidden states for its own Brier scores. We train a prediction head for the Brier scores (i.e., $1 - \frac{1}{2}\|\delta_y - p_i\|_2^2$ ) of the model's own predictions, using the LLM's last layer hidden state as input. The $x$-axis is the actual Brier scores, and the $y$-axis is the predicted Brier scores. The top-left plot is evaluated under the setting of homogeneous models with heterogeneous contexts on PubMedQA. The top-right and the bottom-left plots are evaluated with heterogeneous models without context on MedMCQA and MMLU, respectively. The bottom-right plot is evaluated with heterogeneous models and contexts on PubMedQA (relevant context only). For PubMedQA, the models provided with contexts are shown in color, and for other datasets, the best-performing models are shown in color. The Pearson correlations for the colored dots are provided at the top of the plots.

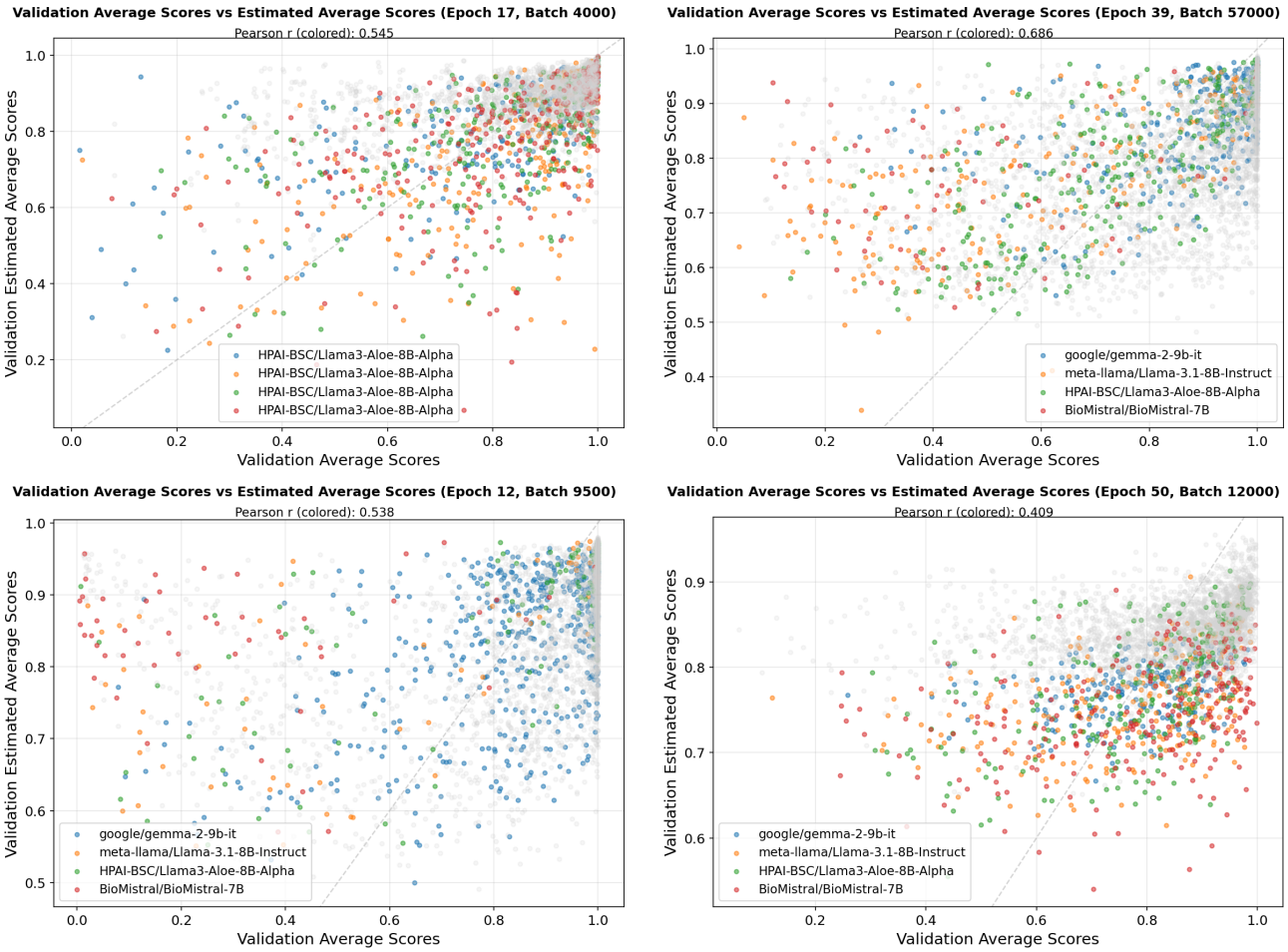


*Figure 5.* Demonstration of predictive power of LLM hidden states for the baseline average scores under Variant I (i.e., $s^{\mathrm{I}}_{-i} = \sum_{j \neq i} w_j s_j / W_{-i}$ where $s_j = 1 - \frac{1}{2}\|\delta_y - p_j\|_2^2$) . We train a prediction head for the baseline average scores, using the LLM's last layer hidden state as input. The $x$-axis is the actual baseline average scores, and the $y$-axis is the predicted baseline average scores. The top-left plot is evaluated under the setting of homogeneous models with heterogeneous contexts on PubMedQA. The top-right and the bottom-left plots are evaluated with heterogeneous models without context on MedMCQA and MMLU, respectively. The bottom-right plot is evaluated with heterogeneous models and contexts on PubMedQA (relevant context only). For PubMedQA, the models provided with contexts are shown in color, and for other datasets, the best-performing models are shown in color. The Pearson correlations for the colored dots are provided at the top of the plots.

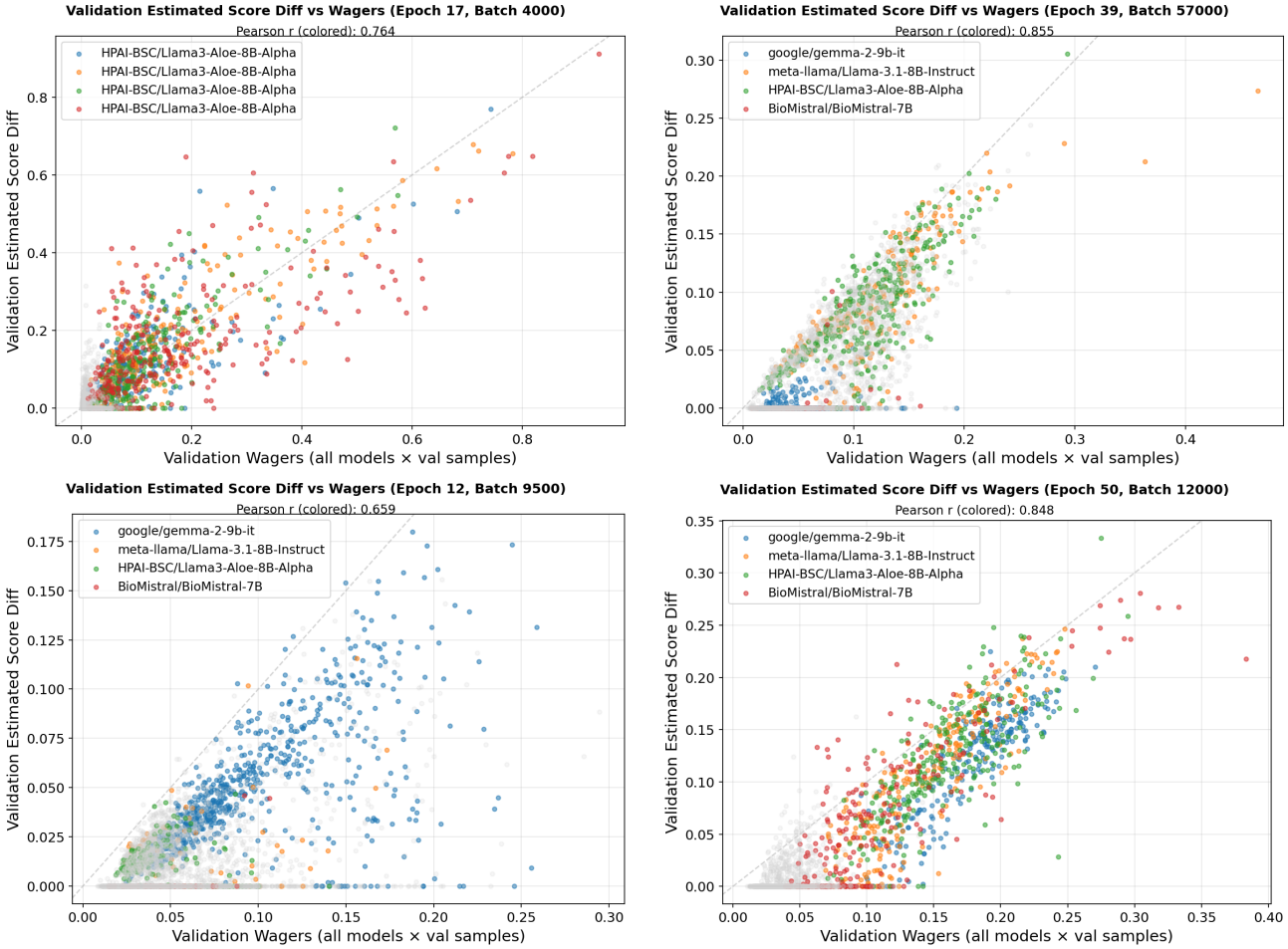


*Figure 6.* Demonstration of the alignment between the wagers and the differences between the predicted own scores and the predicted baseline average scores, which can be interpreted as the optimal wager of one's belief. We train a prediction head for each of LLM's own scores, the baseline average scores, and the wager, using the LLM's last layer hidden state as input. The $x$-axis is the wager placed, and the $y$-axis is the predicted score differences. The top-left plot is evaluated under the setting of homogeneous models with heterogeneous contexts on PubMedQA. The top-right and the bottom-left plots are evaluated with heterogeneous models without context on MedMCQA and MMLU, respectively. The bottom-right plot is evaluated with heterogeneous models and contexts on PubMedQA (relevant context only). For PubMedQA, the models provided with contexts are shown in color, and for other datasets, the best-performing models are shown in color. The Pearson correlations for the colored dots are provided at the top of the plots. They show strong alignment even though the wagers, the predicted own scores, and the predicted baseline average scores are trained with separate heads.

*Table 6.* Homogeneous models (Gemma2) on BayesX with private contexts. For each question, one of $M \in \{4, 8, 12\}$ models is randomly selected to receive evidence.[17]

| | 4 | | | | 8 | | | | 12 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | KLD ↓ | TVD ↓ | MRR ↑ | DRegret ↓ | KLD ↓ | TVD ↓ | MRR ↑ | DRegret ↓ | KLD ↓ | TVD ↓ | MRR ↑ | DRegret ↓ |
| UniformAvg | 25.64 | 32.96 | - | 19.65 | 41.09 | 38.93 | - | 28.79 | 48.67 | 40.76 | - | 32.02 |
| SelfCertainty | 61.44 | 43.99 | 25.00 | 37.37 | 68.46 | 44.58 | 12.50 | 38.74 | 70.80 | 44.59 | 8.33 | 38.93 |
| PackLLM | 21.58 | 30.70 | 100.00 | 16.66 | 37.36 | 37.75 | 100.00 | 26.75 | 45.32 | 39.93 | 100.00 | 30.60 |
| RouterDC w/ context | 22.88±0.26 | 30.00±0.38 | 65.55±12.27 | 16.77±0.18 | 39.96±14.38 | 38.38±6.81 | 39.01±64.03 | 28.02±9.66 | 48.96±0.60 | 40.73±0.12 | 26.15±3.64 | 32.00±0.22 |
| NIRTRouter w/ context | 22.49±17.12 | 30.72±16.17 | 52.11±0.32 | 17.11±15.72 | 40.50±0.49 | 38.76±0.03 | 33.97±0.00 | -0.35±1.46 | 45.18±0.11 | 38.97±0.02 | 25.91±0.11 | 29.74±0.11 |
| RouteLLM w/o context | 25.67±0.22 | 32.96±0.01 | 52.08±0.00 | 19.66±0.08 | 41.11±0.20 | 38.92±0.00 | 33.97±0.00 | 28.78±0.01 | 48.68±0.15 | 40.76±0.06 | 25.91±0.05 | 32.01±0.10 |
| RouteLLM w/ context | 0.64±0.07 | 3.75±0.92 | 100.00±0.00 | -2.31±0.07 | 0.83±2.32 | 4.67±13.61 | 100.00±0.00 | -2.12±2.25 | 0.93±2.46 | 5.27±10.68 | 100.00±0.00 | -2.02±2.40 |
| StackedGen | 17.63±0.55 | 27.88±0.38 | 100.00±0.00 | 13.43±0.45 | 32.69±0.51 | 35.88±0.19 | 97.59±7.66 | 24.30±1.36 | 41.43±0.11 | 38.80±0.05 | 91.98±10.80 | 28.73±0.07 |
| WALLA I | 21.46±2.09 | 29.46±1.03 | 81.25±0.00 | 15.77±1.35 | 36.04±0.45 | 37.08±0.14 | 89.06±0.00 | 25.86±0.22 | 44.61±0.14 | 39.69±0.04 | 92.33±0.00 | 30.18±0.07 |
| WALLA II | 20.46±0.19 | 28.98±0.13 | 81.25±0.00 | 15.13±0.14 | 36.19±0.18 | 37.14±0.08 | 89.06±0.00 | 25.95±0.12 | 42.98±0.75 | 39.31±0.20 | 92.33±0.00 | 29.51±0.34 |

*Table 7.* Evaluation of homogeneous models with heterogeneous contexts using Aloe on PubMedQA. Out of 4, 8, or 12 models, only one model is provided with the relevant abstract to the question. In this table, we filter out data samples where all models can answer correctly. All metrics are scaled by $\times 100$.

| | 4 | | | | | 8 | | | | | 12 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | DRegret ↓ | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | DRegret ↓ | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | DRegret ↓ |
| RouterDC | 24.58±27.74 | 31.57±61.88 | 30.69±59.00 | 61.57±366.04 | 39.32±39.64 | 19.86±2.61 | 20.32±3.61 | 43.30±4.81 | 48.12±146.06 | 48.17±4.22 | 20.22±6.03 | 17.03±1.03 | 47.61±0.21 | 41.98±186.21 | 51.47±2.50 |
| NIRTRouter | 25.81±3.03 | 30.56±1.55 | 30.77±1.54 | 52.02±8.94 | 39.27±0.04 | 19.29±0.00 | 20.36±0.00 | 43.32±0.00 | 33.89±17.09 | 48.38±0.00 | 19.92±0.03 | 17.03±0.00 | 47.67±0.00 | 24.50±1.64 | 51.66±0.01 |
| RouteLLM | 72.54±0.15 | 69.27±0.15 | 4.89±0.09 | 89.30±0.76 | 6.47±0.14 | 72.55±0.07 | 72.55±0.07 | 4.99±0.26 | 87.92±0.79 | 6.53±0.05 | 73.11±0.11 | 69.08±0.08 | 5.15±0.08 | 86.59±0.89 | 6.55±0.04 |
| StackedGen | 73.18±0.70 | 69.44±0.25 | 4.72±0.24 | 88.76±0.25 | 6.31±0.08 | 73.43±0.37 | 69.55±0.25 | 4.62±0.44 | 87.34±0.32 | 17.89±0.37 | 73.63±0.29 | 69.16±0.16 | 5.08±0.16 | 85.87±0.16 | 6.57±0.13 |
| UniformAvg | 26.05 | 30.44 | 30.89 | 54.57 | 39.27 | 19.29 | 20.36 | 43.32 | 39.10 | 48.38 | 19.92 | 17.03 | 47.68 | 31.92 | 51.66 |
| PackLLM | 78.23 | 63.72 | 4.72 | 89.58 | 15.18 | 48.75 | 55.15 | 9.35 | 88.83 | 22.87 | 41.43 | 48.42 | 41.43 | 88.45 | 28.00 |
| SelfCertainty | 54.42 | 48.94 | 23.12 | 67.94 | 31.05 | 51.38 | 43.15 | 27.94 | 62.45 | 36.41 | 49.71 | 39.82 | 30.81 | 60.65 | 39.30 |
| Variant I | 68.66±1.34 | 66.66±1.02 | 4.39±0.88 | 85.36±1.33 | 9.68±0.97 | 67.84±3.25 | 61.23±4.98 | 7.54±1.01 | 78.60±4.56 | 13.53±4.19 | 59.92±1.65 | 63.38±0.43 | 3.79±1.19 | 86.48±0.50 | 14.93±0.70 |
| Variant II | 68.77±0.42 | 67.05±0.51 | 3.85±0.29 | 85.87±0.81 | 9.39±0.38 | 68.87±0.99 | 61.92±0.93 | 8.11±0.96 | 79.29±1.00 | 12.73±0.74 | 60.55±1.10 | 63.97±0.52 | 2.94±1.61 | 86.79±1.00 | 14.39±0.63 |

*Table 8.* 2 Heterogeneous models (Llama3.1 and Aloe) without context on three datasets. MedMCQA and MMLU are in-distribution; ARC-Challenge is out-of-distribution, with MMLU as the training set.

| | MedMCQA (i.d.) | | | | | MMLU (i.d.) | | | | | ARC-Challenge (o.o.d.) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ |
| Llama3.1 | 78.97 | 2.64 | 69.51 | -21.95 | 10.62 | 81.99 | 2.52 | 68.51 | -25.96 | 7.37 | 81.30 | 3.06 | 68.39 | -26.43 | 8.89 |
| Aloe | 79.77 | 3.18 | 80.49 | 21.95 | 10.42 | 80.23 | 8.77 | 81.49 | 25.96 | 11.10 | 80.35 | 8.58 | 81.61 | 26.43 | 11.31 |
| RouterDC | 82.01±3.58 | 4.36±0.62 | 80.18±3.91 | 20.71±15.64 | 7.49±1.11 | 82.21±0.04 | 2.54±0.13 | 76.30±8.83 | 5.19±35.31 | 6.68±0.03 | 82.89±0.05 | 2.86±0.45 | 76.32±8.99 | 5.29±35.96 | 7.07±0.03 |
| NIRTRouter | 81.98±3.61 | 4.27±0.98 | 80.56±0.94 | 22.24±3.75 | 7.49±1.13 | 82.36±0.57 | 2.23±0.23 | 68.51±0.03 | -25.97±0.13 | 6.54±0.55 | 82.78±0.00 | 3.42±0.00 | 68.39±0.00 | -26.43±0.00 | 7.00±0.00 |
| RouteLLM | 81.82±2.71 | 3.89±1.74 | 80.56±0.94 | 22.24±3.75 | 7.51±1.14 | 82.56±0.19 | 1.86±1.28 | 68.51±0.03 | -25.97±0.13 | 6.37±0.07 | 83.13±0.00 | 2.13±0.00 | 68.39±0.00 | -26.43±0.00 | 6.94±0.00 |
| StackedGen | 82.61±0.28 | 2.16±0.32 | 85.51±0.39 | 42.04±1.57 | 6.35±0.11 | 82.52±0.21 | 2.29±0.21 | 77.47±1.03 | 9.88±4.12 | 6.21±0.19 | 83.33±0.21 | 3.07±0.35 | 75.22±0.37 | 0.87±1.46 | 6.61±0.03 |
| UniformAvg | 81.76 | 4.60 | 69.51 | 0.00 | 7.58 | 82.22 | 2.47 | 68.51 | 0.00 | 6.66 | 82.87 | 3.05 | 68.39 | 0.00 | 7.05 |
| PackLLM | 81.54 | 4.18 | 79.04 | 16.17 | 7.62 | 82.27 | 2.49 | 72.53 | -9.90 | 6.65 | 82.70 | 2.80 | 75.70 | 2.78 | 7.15 |
| SelfCertainty | 81.73 | 2.24 | 88.64 | 54.56 | 7.32 | 82.15 | 5.07 | 88.57 | 54.29 | 7.56 | 82.87 | 3.94 | 87.74 | 50.96 | 7.34 |
| WALLA I | 82.02±0.32 | 4.53±0.28 | 73.69±3.26 | -5.25±13.06 | 7.57±0.12 | 82.41±0.17 | 1.84±0.09 | 69.07±1.01 | -23.72±4.03 | 6.49±0.11 | 83.00±0.14 | 2.44±0.17 | 68.30±0.20 | -26.78±0.81 | 7.01±0.04 |
| WALLA II | 82.02±0.23 | 4.48±0.32 | 74.07±6.44 | -3.73±25.78 | 7.56±0.10 | 82.35±0.33 | 1.90±0.23 | 68.69±0.67 | -25.23±2.67 | 6.51±0.12 | 82.91±0.18 | 2.34±0.37 | 68.30±0.28 | -26.78±1.13 | 7.01±0.05 |
| WALLA I (log-pool) | 81.73±0.17 | 2.70±0.66 | 73.96±5.91 | -4.17±23.66 | 7.19±0.08 | 82.33±0.08 | 3.09±0.43 | 70.08±0.37 | -19.67±1.50 | 6.55±0.09 | 82.65±0.29 | 3.02±0.80 | 68.02±0.49 | -27.91±1.95 | 6.98±0.37 |
| WALLA I (calibrated) | 82.55±0.16 | 5.20±0.17 | 75.03±4.71 | 0.10±18.83 | 6.78±0.02 | 82.76±0.05 | 2.45±0.08 | 79.11±1.47 | 16.44±5.88 | 5.79±0.03 | 82.87±0.11 | 2.88±0.44 | 81.67±0.17 | 26.70±0.70 | 6.64±0.05 |

*Table 9.* Heterogeneous models without context on three datasets. MedMCQA and MMLU are in-distribution; ARC-Challenge is out-of-distribution, with MMLU as the training set.

| | MedMCQA (i.d.) | | | | | MMLU (i.d.) | | | | | ARC-Challenge (o.o.d.) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ |
| Gemma2 | 62.90 | 25.28 | 64.99 | 5.93 | 47.28 | 87.18 | 8.72 | 82.80 | 32.61 | 12.94 | 90.43 | 7.55 | 89.18 | 38.43 | 10.67 |
| Aloe | 79.77 | 3.18 | 59.49 | 14.42 | 15.77 | 80.23 | 8.77 | 53.42 | 3.68 | 21.34 | 80.35 | 8.58 | 51.17 | -0.43 | 23.43 |
| Llama3.1 | 78.97 | 2.64 | 51.56 | 4.47 | 15.97 | 81.99 | 2.52 | 40.88 | -8.01 | 17.61 | 81.30 | 3.06 | 37.43 | -12.20 | 21.01 |
| BioMistral | 47.47 | 17.10 | 37.71 | -24.82 | 56.24 | 66.05 | 10.92 | 32.59 | -28.27 | 39.95 | 66.61 | 10.95 | 30.68 | -25.80 | 40.65 |
| UniformAvg | 76.45 | 8.10 | - | 0.00 | 21.05 | 85.15 | 5.10 | - | 0.00 | 13.87 | 87.39 | 7.40 | - | 0.00 | 14.23 |
| SelfCertainty | 74.42 | 4.58 | 79.78 | 43.61 | 22.08 | 86.58 | 1.36 | 88.75 | 63.21 | 11.86 | 89.22 | 2.70 | 91.59 | 66.96 | 10.71 |
| PackLLM | 79.00 | 9.93 | 50.77 | 8.85 | 18.93 | 84.94 | 5.13 | 37.79 | -27.81 | 14.28 | 86.26 | 6.87 | 36.38 | -21.80 | 14.93 |
| RouterDC | 78.95±5.46 | 8.77±0.36 | 55.65±1.86 | 27.65±1.30 | 18.35±8.02 | 85.28±0.22 | 4.75±0.24 | 61.71±22.90 | 30.16±20.00 | 13.61±0.15 | 87.48±0.19 | 7.22±0.22 | 63.21±30.19 | 27.50±25.39 | 13.98±0.18 |
| NIRTRouter | 81.15±0.83 | 6.99±0.22 | 55.64±1.77 | 27.67±1.17 | 14.56±0.14 | 86.91±5.79 | 2.15±0.22 | 82.72±0.93 | 43.72±4.81 | 11.51±5.78 | 90.17±2.21 | 5.12±1.75 | 89.18±0.00 | 46.38±0.00 | 10.65±3.17 |
| RouteLLM | 81.70±1.56 | 3.62±3.15 | 55.64±1.77 | 27.67±1.17 | 12.82±1.79 | 86.85±6.55 | 1.96±2.21 | 82.72±0.93 | 43.72±4.81 | 11.03±5.91 | 90.61±0.00 | 2.37±0.52 | 89.18±0.00 | 46.38±0.00 | 9.35±1.86 |
| StackedGen | 82.66±0.25 | 2.21±0.49 | 59.06±0.38 | 30.91±0.50 | 11.56±0.18 | 87.09±0.69 | 1.99±0.54 | 82.24±0.38 | 42.17±1.58 | 10.60±0.79 | 90.35±0.11 | 2.19±0.87 | 89.04±0.18 | 46.05±0.26 | 8.85±0.05 |
| WALLA I | 81.17±0.29 | 7.59±0.36 | 55.26±0.55 | 24.17±1.16 | 14.86±0.47 | 86.78±0.49 | 3.34±0.35 | 82.60±0.62 | 45.11±4.05 | 11.89±0.43 | 89.89±0.31 | 6.24±0.39 | 86.76±0.97 | 44.90±1.38 | 11.06±0.14 |
| WALLA II | 81.00±0.34 | 7.85±0.61 | 55.30±0.75 | 21.95±1.94 | 15.36±0.51 | 86.98±0.54 | 3.01±0.38 | 82.52±0.52 | 40.42±7.92 | 11.70±0.49 | 90.28±0.24 | 6.17±0.81 | 88.07±0.75 | 42.23±7.02 | 10.59±0.11 |
| WALLA I (log-pool) | 80.32±0.10 | 2.07±0.10 | 55.01±0.22 | 23.15±0.24 | 14.58±0.11 | 87.36±0.05 | 2.88±0.33 | 72.32±1.99 | 25.55±4.73 | 10.48±0.01 | 90.65±0.08 | 2.96±0.19 | 79.94±1.26 | 30.41±3.84 | 9.16±0.03 |
| WALLA I (calibrated) | 82.53±0.05 | 10.56±0.13 | 65.98±0.78 | 38.92±0.94 | 11.69±0.09 | 87.68±0.08 | 8.21±0.12 | 70.71±0.18 | 41.02±0.60 | 9.99±0.07 | 89.59±0.07 | 8.64±0.37 | 78.77±0.20 | 50.24±0.32 | 9.84±0.04 |

*Table 10.* 2 Heterogeneous models (Llama3.1 and Aloe) without context on three datasets. MedMCQA and MMLU are in-distribution; ARC-Challenge is out-of-distribution, with MMLU as the training set. This table filters out data samples where both models answer correctly.

| | MedMCQA (i.d.) | | | | | MMLU (i.d.) | | | | | ARC-Challenge (o.o.d.) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ |
| Llama3.1 | 26.38 | 30.11 | 77.69 | 10.78 | 24.86 | 26.94 | 38.89 | 81.74 | 26.94 | 21.41 | 27.85 | 38.59 | 80.37 | 21.48 | 24.50 |
| Aloe | 29.20 | 36.31 | 72.31 | -10.78 | 32.14 | 19.79 | 54.53 | 68.26 | -26.94 | 41.29 | 24.16 | 50.18 | 69.63 | -21.48 | 39.75 |
| RouterDC | 36.44±5.23 | 16.94±6.40 | 72.11±2.45 | -11.55±9.82 | 20.66±1.91 | 27.81±0.17 | 34.39±0.46 | 73.65±9.16 | -5.39±36.64 | 22.97±0.17 | 33.96±0.19 | 29.69±0.15 | 73.93±7.30 | -4.30±29.21 | 22.71±0.17 |
| NIRTRouter | 36.34±5.34 | 17.10±6.61 | 72.04±3.38 | -11.84±13.53 | 20.73±1.92 | 28.68±5.62 | 33.36±1.67 | 81.28±5.82 | 25.11±23.28 | 22.06±4.98 | 33.56±0.00 | 28.53±0.01 | 80.37±0.00 | 21.48±0.00 | 22.30±0.02 |
| RouteLLM | 35.78±2.09 | 18.07±2.44 | 72.04±3.38 | -11.84±13.53 | 21.03±1.65 | 29.49±2.49 | 32.43±1.57 | 81.28±5.82 | 25.11±23.28 | 20.73±2.98 | 34.90±0.00 | 26.61±0.02 | 80.37±0.00 | 21.48±0.00 | 21.33±0.04 |
| StackedGen | 38.65±1.16 | 20.23±1.00 | 78.26±0.29 | 13.02±1.15 | 13.02±1.15 | 29.80±0.60 | 32.88±0.88 | 79.54±0.40 | 18.17±1.61 | 20.44±0.67 | 35.65±0.80 | 29.35±1.20 | 80.29±0.15 | 21.14±0.62 | 20.42±0.10 |
| UniformAvg | 36.17 | 17.01 | 77.69 | 0.00 | 20.68 | 27.87 | 34.11 | 81.74 | 0.00 | 22.88 | 33.89 | 31.10 | 80.37 | 0.00 | 22.62 |
| PackLLM | 35.40 | 18.32 | 73.39 | -6.45 | 21.08 | 28.08 | 33.83 | 78.34 | 13.37 | 22.79 | 33.22 | 31.28 | 73.32 | -6.71 | 23.04 |
| SelfCertainty | 36.03 | 23.67 | 72.61 | -9.55 | 22.65 | 27.59 | 40.88 | 67.33 | -30.68 | 28.38 | 33.89 | 34.01 | 69.46 | -22.15 | 25.80 |
| Variant I | 36.56±1.20 | 16.28±1.50 | 75.88±0.87 | 3.50±3.47 | 20.64±0.19 | 29.37±0.81 | 31.47±1.36 | 81.04±0.99 | 24.17±3.96 | 20.85±0.66 | 34.40±0.53 | 26.95±0.46 | 80.16±0.26 | 20.64±1.02 | 21.00±0.03 |
| Variant II | 36.55±0.88 | 16.34±0.47 | 75.67±2.61 | 2.68±10.45 | 20.64±0.28 | 29.12±0.75 | 31.71±0.88 | 81.01±0.84 | 24.03±3.37 | 21.00±0.85 | 34.06±0.69 | 26.87±0.74 | 80.20±0.38 | 20.81±1.51 | 21.03±0.09 |
| Variant I (log-pooling) | 36.06±0.58 | 19.84±1.13 | 75.57±3.00 | 2.27±11.99 | 21.32±0.62 | 28.31±0.31 | 34.79±0.69 | 80.67±0.64 | 22.67±2.56 | 22.92±0.97 | 33.05±1.11 | 31.67±2.59 | 79.70±0.58 | 18.79±2.31 | 22.31±0.29 |
| Variant I (calibrated) | 38.92±0.55 | 11.68±1.05 | 74.77±1.09 | -0.91±4.38 | 18.42±0.07 | 30.05±0.20 | 24.90±0.29 | 76.88±0.21 | 7.54±0.85 | 18.15±0.09 | 33.89±0.44 | 26.35±0.62 | 76.26±0.27 | 5.03±1.07 | 19.98±0.08 |

*Table 11.* Comparison of the average wagers and net payouts between (1) when no model is calibrated and (2) when Gemma2 is calibrated, evaluated under Scenario II on MedMCQA and MMLU. We can see that when Gemma2 is calibrated, it gains better net payout. On MedMCQA, as it performs not as well as Llama3.1 and Aloe, it reduces the wager, while on MMLU, it increases the wager, as it is the best performing among all participants. Both wagers and net payouts $\times 100$.

| | MedMCQA | | | | MMLU | | | |
|---|---|---|---|---|---|---|---|---|
| | **All Uncalibrated** | | **Gemma2 Calibrated** | | **All Uncalibrated** | | **Gemma2 Calibrated** | |
| **Method** | Wagers | Net Payouts | Wagers | Net Payouts | Wagers | Net Payouts | Wagers | Net Payouts |
| **Gemma2** | 5.87±0.29 | -0.93±0.05 | 5.13±0.42 | -0.45±0.05 | 16.30±8.74 | -0.52±1.19 | 16.62±5.81 | -0.28±0.67 |
| Llama3.1 | 5.87±0.29 | 0.23±0.02 | 8.45±0.26 | 0.09±0.01 | 8.59±3.21 | -0.30±0.31 | 8.29±4.03 | -0.39±0.38 |
| Aloe | 10.95±0.07 | 0.26±0.02 | 9.38±0.35 | 0.10±0.02 | 7.96±2.96 | -0.45±0.29 | 7.80±2.98 | -0.52±0.33 |
| BioMistral | 2.96±0.17 | -0.54±0.03 | 2.36±0.11 | -0.46±0.02 | 4.76±1.41 | -0.69±0.25 | 4.14±1.40 | -0.62±0.24 |

*Table 12.* Heterogeneous models without context on three datasets. MedMCQA and MMLU are in-distribution; ARC-Challenge is out-of-distribution, with MMLU as the training set. This table filters out data samples where both models answer correctly.

| | MedMCQA (i.d.) | | | | | MMLU (i.d.) | | | | | ARC-Challenge (o.o.d.) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ |
| Gemma2 | 43.25 | 40.85 | 56.83 | -8.50 | 71.61 | 70.20 | 21.74 | 76.62 | 21.61 | 29.71 | 78.04 | 17.82 | 83.15 | 28.84 | 24.36 |
| Aloe | 69.05 | 7.86 | 60.32 | 16.12 | 23.26 | 54.05 | 26.10 | 50.61 | -3.01 | 47.75 | 54.89 | 24.89 | 48.85 | -5.09 | 52.09 |
| Llama3.1 | 67.82 | 2.57 | 56.19 | 11.84 | 22.25 | 58.15 | 15.59 | 43.17 | 0.69 | 37.34 | 57.09 | 15.80 | 39.87 | -2.76 | 43.73 |
| BioMistral | 19.64 | 38.15 | 41.48 | -19.47 | 78.88 | 21.09 | 45.46 | 38.26 | -19.29 | 82.59 | 23.35 | 42.36 | 35.84 | -20.99 | 84.41 |
| RouterDC | 67.42±3.61 | 9.29±0.91 | 59.79±0.83 | 30.02±1.96 | 27.16±8.06 | 65.78±0.51 | 8.28±0.51 | 59.41±18.55 | 21.66±13.68 | 29.83±0.32 | 71.26±0.43 | 12.43±0.47 | 60.92±25.76 | 22.91±19.13 | 30.36±0.38 |
| NIRTRouter | 70.82±3.04 | 6.77±0.10 | 59.79±0.81 | 30.03±1.91 | 21.47±2.98 | 70.08±7.19 | 4.81±0.11 | 76.51±1.37 | 32.92±8.11 | 25.38±7.51 | 77.45±5.07 | 8.87±0.99 | 83.15±0.00 | 39.79±0.00 | 23.58±6.45 |
| RouteLLM | 71.67±1.79 | 2.09±4.02 | 59.79±0.81 | 30.03±1.91 | 18.77±0.39 | 69.95±8.91 | 8.92±4.87 | 76.51±1.37 | 32.92±8.11 | 24.56±7.84 | 78.44±0.00 | 6.28±3.05 | 83.15±0.00 | 39.79±0.00 | 20.90±3.11 |
| StackedGen | 73.13±0.30 | 1.47±0.59 | 64.16±0.20 | 33.91±0.07 | 17.09±0.16 | 70.59±1.23 | 8.42±1.15 | 76.20±0.58 | 32.87±1.56 | 23.62±1.43 | 77.84±0.26 | 7.41±1.70 | 82.96±0.31 | 39.40±0.18 | 19.97±0.10 |
| UniformAvg | 63.97 | 8.20 | 58.06 | 0.00 | 30.79 | 65.48 | 9.12 | 76.33 | 0.00 | 30.32 | 71.06 | 12.69 | 82.58 | 0.00 | 30.86 |
| PackLLM | 67.87 | 10.75 | 55.85 | 21.00 | 27.51 | 64.99 | 8.92 | 41.64 | -15.83 | 31.16 | 68.46 | 11.31 | 38.52 | -16.10 | 32.40 |
| SelfCertainty | 60.87 | 9.05 | 71.25 | 19.89 | 33.34 | 68.81 | 7.37 | 78.04 | 27.88 | 27.03 | 75.25 | 7.46 | 83.53 | 36.66 | 24.16 |
| Variant I | 70.82±0.24 | 7.38±0.43 | 59.90±0.60 | 28.92±0.64 | 21.93±0.41 | 69.88±1.10 | 6.51±1.01 | 76.32±0.35 | 32.16±0.77 | 26.00±0.55 | 76.80±0.70 | 10.87±1.65 | 80.47±0.97 | 38.59±0.85 | 24.24±0.28 |
| Variant II | 70.56±0.35 | 7.62±0.78 | 59.10±0.61 | 26.11±0.83 | 22.65±0.42 | 70.32±1.11 | 6.20±0.94 | 76.41±0.56 | 29.48±4.68 | 25.63±0.70 | 77.69±0.54 | 11.16±2.10 | 81.94±0.85 | 36.48±5.60 | 23.26±0.32 |
| Variant I (log-pooling) | 69.89±0.16 | 1.26±0.14 | 59.32±0.08 | 28.42±0.36 | 21.81±0.17 | 70.63±0.11 | 10.02±0.62 | 70.92±1.48 | 23.15±2.32 | 24.03±0.02 | 78.54±0.18 | 8.91±0.49 | 75.94±1.27 | 28.08±2.08 | 20.90±0.07 |
| Variant I (calibration) | 73.27±0.07 | 10.17±0.14 | 67.35±0.52 | 34.12±0.52 | 16.57±0.10 | 71.36±0.19 | 9.27±0.17 | 70.23±0.10 | 21.01±0.13 | 31.58±0.36 | 76.10±0.16 | 13.39±0.76 | 76.92±0.20 | 38.81±0.48 | 21.06±0.10 |

*Table 13.* 4 Heterogeneous models with private contexts on the PubMedQA dataset. For each question, we randomly assign one model with a relevant context (left columns) or two models, each with a relevant and an irrelevant context (right columns). This table filters out data samples where all 4 models answer correctly.

| | 1 Relevant Context | | | | | | 1 Relevant Context & 1 Irrelavent Context | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ | AUC ↑ | ACC ↑ | ECE ↓ | MRR ↑ | K-Tau ↑ | DRegret ↓ |
| RouterDC | 62.16±1.03 | 67.18±1.76 | 6.04±2.34 | 54.05±21.81 | 7.18±13.04 | 29.58±0.83 | 65.03±1.62 | 68.80±3.66 | 8.11±4.30 | 50.04±1.33 | -0.76±14.62 | 31.39±0.78 |
| NIRTRouter | 62.06±2.79 | 67.69±1.18 | 6.56±8.74 | 46.88±40.59 | 0.07±83.43 | 29.39±2.96 | 65.08±1.26 | 69.42±0.00 | 8.77±3.77 | 45.02±68.41 | -2.09±81.12 | 31.17±1.96 |
| RouteLLM (without context) | 62.54±11.96 | 66.78±0.29 | 2.74±1.75 | 54.72±32.92 | 7.50±6.67 | 28.87±3.44 | 64.62±21.30 | 69.93±4.44 | 2.90±1.45 | 61.40±4.40 | 12.26±14.45 | 28.98±10.12 |
| RouteLLM (with context) | 70.38±0.62 | 70.75±0.82 | 2.97±0.68 | 62.28±1.39 | 19.27±0.41 | 23.45±0.67 | 70.39±0.59 | 72.52±0.80 | 3.13±0.81 | 63.46±2.28 | 17.10±1.38 | 25.22±0.57 |
| StackedGen | 76.58±0.33 | 73.43±0.28 | 4.24±0.29 | 69.19±0.08 | 21.21±0.19 | 19.02±0.23 | 68.86±0.87 | 73.40±0.52 | 5.36±0.66 | 68.71±1.15 | 17.02±1.13 | 25.47±0.70 |
| UniformAvg | 62.26 | 67.04 | 5.93 | 61.04 | 0.00 | 29.64 | 65.29 | 68.43 | 7.61 | 65.47 | 0.00 | 31.42 |
| PackLLM | 73.28 | 77.04 | 11.34 | 69.89 | 22.08 | 20.68 | 66.20 | 71.43 | 9.66 | 59.43 | 7.04 | 29.60 |
| SelfCertainty | 64.02 | 68.15 | 9.67 | 80.56 | 25.57 | 29.57 | 63.05 | 64.45 | 16.27 | 79.63 | 25.50 | 38.47 |
| Variant I | 73.49±0.87 | 74.95±0.26 | 7.64±0.59 | 73.65±0.30 | 24.48±0.63 | 20.75±0.20 | 73.50±0.43 | 76.51±0.29 | 9.52±0.38 | 75.21±0.55 | 25.33±0.41 | 23.09±0.06 |
| Variant II | 73.68±0.31 | 75.07±0.43 | 7.31±0.68 | 73.43±0.07 | 23.53±0.64 | 20.45±0.14 | 72.93±0.05 | 76.18±0.29 | 8.45±0.47 | 75.29±0.59 | 22.84±0.58 | 23.05±0.08 |
| Variant I (log-pooling) | 72.53±0.76 | 74.24±0.23 | 3.36±0.86 | 73.59±0.27 | 24.61±0.72 | 20.32±0.36 | 73.30±0.48 | 76.57±0.23 | 9.60±0.39 | 75.22±0.59 | 25.37±0.24 | 23.12±0.13 |
| Variant I (calibrated) | 83.43±0.32 | 80.65±0.20 | 10.01±0.24 | 78.36±0.10 | 30.74±0.30 | 15.18±0.08 | 83.86±0.28 | 84.08±0.06 | 10.56±0.13 | 80.52±0.19 | 32.92±0.31 | 13.40±0.10 |