Active Model Selection for Large Language Models

Yavuz Durmazkeser
TU Delft
y.durmazkeser@tudelft.nl

Patrik Okanovic
ETH Zurich
patrik.okanovic@inf.ethz.ch

Andreas Kirsch
blackhc@gmail.com

Torsten Hoefler
ETH Zurich
htor@ethz.ch

Nezihe Merve Gürel
TU Delft
n.m.gurel@tudelft.nl

Abstract

We introduce LLM SELECTOR, the first framework for active model selection of Large Language Models (LLMs). Unlike prior work on evaluation and benchmarking, LLM SELECTOR addresses the challenge of selecting the best LLM for a given task *under limited annotation budgets*. In particular, LLM SELECTOR adaptively identifies a small set of informative queries to annotate in order to efficiently select the best LLM for the given task. To further reduce annotation cost, we leverage a judge-based annotation model using an oracle. Through extensive experiments, we show that LLM SELECTOR reduces annotation costs by up to 58.33% for identifying the best model, and by 62.50% when selecting a near-best LLM that is within close vicinity of best model.

1 Introduction

How can we select the best Large Language Model (LLM) for a given application or data distribution without retraining? Answering this question has become increasingly difficult as the number of readily available models continues to expand. Recent advances in architectures, training strategies, and access to massive datasets have enabled impressive zero-shot capabilities, allowing LLMs to perform a wide range of tasks without task-specific fine-tuning [72, 34]. As a result of these developments, a large and diverse collection of pretrained models differing in architecture, training data, and optimization objectives is now easily accessible through academic repositories and commercial platforms [23, 54, 21, 3].

While the abundance of LLMs offers deployment flexibility, it also poses challenges as performance varies widely across domains, tasks, and languages [42]. Despite extensive benchmarking efforts [42, 17, 56], the rapid growth of models and evaluation settings outpaces existing practices, which often lag behind release cycles or focus narrowly on standardized tasks. Model selection typically relies on randomly or heuristically chosen annotated subsets [61, 71], but these waste resources and fail to capture model differences [47, 36]. Active selection [48] addresses this by strategically annotating subsets, yet prior work largely targets classification, not generative tasks [53, 33, 48, 30, 60, 45, 31, 73, 38, 39]. Thus, reliably identifying the best LLM for a task and distribution under limited annotations remains open.

In this work, we address this problem and ask: *Given a pool of queries and a set of candidate LLMs, which examples should be annotated in order to reliably identify the best LLM, both in a model-agnostic and annotation-efficient manner?*

Contributions: In this paper, we introduce LLM SELECTOR, a principled framework for selecting the best LLM under a limited annotation budget. Given a large set of n queries and a limited annotation budget b with $b \ll n$, LLM SELECTOR selects b queries whose annotations are expected

39th Conference on Neural Information Processing Systems (NeurIPS 2025)
Workshop: Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling.

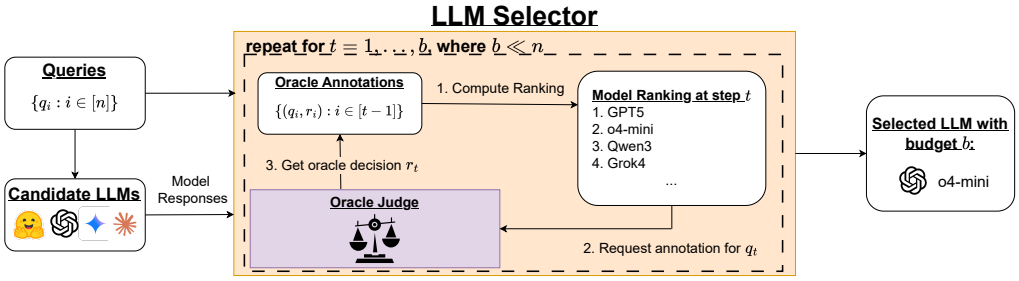

Figure 1: An overview of LLM SELECTOR. For an arbitrary pool of n LLM queries and a set of candidate models, LLM SELECTOR adaptively picks and annotates most informative $b \ll n$ queries for identifying the best language model.

to maximally reduce uncertainty about the best model under the budget constraint. Our approach builds on information-gain criteria [10], and quantifies informativeness using a two-parameter model that measures information gain as Shannon's mutual information between the unknown best model and annotations.

We employ a judge-based annotation process in which each query is annotated with a vector over candidate models. For each model candidate, we compare its response to the query against that of a baseline model using oracle preference judgments. This judge-based design alleviates the need for costly reference answers or summaries that are known to be far more expensive than pairwise judgments [81, 57, 62, 46, 8], and mitigates the noise commonly introduced by reference-based evaluation metrics [81, 57, 62, 52].

We validate LLM SELECTOR through an extensive set of experiments across diverse tasks, including long-form question answering, code generation, summarization, translation, and mathematics. Across all settings, LLM SELECTOR consistently identifies the best-performing model with only a small fraction of the annotation budget required by baseline selection strategies, demonstrating both efficiency and reliability under constrained annotation resources. LLM SELECTOR reduces annotation costs by up to 58.33% while achieving a 57.55% reduction when selecting models within a 1% win-rate vicinity of the best model.

Once the best LLM is selected based on b annotated queries, we use it to generate outputs for the remaining $n-b$ queries where $n-b \gg b$. Our method is fully model-agnostic: it requires no access to internal parameters and imposes no restrictions on output format, making it directly applicable in black-box or API-only settings. An overview of LLM SELECTOR is shown in Figure 1.

2 LLM SELECTOR

In this section, we introduce LLM SELECTOR. Section 2.1 defines the problem, followed by our preference-based annotation framework in Section 2.2. Section 2.3 presents the active selection algorithm.

2.1 Problem Setting

We consider inference-time with a set of n unannotated queries $Q = \{q_i \in \mathcal{Q} \mid i \in [n]\}$, where each query q_i is a user-issued prompt to an oracle. The oracle response is $r_i \in \mathcal{R}$, unobserved at inference, and denoted by the random variable R_i.

Given m pretrained models $\mathcal{M} = \{h_j : \mathcal{Q} \to \mathcal{R} \mid j \in [m]\}$, our goal is to identify the best model for generating responses to Q. Since oracle annotations are costly, we assume access to only $b \ll n$ annotations. The task is therefore to select b queries whose annotations provide maximal information about the best model.

We define the best model h^* as the one with highest utility if all responses $\{r_i \mid i \in [n]\}$ were known, and denote it as the random variable H. The utility function can be any metric that evaluates model quality, such as accuracy on a dataset or other established LLM evaluation measures. Once h^* is identified, it is used to generate responses for the remaining $n-b$ queries.

Formally, we pose selection as maximizing mutual information. We seek a subset $\mathcal{A} \subseteq \{(q_i, r_i) \mid i \in [n]\}$ of at most b annotated examples that maximizes the mutual information between H and $\mathcal{A}$:

$$\mathcal{A}_{\text{opt}}[b] = \underset{\substack{\mathcal{A} \subseteq \{(q_i, r_i) \mid i \in [n]\} \\ |\mathcal{A}| \leq b}}{\arg\max} \mathbb{I}(H; \mathcal{A}). \tag{1}$$

2.2 Annotation via Direct Preference Judgments

Evaluating long-form responses cannot rely on exact string matching and requires more sophisticated methods. Beyond correctness, aspects such as relevance, helpfulness, complexity, and detail affect answer quality. Since reference-based metrics yield noisy scores [52, 8], we instead use direct preference judgments [79, 62], which compare responses pairwise and are more stable than individual ratings [29, 81]. Preference-based evaluation is already widely adopted in open-ended LLM benchmarks [79, 40, 41].

Formally, for a query $q_i \in Q$, an *oracle judge* compares responses of models h_j and h_k, where $h_j(q_i) > h_k(q_i)$ means h_j's response is preferred, $h_j(q_i) < h_k(q_i)$ means h_k's response is preferred, and $h_j(q_i) = h_k(q_i)$ means the responses are judged equally good (or poor).

The oracle judgment is expressed as $\text{OracleJudge}(q_i, h_j(\cdot), h_k(\cdot)) = \mathbb{1}[h_j(q_i) > h_k(q_i)] + \frac{1}{2} \cdot \mathbb{1}[h_j(q_i) = h_k(q_i)]$, where $\mathbb{1}[\cdot]$ is the indicator function.

To compare models across queries, we adopt the *win rate* metric [41]. For query set Q, the win rate of h_j over h_k is defined as $\text{WR}_Q(h_j, h_k) = \frac{1}{n} \sum_{i=1}^{n} \text{OracleJudge}(q_i, h_j(\cdot), h_k(\cdot))$, with $\text{WR}_Q(h_j, h_k) + \text{WR}_Q(h_k, h_j) = 1$ for $j, k \in [m]$.

Since full pairwise ranking requires $\mathcal{O}(m^2)$ annotations, we reduce cost by comparing all models against a single *baseline model* $\bar{h} \in \mathcal{M}$, which has become standard practice in LLM evaluation benchmarks [40]. Each model is evaluated by its win rate relative to $\bar{h}$, and LLM SELECTOR selects the one with the highest win rate on the annotated queries.

To characterize mutual information between the unknown best model and annotations, we use a two-parameter model describing the relation between the best model and the baseline under oracle preference:

$$\begin{aligned} &\mathbb{P}(H(q) < \bar{h}(q) \mid H = h^*) = \epsilon_1, \qquad &&\mathbb{P}(H(q) = \bar{h}(q) \mid H = h^*) = \epsilon_2, \\ &\mathbb{P}(H(q) > \bar{h}(q) \mid H = h^*) = 1 - \epsilon_1 - \epsilon_2 \end{aligned} \tag{2}$$

where $\epsilon_1, \epsilon_2 \in [0, 1]$ and $\epsilon_1 + \epsilon_2 \leq 1$.

2.3 The Algorithm

Given the query set Q, our goal is to select at most b queries whose annotations maximize information about the best LLM as defined in equation 1. We adopt a sequential information maximization strategy [10], selecting queries one at a time until the budget b is exhausted.

At step $t = 0$, the unannotated set is $U_0 = Q$ and the annotated set is $A_0 = \emptyset$. More generally, at step t, U_t denotes the pool of unannotated queries and A_t the annotated ones so far. The next query q_t is chosen as

$$\begin{aligned} q_t &= \underset{q \in U_t}{\arg\max}\, \mathbb{I}(H; R \mid A_t, q) = \underset{q \in U_t}{\arg\max}\, \mathbb{H}(H \mid A_t) - \mathbb{E}_R[\mathbb{H}(H \mid A_t \cup \{(q, R)\})] \\ &= \underset{q \in U_t}{\arg\min}\, \mathbb{E}_R[\mathbb{H}(H \mid A_t \cup \{(q, R)\})], \end{aligned} \tag{3}$$

where $\mathbb{H}(H \mid A_t)$ is the conditional entropy of H given the observed annotations at step t.

Thus, selecting the next query amounts to minimizing the expected conditional entropy of H as in equation 3. Since oracle responses are unavailable, this expectation is approximated using noisy annotations from *weak judges*.

2.3.1 Noisy Annotations via Weak Judges

The intuition behind noisy annotations is to evaluate a candidate response by comparing it to other model responses, assigning higher preference to those more similar to others.

Formally, each response is tokenized as $(w_1, \ldots, w_L)$. For $k \in \mathbb{N}$, we build a k-gram language model, where $\mathbb{P}(w_l | w_{1:L}) := \mathbb{P}(w_l | w_{l-k+1:l-1})$. The model is fit independently for each query q

using responses of candidate models M. The likelihood of a sequence is the average of its word probabilities: $\mathbb{P}((w_1, \ldots, w_L)) = \frac{1}{L} \sum_{l=1}^{L} \mathbb{P}(w_l | w_{l-k+1:l-1})$.

To compare models h and $\bar{h}$, a weak judge selects the response with higher likelihood, yielding $h(q) >_{(k)} \bar{h}(q)$, $h(q) =_{(k)} \bar{h}(q)$, or $h(q) <_{(k)} \bar{h}(q)$. We denote this noisy annotation as $r_{(k)}$. Given the parameter model in equation 2, the following probability is used to compute the information gain:

$$\mathbb{P}(H = h_j | A_t \cup \{(q, r_{(k)})\}) \propto \epsilon_1^{\mathbb{1}[h_j(q) <_{(k)} \bar{h}(q)]} \epsilon_2^{\mathbb{1}[h_j(q) =_{(k)} \bar{h}(q)]} (1 - \epsilon_1 - \epsilon_2)^{\mathbb{1}[h_j(q) >_{(k)} \bar{h}(q)]} \mathbb{P}(H = h_j | A_t).$$

With $z \geq 1$ weak judges, each using a k-gram model ($k \in [z]$), the expected entropy is computed by averaging weak judge estimates: $q_t = \arg\min_{q \in U_t} \frac{1}{z} \sum_{k=1}^{z} \mathbb{H}(H \mid A_t \cup \{(q, r_{(k)})\})$, using a uniform distribution over judges.

2.3.2 Updating Model Posterior Belief

After annotating the selected query at step t, we update the posterior belief over the best language model conditioned on all annotations observed up to time t: $\mathbb{P}(H = h_j | A_t \cup \{(q, R = r)\}) \propto \mathbb{P}(A_t \cup \{(q, R = r)\} | H = h_j) \cdot \mathbb{P}(H = h_j)$.

With the two-parameter annotation model in equation 2, the posterior belief is updated as:

$$\begin{aligned} \mathbb{P}(H = h_j | A_{t+1}) &\propto \mathbb{P}(H(q_t) = r_t | H = h_j) \cdot \mathbb{P}(H = h_j | A_t) \\ &\propto \epsilon_1^{\mathbb{1}[h_j(q_t) = \bar{h}(q_t)]} \epsilon_2^{\mathbb{1}[h_j(q_t) > \bar{h}(q_t)]} (1 - \epsilon_1 - \epsilon_2)^{\mathbb{1}[h_j(q_t) > \bar{h}(q_t)]} \mathbb{P}(H = h_j | A_t) \quad (4) \end{aligned}$$

Further details on selecting ϵ_1 and ϵ_2, as well as the complete pseudocode of LLM SELECTOR, are provided in Appendices D and F, respectively.

3 Experiments

We evaluate our approach on three datasets: AlpacaEval [41], Arena-Hard [40], and MT-Bench [79], and compare against baselines including `Random`, `Bradley-Terry`, `Confidence`, `Uncertainty`, and `Most Draws`. Further details on the datasets, models and baselines are provided in Appendices B and C.

In our experiments, we uniformly sample a pool of n examples from the test set. Model selection strategies then choose b queries to annotate from the pool, and the LLM with the highest average utility on these annotations is selected. We call this a *realization*, and evaluate each strategy over multiple realizations to estimate performance.

We compare strategies by three metrics. *Identification probability* is the fraction of realizations that correctly identify the best model for budget b, reported for $b = 1, \ldots, n$. *Label efficiency* measures the percentage reduction in labels required to identify the best model or reach within δ of it. *95th Percentile Win Rate Gap* is the 95th percentile of the win rate difference between the selected model and the best model.

Identification Probability Figure 2 shows the best model identification probability of LLM SELECTOR and baselines. LLM SELECTOR consistently outperforms baselines, identifying the best LLM with fewer annotations. In contrast, baselines exhibit unstable performance, with probabilities that do not always improve with budget and occasionally fall below `Random`.

Label Efficiency Table 1 shows the label efficiency of LLM SELECTOR to recover the near-best models on all datasets.

Dataset	$\delta = 1\%$	$\delta = 0.5\%$	$\delta = 0.1\%$
AlpacaEval	$\uparrow 11.65\%$	$\uparrow 4.59\%$	$\uparrow 8.87\%$
Arena-Hard	$\downarrow$ **57.55**%	$\downarrow$ **62.50**%	$\downarrow$ **53.57**%
MT-Bench	$\downarrow$ **50.00**%	$\downarrow$ **57.14**%	$\downarrow$ **50.00**%

Table 1: Label efficiency for near-best models across datasets. Bolded numbers with $\downarrow$ indicate decreases.

Robustness Analysis Additionally, we have analyzed the 95th percentile win rate gap between the chosen and the best LLM (see Appendix E). This shows that LLM SELECTOR consistently returns a near-best model.

4 Conclusion

We introduced LLM SELECTOR, a framework for active model selection that identifies the best LLM under limited annotation budgets. By leveraging preference-based judgments and information-theoretic query selection our method significantly reduces annotation costs. Overall, LLM SELEC-

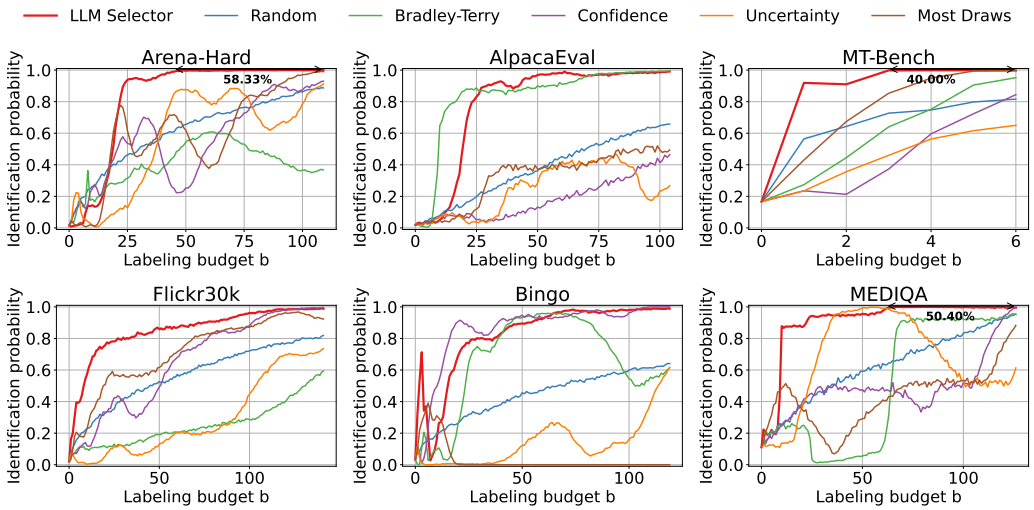

Figure 2: Best model identification probability of LLM SELECTORand the baselines on Arena-Hard (left), AlpacaEval (middle), and MT-Bench (right).

TOR provides a principled and efficient solution for reliable LLM selection in resource-constrained settings.

References

[1] Alnur Ali, Rich Caruana, and Ashish Kapoor. Active learning with model selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[3] Anthropic. Anthropic claude api. `https://console.anthropic.com`.

[4] Anthropic. Model card and evaluations for claude models, 2023. URL `https://www.anthropic.com/news/claude-2`.

[5] Anthropic. Introducing the next generation of claude, 2024. URL `https://www.anthropic.com/news/claude-3-family`.

[6] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[7] Ralph Allan Bradley and Milton E Terry. The rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[8] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of Bleu in machine translation research. In Diana McCarthy and Shuly Wintner (eds.), *11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 249–256, Trento, Italy, April 2006. Association for Computational Linguistics. URL `https://aclanthology.org/E06-1032/`.

[9] Francois Caron and Arnaud Doucet. Efficient Bayesian Inference for Generalized Bradley-Terry Models. *Journal of Computational and Graphical Statistics*, 21(1):174–196, 2012. URL `https://inria.hal.science/inria-00533638`.

[10] Yuxin Chen, S. Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In Peter Grünwald, Elad Hazan, and Satyen Kale (eds.), *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 338–363, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Chen15b.html.

[11] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[12] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[13] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[14] Databricks. Introducing dbrx: A new state-of-the-art open llm. https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm, March 2024. Accessed: 2025-08-31.

[15] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[16] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: general language model pretraining with autoregressive blank infilling. pp. 320–335, 2022.

[17] Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

[18] Jacob Gardner, Gustavo Malkomes, Roman Garnett, Kilian Q Weinberger, Dennis Barbour, and John P Cunningham. Bayesian active model selection with an application to automated audiometry. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/d9731321ef4e063ebbee79298fa36f56-Paper.pdf.

[19] Google. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[20] Google. Gemma open models, 2024. URL https://ai.google.dev/gemma.

[21] Google DeepMind. Gemini api. https://ai.google.dev.

[22] Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022. doi: 10.1162/tacl_a_00474. URL https://aclanthology.org/2022.tacl-1.30/.

[23] Hugging Face. Hugging face hub. https://huggingface.co.

[24] InternLM. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM-techreport, 2023.

[25] Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2, 2023.

[26] Kevin G. Jamieson and Robert D. Nowak. Active ranking using pairwise comparisons. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'11, pp. 2240–2248, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

[27] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[28] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[29] Nicolas Jones, Armelle Brun, and Anne Boyer. Comparisons instead of ratings: Towards more stable preferences. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pp. 451–456, 2011. doi: 10.1109/WI-IAT.2011.13.

[30] Mohammad Reza Karimi, Nezihe Merve Gürel, Bojan Karlaš, Johannes Rausch, Ce Zhang, and Andreas Krause. Online active model selection for pre-trained classifiers. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 307–315. PMLR, April 2021.

[31] Parnian Kassraie, Nicolas Emmenegger, Andreas Krause, and Aldo Pacchiano. Anytime model selection in linear bandits. In *Proc. Neural Information Processing Systems (NeurIPS)*, December 2023.

[32] Namit Katariya, Arun Iyer, and Sunita Sarawagi. Active evaluation of classifiers on large datasets. In *2012 IEEE 12th International Conference on Data Mining*, pp. 329–338, 2012. doi: 10.1109/ICDM.2012.161.

[33] Justin Kay, Grant Van Horn, Subhransu Maji, Daniel Sheldon, and Sara Beery. Consensus-driven active model selection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. ICCV 2025 Highlight.

[34] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213, 2022.

[35] Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, pp. 5753–5763. PMLR, 2021.

[36] Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5753–5763. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/kossen21a.html`.

[37] Rui Leite and Pavel Brazdil. Active testing strategy to predict the best classification algorithm via sampling and metalearning. In *ECAI 2010*, pp. 309–314. IOS Press, 2010.

[38] Junfan Li, Zenglin Xu, Zheshun Wu, and Irwin King. On the necessity of collaboration in online model selection with decentralized data. *arXiv preprint arXiv:2404.09494*, 2024.

[39] Po-han Li, Oyku Selin Toprak, Aditya Narayanan, Ufuk Topcu, and Sandeep Chinchali. Online foundation model selection in robotics. *arXiv preprint arXiv:2402.08570*, 2024.

[40] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From live data to high-quality benchmarks: The arena-hard pipeline, April 2024. URL `https://lmsys.org/blog/2024-04-19-arena-hard/`.

[41] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.

[42] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=iO4LZibEqW. Featured Certification, Expert Certification.

[43] Shen Liang, Yanchun Zhang, and Jiangang Ma. Active model selection for positive unlabeled time series classification. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 361–372, 2020. doi: 10.1109/ICDE48307.2020.00038.

[44] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

[45] Xuefeng Liu, Fangfang Xia, Rick L Stevens, and Yuxin Chen. Contextual active online model selection with expert advice. In *ICML2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*. ICML, 2022.

[46] Ge Luo, Hebi Li, Youbiao He, and Forrest Sheng Bao. PrefScore: Pairwise preference learning for reference-free summarization quality assessment. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (eds.), *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 5896–5903, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.515/.

[47] Omid Madani, Daniel J. Lizotte, and Russell Greiner. Active model selection. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pp. 357–365, Arlington, Virginia, USA, 2004. AUAI Press. ISBN 0974903906.

[48] Omid Madani, Daniel J. Lizotte, and Russell Greiner. Active model selection, 2012. URL https://arxiv.org/abs/1207.4138.

[49] Mitsuru Matsuura and Satoshi Hara. Active model selection: A variance minimization approach. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023. URL https://openreview.net/forum?id=vBwfTUDTtz.

[50] Meta AI. Introducing meta llama 3: The most capable openly available llm to date, 2024. URL https://ai.meta.com/blog/meta-llama-3.

[51] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

[52] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for NLG. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2241–2252, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1238. URL https://aclanthology.org/D17-1238/.

[53] Patrik Okanovic, Andreas Kirsch, Jannes Kasper, Torsten Hoefler, Andreas Krause, and Nezihe Merve Gürel. All models are wrong, some are useful: Model selection with limited labels. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2035–2043. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/okanovic25a.html.

[54] OpenAI. Openai api. https://platform.openai.com.

[55] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[56] OpenCompass. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

[57] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[58] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[59] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

[60] Vihari Piratla, Soumen Chakrabarti, and Sunita Sarawagi. Active assessment of prediction services as accuracy surface over attribute combinations. *Advances in Neural Information Processing Systems*, 34:23140–23151, 2021.

[61] Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[62] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

[63] Christoph Sawade, Niels Landwehr, Steffen Bickel, and Tobias Scheffer. Active risk estimation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 951–958, 2010.

[64] Christoph Sawade, Niels Landwehr, and Tobias Scheffer. Active comparison of prediction models. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/92fb0c6d1758261f10d052e6e2c1123c-Paper.pdf.

[65] Abigail See, Peter Liu, and Christopher Manning. Get to the point: Summarization with pointer-generator networks. In *Association for Computational Linguistics*, 2017. URL https://arxiv.org/abs/1704.04368.

[66] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[67] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[68] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[69] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[70] Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=aKkAwZB6JV`.

[71] Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.eacl-long.95/`.

[72] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

[73] Yu Xia, Fang Kong, Tong Yu, Liya Guo, Ryan A. Rossi, Sungchul Kim, and Shuai Li. Convergence-aware online model selection with time-increasing bandits. In *The Web Conference 2024*, 2024. URL `https://openreview.net/forum?id=2IwSOTWvXu`.

[74] Yu Xia, Fang Kong, Tong Yu, Liya Guo, Ryan A Rossi, Sungchul Kim, and Shuai Li. Which llm to play? convergence-aware online model selection with time-increasing bandits. In *Proceedings of the ACM on Web Conference 2024*, pp. 4059–4070, 2024.

[75] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=CfXh93NDgH`.

[76] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

[77] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL `https://arxiv.org/abs/1904.09675`.

[78] Bin Zhao, Fei Wang, Changshui Zhang, and Yangqiu Song. Active model selection for graph-based semi-supervised learning. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1881–1884, 2008. doi: 10.1109/ICASSP.2008.4518001.

[79] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[80] Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. Starling-7b: Improving helpfulness and harmlessness with RLAIF. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=GqDntYTTbk.

[81] Markus Zopf. Estimating summary quality with pairwise preferences. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1687–1696, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1152. URL https://aclanthology.org/N18-1152/.

Supplementary Material for *Active Model Selection for Large Language Models*

A Related Work

Several methodologies exist for the evaluation of LLMs. Traditional multiple-choice [66, 67], or short-answer benchmarks [13] provide a standardized way to evaluate model performance, though they do not assess the generative abilities of LLMs. For tasks such as summarization [65, 51] and translation [22], reference-based benchmarks are commonly used, where model outputs are compared against human-written ground truth using metrics like BLEU [59], ROUGE [44], and BERTScore [77]. More recently, judge-based evaluation gained traction. LMArena [79] is a live leaderboard using human annotators. Static benchmarks like Arena-Hard [40], AlpacaEval [41] and MT-Bench [79] rely on LLM-as-a-Judge for automated evaluation. At a higher level, leaderboards such as HELM [42], OpenCompass [56], and OpenLLM [17] aggregate benchmarks measuring models on different capabilities, aiming to give a full view of LLM capabilities. However, these evaluation methodologies require relying on human annotators or LLM-as-a-Judge, and due to the large scale of modern benchmarks, such evaluations are often not feasible with limited resources.

Many existing studies on model selection under budget constraints concentrate on specific tasks or model families. Okanovic et al. [53], Kay et al. [33] address classification models, while Zhao et al. [78] assume a graph-based semi-supervised learning task. Liang et al. [43] examine binary time-series classification models, and Gardner et al. [18] focus on Gaussian processes. Several studies make less assumptions about the task or the model, focusing on estimating the model risk. These settings consider either a single model [63, 32, 35], two models [64, 37], or an unrestricted number of candidate models [49].

While our work assumes that all unannotated examples are available beforehand, much of the literature adopts an online setting in which data arrive sequentially from a stream [48, 30, 60, 45, 31, 73, 38, 39, 74]. Other approaches jointly train the candidate models during the active selection process [1].

Although prior research has explored efficient active ranking based on comparisons [26, 9], existing methods primarily select pairs of models for evaluation. By contrast, our setup compares models on LLM queries spanning diverse levels of difficulty, where the outcome of the evaluation depends on the query itself. This motivates a data-centric perspective in which we prioritize selecting examples for annotation rather than model pairs.

B Dataset and Model Collections

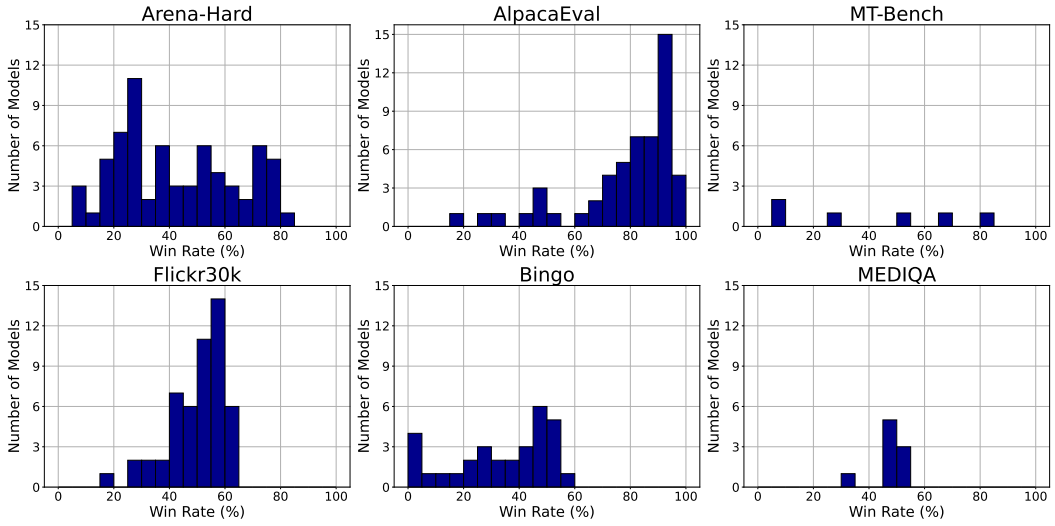

Figure 3: Candidate LLM win rate histograms

We conduct experiments on three datasets: AlpacaEval [41], Arena-Hard [40], and MT-Bench [79]. Candidate models include LLMs from diverse families, including proprietary systems such as GPT-3.5 and GPT-4 [58, 55], Claude 2/3 [4, 5], and Gemini [19], as well as open-weight architectures like LLaMA-2/3 [69, 50], Mistral and Mixtral [27, 28], Falcon [2], Yi [76], Qwen [6], Gemma [20], InternLM [24], GLM [16], and DBRX [14]. We further consider several widely adopted instruction-tuned derivatives, including Alpaca [68], Vicuna [11], Guanaco [15], Tulu-2 [25], WizardLM [75], Zephyr [70], and Starling [80]. The chosen LLMs differ in both number of parameters and training methodology.

Arena-Hard consists of 500 test queries, which are selected from user queries of Chatbot Arena [12]. We use `gpt-4-1106-preview` Judge for annotating model responses, and `gpt-4-0314` as the baseline model. In total, we evaluate 68 LLMs on Arena-Hard.

AlpacaEval is an LLM benchmark consisting of 805 test samples. The responses are evaluated by GPT-4 Judge, and the baseline model is `text_davinci_003`. We test 55 LLMs on AlpacaEval.

MT-Bench contains only multi-turn conversations, making it more challenging than other two benchmarks. However, the size is smaller as it only contains 80 queries. We evaluate 6 LLMs on this benchmark.

The performance of the candidate LLMs is plotted in Figure 3. The plots show the histogram of models which are in the different win rate ranges for each dataset.

We choose the parameters ϵ_1 and ϵ_2 independently for each dataset, based on the procedure described in Section D. We choose the number of weak judges z as 10 in all experiments.

C Baselines

We compare LLM SELECTOR against several selection strategies:

`Random` Test queries are sampled uniformly from the set of unannotated examples.

`Bradley-Terry` Based on judge annotations, Bradley-Terry coefficients [7] are computed to model pairwise preferences. These coefficients are then used to form a model posterior. The next sample is greedily chosen as the one expected to yield the greatest decrease in the entropy of this posterior.

`Most Draws` The queries are sorted according to the number of responses that have drawn against the baseline response. The comparisons are done by the pseudo-judge described in Section D. At each step, unannotated sample with the most number of draws is selected.

`Confidence` For each test sample, the number of responses that win, draw, and lose against the baseline is counted. The counts are then normalized by dividing to the number of responses to compute the outcome frequencies. Test samples are sorted by the entropies of the computed frequencies. The query with the lowest entropy is selected.

`Uncertainty` Same as `Confidence`, we compute outcome frequencies for each sample and their entropies. The query with the highest entropy is selected.

D Parameter Selection

We choose the parameters ϵ_1 and ϵ_2 prior to LLM selection, therefore the oracle annotations are not available during parameter optimization. As a replacement, we use ensemble of all weak judges as a noisy oracle. More specifically, the noisy oracle behaves as follows:

$$\text{WeakJudges}(q, h(q), \bar{h}(q)) = \begin{cases} 1 & \text{if } \nu \geq 2/3 \quad \texttt{// win} \\ 0.5 & \text{if } 2/3 > \nu \geq 1/3 \quad \texttt{// draw} \\ 0 & \text{otherwise} \quad \texttt{// loss} \end{cases}$$

where $\nu = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}[h(q) >_{(k)} \bar{h}(q)] + \frac{1}{2} \cdot \mathbb{1}[h(q) =_{(k)} \bar{h}(q)]$. We take average of all weak judge predictions, and round the average to the closest option in the set $\{0, 0.5, 1\}$.

We perform a grid search over ϵ_1 and ϵ_2 using the weak judge decisions as the ground-truth annotations. We select the parameter set that maximizes the identification probability, defined as the probability of correctly recognizing the best LLM under the budget b.

Dataset Identification probability	LLM SELECTOR (70%/80%/90%/100%)	Random (70%/80%/90%/100%)	Bradley-Terry (70%/80%/90%/100%)	Confidence (70%/80%/90%/100%)	Uncertainty (70%/80%/90%/100%)	Most Draws (70%/80%/90%/100%)
AlpacaEval	6.1/4.6/**3.1**/**0.0**	8.6/8.2/7.9/2.9	**4.2**/**3.9**/3.7/**0.0**	4.5/3.4/3.4/2.3	8.4/8.4/**3.1**/2.7	11.6/11.4/8.6/2.8
Arena-Hard	13.1/**11.8**/**8.1**/**0.0**	13.6/13.4/13.1/8.3	13.3/13.0/13.0/7.9	13.7/13.5/13.1/9.6	14.4/14.1/14.0/6.8	**12.6**/12.9/13.3/7.1
MT-Bench	**12.5**/**12.5**/**12.5**/**0.0**	34.2/34.2/34.2/14.2	33.3/33.3/33.3/**0.0**	34.2/34.2/34.2/**0.0**	34.2/34.2/34.2/15.8	35.3/35.3/35.3/**0.0**

Table 2: 95th percentile win rate gap (%) at budget needed to reach identification probability 70%, 80%, 90%, and 100% on three benchmarks. Best results are in bold; second-best underlined.

E Robustness Analysis

Table 2 show 95th percentile win rate gap between the chosen and the best LLMs on AlpacaEval, Arena-Hard and MT-Bench. The 95th percentile is chosen accross all realizations, where strategies use the budget which allows LLM SELECTOR to reach 70%, 80%, 90% and 100% identification probability.

F Pseudocode

Algorithm 1 LLM SELECTOR Algorithm

Require: models $\mathcal{M}$, test queries $\mathcal{Q}$, parameters $\epsilon_1, \epsilon_2, \epsilon_3$, labeling budget b

$\mathcal{A}_0 \leftarrow \{\}, \mathcal{U}_0 \leftarrow \mathcal{Q}$

`//Uniform model prior`

$\mathbb{P}(H = h^j | \mathcal{A}_0) \leftarrow 1/M$

for $t = 0$ to $b - 1$ **do**

  for $k = 1$ to z **do**

    `//Estimate model posterior with weak judge decisions`

    $\mathbb{P}(H = h^j | \mathcal{A}_t \cup \{(q, r_{(k)})) \leftarrow \frac{1}{Z}\mathbb{P}(H = h^j | \mathcal{A}_t) \cdot \epsilon_1^{\mathbb{1}[h^j(q) <_{(k)} \bar{h}(q)]} \epsilon_2^{\mathbb{1}[h^j(q) =_{(k)} \bar{h}(q)]} \epsilon_3^{\mathbb{1}[h^j(q) >_{(k)} \bar{h}(q)]}$

  end for

  `//Choose the sample with minimum expected entropy`

  $q_t \leftarrow \arg\min_{q \in \mathcal{U}_t} \frac{1}{z} \sum_{k=1}^{z} \mathbb{H}(H | \mathcal{A}_t \cup \{(q, r_{(k)})\})$

  `//Get oracle decision`

  $r_t \leftarrow \text{OracleJudge}(q_t, h_j(\cdot), \bar{h}(\cdot))$

  $\mathcal{A}_{t+1} \leftarrow \mathcal{A}_t \cup \{(q_t, r_t)\}$

  $\mathcal{U}_{t+1} \leftarrow \mathcal{U}_t \setminus \{q_t\}$

  `//Update model posterior`

  $\mathbb{P}(H = h^j | \mathcal{A}_{t+1}) \leftarrow \frac{1}{Z}\mathbb{P}(H = h^j | \mathcal{A}_t) \cdot \epsilon_1^{\mathbb{1}[h^j(q) < \bar{h}(q)]} \epsilon_2^{\mathbb{1}[h^j(q) = \bar{h}(q)]} \epsilon_3^{\mathbb{1}[h^j(q) > \bar{h}(q)]}$

end for

return $\arg\max_{h \in \mathcal{M}} \text{WR}_{\mathcal{A}_b}(h, \bar{h})$