Evaluation of Large Language Models via Coupled Token Generation

Nina Corvelo Benz
MPI of Biochemistry

Stratis Tsirtsis
Hasso Plattner Institute

Eleni Straitouri
MPI for Software Systems

Ivi Chatzi
MPI for Software Systems

Ander Artola Velasco
MPI for Software Systems

Suhas Thejaswi
Aalto University

Manuel Gomez-Rodriguez
MPI for Software Systems

Abstract

State-of-the-art large language models rely on randomization to respond to a prompt. Consequently, a model may respond differently to the same prompt if asked multiple times. In this work, we argue that the evaluation and ranking of large language models should control for this randomization. Our starting point is the development of a causal model for coupled autoregressive generation, which allows different large language models to sample responses with the same source of randomness. Building upon our causal model, we first show that, on evaluations based on benchmark datasets, coupled autoregressive generation leads to the same conclusions as vanilla autoregressive generation but using provably fewer samples. However, we further show that, on evaluations based on pairwise comparisons, the two approaches can surprisingly lead to different rankings when comparing more than two models. To complement our theoretical results, we conduct experiments with several models from the `Llama`, `Mistral` and `Qwen` families. We find that, across multiple benchmark datasets, coupled autoregressive generation requires up to 75% fewer samples to reach the same conclusions as vanilla autoregressive generation. Further, we find that the win-rates derived from pairwise comparisons by a strong large language model to prompts from the LMSYS Chatbot Arena platform differ under coupled and vanilla autoregressive generation.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

1 INTRODUCTION

One of the most celebrated aspects of state-of-the-art large language models (LLMs) is that they can solve open-ended, complex tasks across many different application domains such as coding, healthcare and scientific discovery (Bubeck et al., 2023; Mozannar et al., 2024; Haupt and Marks, 2023; Romera-Paredes et al., 2023). However, this is crucially what also makes the evaluation and comparison of LLMs very challenging—it is very difficult, if not impossible, to create a single benchmark. As a consequence, in recent years, there has been a flurry of papers introducing different benchmarks (Bach et al., 2022; Wei et al., 2022; Talmor et al., 2019; Mishra et al., 2022; Chen et al., 2021; Liang et al., 2023; Longpre et al., 2023; Hendrycks et al., 2021a; Wang et al., 2023a; Chiang et al., 2025; Zheng et al., 2023; Li et al., 2024a,b). In fact, one of the flagship conferences in machine learning has created a separate datasets and benchmarks track!

In this context, it is surprising that, in comparison, there has been a paucity of work understanding, measuring or controlling for the different sources of uncertainty present in the evaluations and comparisons of LLMs based on these benchmarks (Miller, 2024; Madaan et al., 2024; et al., 2024; Saad-Falcon et al., 2024; Boyeau et al., 2024; Chatzi et al., 2024; Dorner et al., 2024; Gera et al., 2024). In our work, we focus on one source of uncertainty that has been particularly overlooked, the uncertainty in the outputs of the LLMs under sampling-based decoding.

Under sampling-based decoding, given an input prompt, LLMs generate a sequence of tokens (*e.g.*, sub-words) as output using a non-deterministic autoregressive process (Bengio et al., 2000; Radford et al., 2019). At each time step, they first use a neural network to map the prompt and the (partial) sequence of tokens generated so far to a token distribution. Then, they use a sampler to draw the next token at random from the token

distribution. Finally, they append the next token to the (partial) sequence of tokens, and continue until a special end-of-sequence token is sampled. To illustrate why, in the context of LLM evaluation and ranking, one may like to control for the randomization underpinning sampling-based decoding, we will use a stylized example. Consider that we are given two LLMs m and m' and we need to evaluate the accuracy of their responses to the prompts of a benchmark dataset. However, we are not told that both LLMs are in reality identical copies of each other. It is easy to see that, due to the randomization of the autoregressive processes used by m and m', there may exist prompts where the (correctness of) their answers differ. As a consequence, one may need many responses to the prompts of the benchmark dataset to conclude confidently that m and m' achieve the same accuracy. In our work, we show both theoretically and empirically that controlling for the randomization of the autoregressive processes underpinning the LLMs under comparison can significantly reduce the number of responses required to reliably compare the performance of LLMs, and this advantage generalizes to non-identical LLMs.

Our contributions. Our key idea is to couple the autoregressive processes underpinning a set of LLMs under comparison, particularly their samplers, by means of sharing the same source of randomness. To this end, we treat the sampler of each LLM as a causal mechanism that receives as input the distribution of the next token and the same set of noise values, which determine the sampler's (stochastic) state. By doing so, at each time step of the generation, we can expect that, if different LLMs map the prompt and the (partial) sequence of tokens generated so far to the same token distribution, they will sample the same next token. Loosely speaking, in the context of LLM evaluation and ranking, coupled autoregressive generation ensures that no LLM will have better luck than others. More formally, on evaluations based on benchmark datasets, we show that the difference in average performance of each pair of LLMs under comparison is asymptotically the same under coupled and vanilla autoregressive generation, but coupled autoregressive generation provably leads to a reduction in the required sample size. On evaluations based on pairwise comparisons, we show that the win-rates of the LLMs under comparison can be asymptotically different under coupled and vanilla autoregressive generation and, surprisingly, the resulting rankings can differ. This suggests that the apparent advantage of an LLM over others in existing evaluation protocols may not be genuine but rather confounded by the randomness inherent to the generation process.

To illustrate and complement our theoretical results, we conduct experiments with several LLMs of the `Llama`, `Mistral` and `Qwen` families. We find that, across multiple benchmark datasets, namely, MMLU, GSM8K and HumanEval, coupled autoregressive generation leads to a reduction of up to 75% in the required number of samples to reach the same conclusions as vanilla autoregressive generation. Further, we find that the win-rates derived from pairwise comparisons by a strong large language model to prompts from the LMSYS Chatbot Arena platform differ under coupled and vanilla autoregressive generation. An open-source implementation of coupled autoregressive generation is available at `https://github.com/Networks-Learning/coupled-llm-evaluation`.

Related work. Our work builds upon a very recent work by Chatzi et al. (2025), which also treats the sampler of an LLM as a causal mechanism. However, their focus is different to ours; they augment a single LLM with the ability to reason counterfactually about alternatives to its own outputs if individual tokens had been different. Our work also shares technical elements with a recent work by Ravfogel et al. (2024), which develops a causal model to generate counterfactual strings resulting from interventions within (the network of) an LLM. However, their work does not study counterfactual generation for the purposes of model evaluation. In this context, it is also worth pointing out that the specific class of causal models used in the aforementioned works and our work, called the Gumbel-max structural causal model (Oberst and Sontag, 2019), has also been used to enable counterfactual reasoning in different contexts (Tsirtsis et al., 2021; Noorbakhsh and Gomez-Rodriguez, 2022; Benz and Gomez-Rodriguez, 2022). Lastly, our work broadly relates to the rapidly increasing literature on evaluation and comparison of LLMs Chang et al. (2024)—for further discussion, we refer the reader to Appendix A.

2 A CAUSAL MODEL FOR COUPLED AUTOREGRESSIVE GENERATION

Let V denote a vocabulary (set) of tokens, including an end-of-sequence token $\perp$, $V^* = V \cup V^2 \cup \cdots \cup V^K$ be the set of sequences of tokens up to a (context) length K, and $\varnothing$ be the empty token. An LLM $m \in \mathcal{M}$ takes as input a prompt sequence $s_q \in V^*$ and responds with an output sequence $s \in V^*$, generated using an autoregressive process. At each time step $i \in [K]$ of the process, the LLM first takes as input the concatenation of the prompt sequence s_q and the (partial) output sequence s_{i-1}, and generates a distribution over tokens $d_i \in \Delta(V)$. Then, under sampling-based decoding, it samples the next token $t_i \sim d_i$ from the distribution d_i and creates the output sequence $s_i = s_{i-1} \circ t_i$, where

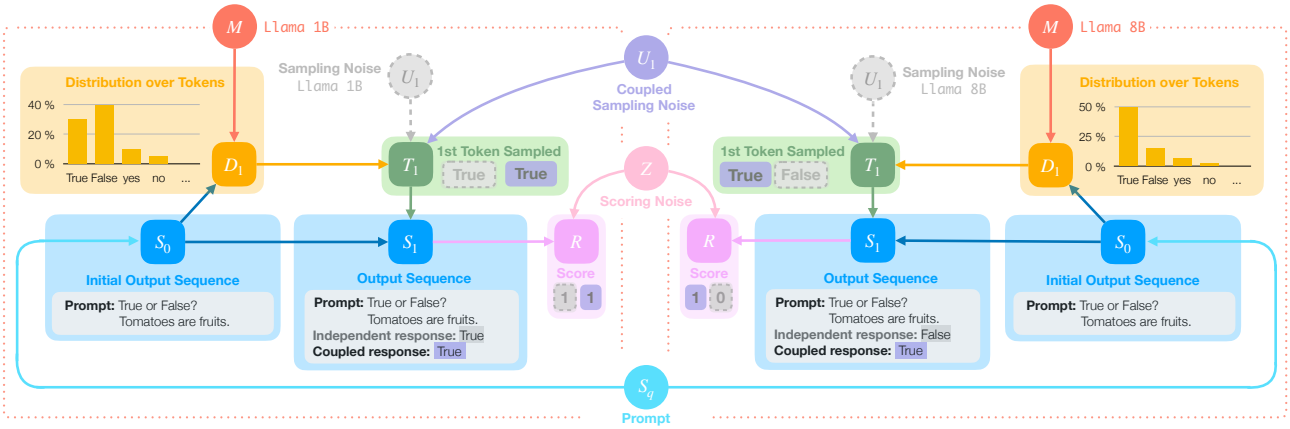


Figure 1: **Example of coupled autoregressive generation for `Llama 1B` and `Llama 8B`.** Boxes represent endogenous random variables and circles represent exogenous random variables. The value of each endogenous variable is given by a function of the values of its ancestors in the causal graph, as defined by Eq. 1. The value of the coupled noise variable U_1 (purple) is sampled independently from a given distribution P_U, and it determines the stochastic state of the samplers used by both `Llama 1B` and `Llama 8B` during the generation of token T_1.

$\circ$ denotes the concatenation of a token or sequence with another sequence. If $t_i = \perp$, it terminates and returns $s = s_i$, otherwise, it continues to the next step $i+1$ in the generation. Once the process is completed, the output sequence s is assigned a score r, which is subsequently used for model evaluation.

Following Chatzi et al. (2025), we augment the above autoregressive process using a structural causal model (SCM) (Pearl, 2009; Peters et al., 2017), denoted as $\mathcal{C}$, and use capital and lowercase letters for random variables and their realizations, respectively. The SCM $\mathcal{C}$ is defined by the following structural equations:

$$S_0 = S_q, \quad T_i = \begin{cases} f_T(D_i, U_i) & \text{if } D_i \neq P_\varnothing, \\ \varnothing & \text{otherwise} \end{cases},$$
$$D_i = \begin{cases} f_D(S_{i-1}, M) & \text{if } \texttt{last}(S_{i-1}) \neq \perp, \\ P_\varnothing & \text{otherwise} \end{cases}, \tag{1}$$
$$S_i = S_{i-1} \circ T_i, \quad S = S_K, \text{ and } \quad R = f_R(S, Z).$$

In the above equations, $M, S_q, \boldsymbol{U} = (U_i)_{i \in \{1,\ldots,K\}}$, and Z are independent exogenous random variables, with $M \sim P_M$, $S_q \sim P_Q$, $U_i \sim P_U$, and $Z \sim P_Z$. Moreover, f_D, f_T and f_R are given functions, $P_\varnothing$ denotes the point mass distribution on $\varnothing$, and $\texttt{last}(S_{i-1})$ denotes the last token of the sequence S_{i-1}. Here, the function f_D maps an input sequence S_{i-1} to a distribution D_i for the next token using the architecture and network weights of the LLM M, the function f_T and distribution P_U specify the sampling mechanism that is used to sample the next token from the distribution D_i, and the function f_R and distribution P_Z specify the scoring process by which the score R is assigned to an output sequence S during the evaluation of the LLM M.

Throughout the paper, we focus on sampling mechanisms that satisfy counterfactual stability (Oberst and Sontag, 2019; Tsirtsis et al., 2021; Chatzi et al., 2025), an intuitive form of consistency between the next token T_i, the distribution D_i, and the corresponding noise variable U_i. Loosely speaking, counterfactual stability ensures that, if the sampling mechanism samples a specific token t_i from a distribution d_i, it would still sample t_i in a counterfactual scenario where d_i is replaced by a distribution d_i' that assigns an even higher probability to t_i (refer to Appendix B for a formal definition). In this context, note that this choice does not restrict the practicality of our approach—the default categorical sampler in `PyTorch` (Paszke et al., 2019), one of the most popular libraries used by state of the art LLMs, is an implementation of the Gumbel-Max SCM (Oberst and Sontag, 2019), an SCM which satisfies counterfactual stability. Moreover, since the choice of the sampling mechanism is entirely up to the evaluator, one may be better off using a sampling mechanism that satisfies counterfactual stability in order to reduce the number of samples needed for evaluation, as we show theoretically and experimentally in our work. That said, evidence from related work (Chatzi et al., 2025) indicates that, even if the sampling mechanism does not satisfy counterfactual stability, coupled generation may still be proven useful, but to a lesser extent. Refer to Section 6 for more details.

Further, we allow the score R to be observable or unobservable, and its semantic meaning and support of its distribution to vary depending on the evaluation protocol. For example, in multiple-choice questions (Hendrycks et al., 2021b), $R \in \{0, 1\}$ may represent whether an LLM outputs a correct ($R = 1$) or

an incorrect ($R = 0$) response. In pairwise comparisons (Chiang et al., 2025), $R \in \mathbb{R}^+$ may represent the level of user's satisfaction with the response provided by an LLM. In this context, the noise variable Z models any potential sources of uncertainty in the scoring process, *e.g.*, uncertainty in users' preferences (Thurstone, 1927; Bradley and Terry, 1952; Luce, 1959).

Building upon the above causal model, we can now formally express what it means to sample (and evaluate) output sequences by different LLMs using the same source of randomness, a process we refer to as *coupled autoregressive generation*.[1] Consider a specific model m, a prompt s_q, and fixed noise values $\mathbf{u}$ and z. It is easy to see that specifying these values is sufficient to compute the exact value of the output sequence S and its score R using the autoregressive generation and scoring process given by Eq. 1. Then, we can formally express the coupled output sequences by two models m and m' and their corresponding scores $R_m(\mathbf{u}, s_q, z)$ and $R_{m'}(\mathbf{u}, s_q, z)$ as the result of *interventions* $do(M = m)$ and $do(M = m')$, respectively, where the $do(\cdot)$ operator forcibly sets the value of M while keeping the prompt s_q and the noise values $\mathbf{u}$, z fixed (Pearl, 1994). For an illustration of coupled autoregressive generation against independent autoregressive generation—the vanilla generation approach—refer to Figure 1.

From a computational perspective, it is important to note that coupled autoregressive generation does not increase the time complexity or memory footprint of the evaluation of LLMs, thus presenting zero overhead compared to the vanilla approach. Specifically, under an efficient implementation of coupled autoregressive generation, computing the (coupled) responses of a set of models under comparison simply reduces to running their generative processes using the same random seed; this effectively ensures that they all share the same prompt s_q and noise values $\mathbf{u}$ and z.[2] From a causal perspective, we can view these runs as realizations of possible worlds where everything is equal except for the (architecture and network weights of the) LLM. Or we can also view one of these runs as a realization of the factual world and the other runs as realizations of different counterfactual worlds. Consequently, this lends support to attribute any difference in the scores $R_m(\mathbf{u}, s_q, z)$ across models $m \in \mathcal{M}$ to the models' architectures and weights rather than the randomness in their autoregressive generation processes. In that context, note that coupled autoregressive generation only modifies the joint distribution of the output of the models under comparison, while keeping their marginal distributions the same as under independent autoregressive generation—it only modifies the *pairing* of the outputs, not the outputs themselves. Therefore, it does not influence any capabilities of the models, such as their creativity, that are typically associated with the randomness of their outputs (Peeperkorn et al., 2024).

[1]We (implicitly) assume that the LLMs share the same vocabulary V; refer to Section 6 for a discussion of potential approaches to relax this assumption.

[2]In benchmark datasets, z is fixed since the scoring process reduces to deterministically comparing each output against a ground truth. In pairwise comparisons, z is fixed because the scoring process for each pair of outputs under comparison is performed by an end user simultaneously.

3 EVALUATION BASED ON BENCHMARK DATASETS

In this section, we focus on the evaluation and comparison of LLMs based on benchmark datasets, *e.g.*, multiple-choice questions (Hendrycks et al., 2021b), and theoretically investigate under which conditions coupled autoregressive generation requires fewer samples than independent autoregressive generation to estimate the competitive advantage of one LLM over another.

The results we present here apply to real-valued scores in a bounded interval, however, we focus on binary scores (*e.g.*, for benchmark datasets where the output sequences can be classified as correct or incorrect) for ease of exposition. Given a benchmark dataset characterized by an input prompt distribution P_Q, for each prompt $s_q \sim P_Q$, let $\textsc{c}(s_q) \subset V^*$ denote the set of correct output sequences. Then, the score is given by $R_m(\mathbf{u}, s_q) = \mathbf{1}\{S_m(\mathbf{u}, s_q) \in \textsc{c}(s_q)\} \in \{0, 1\}$, where $S_m(\mathbf{u}, s_q)$ denotes the output sequence of a model m given a prompt s_q under a realized sequence of noise values $\mathbf{u}$ and $\mathbf{1}\{\cdot\}$ is the indicator function.

The standard approach to compare the performance of any pair of LLMs $m, m' \in \mathcal{M}$ using a benchmark dataset reduces to estimating the difference in their expected score, *i.e.*,

$$\mathbb{E}_{\boldsymbol{U} \sim P_{\boldsymbol{U}}, \boldsymbol{U}' \sim P_{\boldsymbol{U}}, S_q \sim P_Q}[\underset{\substack{\uparrow \\ \text{Independent generation}}}{R_m(\boldsymbol{U}, S_q)} - \underset{\substack{\uparrow \\ \text{Independent generation}}}{R_{m'}(\boldsymbol{U}', S_q)}], \quad (2)$$

where note that we use different noise variables $\boldsymbol{U}$ and $\boldsymbol{U}'$ for each LLM because, in the standard approach, each LLM generates outputs to each query independently (*i.e.*, using independent autoregressive generation). At first, one may think that, in this context, coupled autoregressive generation will not be helpful. Under coupled autoregressive generation, the difference in the expected score adopts the following form:

$$\mathbb{E}_{\boldsymbol{U} \sim P_{\boldsymbol{U}}, S_q \sim P_Q}[\underset{\substack{\uparrow \\ \text{Coupled generation}}}{R_m(\boldsymbol{U}, S_q)} - \underset{\substack{\uparrow \\ \text{Coupled generation}}}{R_{m'}(\boldsymbol{U}, S_q)}]. \quad (3)$$

Therefore, based on the linearity of expectation and the fact that, under independent generation, both $\boldsymbol{U}$ and

$\boldsymbol{U}'$ are sampled from the same distribution $P_{\boldsymbol{U}}$, it is easy to see that Eqs. 2 and 3 are equivalent. However, as we will show next, coupled autoregressive generation allows us to reliably estimate the difference in the two LLMs' scores from finite samples faster. More formally, we first characterize the relation between the variances of the difference of scores between LLMs as follows (refer to Appendix D for all proofs):

Proposition 1 *For any $m, m' \in \mathcal{M}$, it holds that*

$$\begin{aligned}\mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)] \\ = \mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)] \\ + 2 \cdot \mathrm{Cov}[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)]\end{aligned}$$

This result immediately implies that, if the scores achieved by the LLMs under comparison are positively correlated, *i.e.*, the LLMs tend to generate a (in-)correct output sequence on the same prompts under the same noise values, then the variance of the difference in scores is lower under coupled generation than under independent generation, and thus we can expect a reduction in the sample size required to obtain equivalent estimation errors. Next, we theoretically analyze two canonical settings in which this condition holds and, in Section 5, we provide empirical evidence that it also holds in well-known benchmark datasets.

In the first canonical setting, the correct response to each prompt is one of two given single-token sequences and the LLMs m and m' under comparison always output a response that is either of these two sequences. While this setting may seem restrictive, it is found in real-world scenarios. For example, think of evaluation protocols in which the LLMs are explicitly instructed to always output true/false (or one of two options) via their system prompt.[3] The following proposition shows that the variance of the difference in scores is lower under coupled autoregressive generation:

Proposition 2 *Consider a benchmark dataset such that $\mathrm{C}(s_q) \subsetneq \{t_1, t_2\}$ for all $s_q \sim P_Q$, where t_1 and t_2 are two single-token sequences. Let m and m' be two LLMs that assign positive probability to the sequences t_1 and t_2 and zero probability to any other sequence. If the sampling mechanism defined by f_T and P_U satisfies counterfactual stability, then, it holds that*

$$\begin{aligned}\mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)] \\ > \mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)]. \quad (4)\end{aligned}$$

[3]Here, our goal is to illustrate that there exist natural conditions under which coupled autoregressive generation is provably beneficial in comparison to independent autoregressive generation. However, in practice, in this canonical setting, one could directly use the LLMs' probabilities for the two tokens in each prompt to estimate the average difference of scores exactly.

In the second canonical setting, the correct response to each prompt is a single-token sequence, the LLMs m and m' under comparison always output a single-token response, and the sampling mechanism used by the LLMs is given by the Gumbel-Max SCM[4]. Similarly as in the first canonical setting, this second setting is also found in real-world scenarios, particularly taking into account that the default categorical sampler in the library `PyTorch` implements the Gumbel-Max SCM. The following proposition shows that, as long as the model m' is *similar enough* to m, the variance of the difference in scores is lower under coupled generation:

Proposition 3 *Consider a benchmark dataset such that $|\mathrm{C}(s_q)| = 1$ for all $s_q \sim P_Q$. Let m be an LLM that assigns positive probability to every single-token sequence and zero probability to any other sequence. If the sampling mechanism defined by f_T and P_U is given by the Gumbel-Max SCM, then, there exists a constant $\varepsilon(m) > 0$ such that, for every LLM m' that assigns positive probability to every single-token sequence and zero probability to any other sequence and satisfies $d(m, m') = \sup_{s_q} \|f_D(s_q, m) - f_D(s_q, m')\|_\infty < \varepsilon(m)$, it holds that*

$$\begin{aligned}\mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)] \\ > \mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)].\end{aligned}$$

Based on the above proposition, we hypothesize that coupled autoregressive generation will reduce the number of samples required to reliably compare the performance of LLMs whenever these are sufficiently *similar*, *e.g.*, whenever we compare fine-tuned or quantized versions of the same pre-trained LLM.

4 EVALUATION BASED ON PAIRWISE COMPARISONS

In this section, we focus on the evaluation and comparison of LLMs according to their level of alignment with human preferences, as elicited by pairwise comparisons between outputs of different LLMs to the same prompts. Such an evaluation protocol has become particularly popular to evaluate and compare LLMs in open-ended, complex tasks in which, in contrast to benchmark datasets, there are no structured ground-truth outputs. In what follows, we provably show that, perhaps surprisingly, different LLMs may compare differently under coupled autoregressive generation and under independent autoregressive generation.

[4] The Gumbel-Max SCM is defined as $f_T(D_i, U_i) = \mathrm{argmax}_{t \in V} \{\log(D_{i,t}) + U_{i,t}\}$, where $U_{i,t} \sim \mathrm{Gumbel}(0, 1)$ are i.i.d. noise variables associated with each token (Chatzi et al., 2025).

One of the standard approaches to evaluate and compare different LLMs according to their level of alignment with human pairwise preferences reduces to estimating the win-rate achieved by each LLM m against any other LLM $m' \neq m$, *i.e.*,

$$\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}},\boldsymbol{U}'\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_m(\underset{\uparrow}{\boldsymbol{U}}, S_q) > R_{m'}(\underset{\uparrow}{\boldsymbol{U}'}, S_q)\}] \quad (5)$$

Independent generation

where $\mathbf{1}\{R_m(\mathbf{u}, s_q) > R_{m'}(\mathbf{u}, s_q)\} = 1\,(0)$ means that, for prompt s_q and realized sequence of noise values $\mathbf{u}$, the output of m is (not) preferred over the output of m'.[5]

Here, similarly as in Eq. 2 in the evaluation based on benchmark datasets, we use different noise variables $\boldsymbol{U}$ and $\boldsymbol{U}'$ because, in this standard approach, each LLM generates outputs to each prompt independently (*i.e.*, using independent autoregressive generation). Conversely, under coupled autoregressive generation, the win-rate adopts the following form:

$$\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_m(\underset{\uparrow}{\boldsymbol{U}}, S_q) > R_{m'}(\underset{\uparrow}{\boldsymbol{U}}, S_q)\}] \quad (6)$$

Coupled generation

However, in contrast with the comparison of the expected difference in scores under independent and coupled autoregressive generation in the evaluation based on benchmark datasets, we cannot directly claim that Eqs. 5 and 6 are equivalent because the win-rate is non-linear with respect to $R_m(\mathbf{u}, s_q)$ and $R_{m'}(\mathbf{u}', s_q)$. In what follows, we further analyze the difference between win-rates in a canonical setting similar to one of the canonical settings used in Section 3.

Consider that, for each prompt, the response can only be one of two given single-token sequences, one of these sequences is preferred over the other by the user, and the LLMs under comparison always output one of them as a response. Then, we can compute the win-rates achieved by each LLM m against any other LLM $m' \neq m$ under independent and coupled autoregressive generation:

Proposition 4 *Given a fixed prompt $s_q \sim P_Q$, assume that $f_R(s_+) > f_R(s_-)$ for $s_+ = s_q \circ t_+$ and $s_- = s_q \circ t_-$, where t_+ and t_- are single-token sequences. Further, assume that the LLMs m and m' respond t_+ with probability p_m and $p_{m'}$, respectively, and t_- with probability $1 - p_m$ and $1 - p_{m'}$, and the sampling mechanism defined by f_T and P_U satisfies counterfactual stability. Without loss of generality, assume $p_{m'} > p_m$. Then, under coupled autoregressive generation, we have that*

$$\mathbb{E}_{\boldsymbol{U}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) > R_{m'}(\boldsymbol{U}, s_q)\}] = 0,$$
$$\mathbb{E}_{\boldsymbol{U}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) < R_{m'}(\boldsymbol{U}, s_q)\}] = p_{m'} - p_m.$$

Conversely, under independent autoregressive generation, we have that

$$\mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) > R_{m'}(\boldsymbol{U}', s_q)\}] = p_m(1 - p_{m'}),$$
$$\mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) < R_{m'}(\boldsymbol{U}', s_q)\}] = p_{m'}(1 - p_m)$$

From the above proposition, we can readily conclude that, in general, the win-rates do differ under independent and coupled autoregressive generation. Nevertheless, we may be tempted to conclude that, for ranking LLMs, this difference appears inconsequential because, for each fixed prompt s_q, we have that

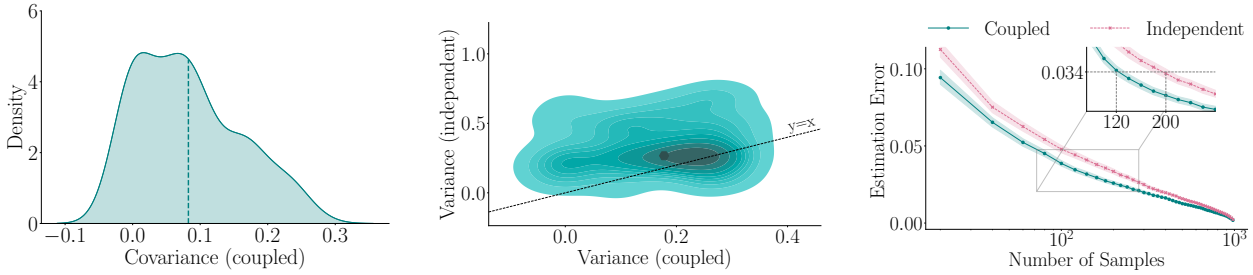
$$\begin{aligned}&\mathbb{E}_{\boldsymbol{U}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) < R_{m'}(\boldsymbol{U}, s_q)\}] \\ &\qquad - \mathbb{E}_{\boldsymbol{U}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) > R_{m'}(\boldsymbol{U}, s_q)\}] \\ &= \mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) < R_{m'}(\boldsymbol{U}', s_q)\}] \\ &\qquad - \mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) > R_{m'}(\boldsymbol{U}', s_q)\}].\end{aligned}$$

However, whenever one needs to rank more than two LLMs, the difference in win-rates can be actually consequential—the rankings derived from the win-rates can be different under independent and coupled autoregressive generation, as illustrated by the following simple example.

Consider we are given three LLMs m_1, m_2, and m_3, and we need to rank them according to the average win-rate they achieve against each other on two input prompts q and q', each with a preferred single-token response out of two single-token responses. Assume that the probability that the LLMs output the preferred single-token response for q and q' are given by $(m_1\colon 0.4, m_2\colon 0.48, m_3\colon 0.5)$ and $(m_1\colon 1, m_2\colon 0.9, m_3\colon 0.89)$, respectively. Under independent autoregressive generation, the average win-rates of m_1, m_2 and m_3 are 0.1545, 0.15675 and 0.16225, respectively. Therefore, m_3 is ranked at the top, followed by m_2, and m_1 is ranked last. In contrast, under coupled autoregressive generation, the average win-rates of m_1, m_2 and m_3 are 0.0525, 0.0225, and 0.03, respectively, and thus m_1 is ranked at the top, followed by m_3, and m_2 is ranked last.[6] Crucially, this case illustrates how rankings obtained using coupled and independent autoregressive generation can differ, leading to opposite conclusions regarding the LLMs' performance. In Appendix C.1, we analyze another canonical setting in which the win-rates also differ under independent and coupled autoregressive generation as long as the LLMs are *sufficiently similar*.

[5] For simplicity, we assume that human preferences are deterministic and thus $R_m(\mathbf{u}, s_q, z) = R_m(\mathbf{u}, s_q)$. We lift this assumption in our experiments in Section 5.

[6] Refer to Appendix D.6 for the detailed calculation of the average win-rates.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 2: **Comparison between `1B` and `3B` on questions from the knowledge area "college computer science" of the MMLU dataset.** Panel (a) shows the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed line corresponds to the average value. Panel (b) shows the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted point corresponds to the median value. Panel (c) shows the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.

5 EXPERIMENTS

In this section, we evaluate three LLMs from the `Llama` family of different sizes, namely, `Llama-3.2-{1B, 3B}-Instruct` and `Llama-3.1-8B-Instruct`, as well as three quantized variants of the latter, namely, `Llama-3.1-8B-Instruct-{AWQ-INT4, bnb-4bit, bnb-8bit}`, under coupled and independent autoregressive generation using an implementation of the Gumbel-Max SCM as a sampler (Chatzi et al., 2025). In the remainder of the paper, we refer to them as `1B`, `3B`, `8B`, `AWQ-INT4`, `bnb-4bit`, and `bnb-8bit`. For implementation details related to the hardware, datasets and models, refer to Appendix E. For additional qualitatively similar results using models from the `Mistral` and `Qwen` families, refer to Appendices G and H. [7]

5.1 Evaluation on Benchmark Datasets

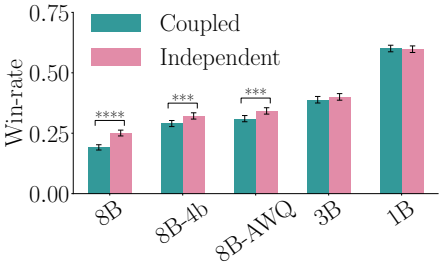
Here, we compare the aforementioned LLMs using the MMLU benchmark dataset (Hendrycks et al., 2021b), which comprises 14,042 multiple choice questions covering 52 knowledge areas. Our goal is to empirically investigate to what extent the theoretical results in Section 3 generalize to evaluations based on well known benchmark datasets.[8] For additional qualitatively similar results beyond multiple-choice benchmarks using the GSM8K (Cobbe et al., 2021) and HumanEval (Chen et al., 2021) benchmark datasets, refer to Appendix F.2.

Experimental setup. In our experiments, for each multiple choice question in the MMLU benchmark dataset, we provide the question itself together with the available options (4 for each question, indexed from A to D) as an input prompt to the LLMs. Further, we instruct the LLMs to generate an output sequence comprising only the index of the selected option through a system prompt—refer to Appendix E for the exact prompt. To evaluate the outputs provided by each LLM, we use a binary score $R \in \{0, 1\}$, which indicates whether the LLM output is the (single) correct ($R = 1$) or incorrect ($R = 0$) answer of the given options. To obtain reliable conclusions, we experiment with each multiple choice question 10 times, each time using a (different) random seed to generate the noise variables used by the sampler. Due to space constraints, in what follows, we compare `1B` and `3B` on the knowledge area "college computer science". In Appendix F.1, we provide qualitatively similar results on other knowledge areas, different sampling parameters, and other LLMs.

Results. Figures 2a and 2b show that the scores of the LLMs are positively correlated under coupled generation and thus the variance of the difference in scores is lower under coupled generation than under independent, in agreement with Proposition 1. Further, we compute the error in the estimation of the expected difference in scores resulting from using the

[7] Although we experiment only with open-weights LLMs, coupled generation is technically applicable to proprietary LLMs as well, since it only requires access to next-token probabilities. For example, it can be used during internal evaluations or on crowd-sourced benchmarking platforms, where providers may be incentivized to share them with the platform to enable more sample-efficient comparisons.

[8] In practice, for multiple-choice benchmarks, the standard evaluation protocol often uses greedy decoding (*i.e.*, deterministic generation). However, multiple-choice benchmarks offer a simple setting to show quantitatively the effect of coupled autoregressive generation in reducing the amount of samples required for the evaluation of models using sampling-based decoding.



(a) Empirical win-rates of `bnb-8bit`

LLM	Coupled		Independent	
	Rank	Avg. win-rate	Rank	Avg. win-rate
`8B`	1	0.3670 ±0.0020	1	0.3863 ±0.0020
`bnb-8bit`	2	0.3562 ±0.0020	1	0.3825 ±0.0020
`bnb-4bit`	3	0.3339 ±0.0020	3	0.3463 ±0.0020
`AWQ-INT4`	4	0.3164 ±0.0019	4	0.3310 ±0.0019
`3B`	5	0.2787 ±0.0019	5	0.2828 ±0.0019
`1B`	6	0.1650 ±0.0015	6	0.1664 ±0.0015

(b) Average win-rate and ranking of each LLM

Figure 3: **Evaluation of six LLMs using pairwise comparisons on questions from the LMSYS-Chat-1M dataset.** We estimate the empirical win-rate of each LLM against each other using pairwise comparisons between the outputs to 500 questions with 10 (different) random seeds under both coupled and independent generation. Panel (a) shows the empirical win-rate of `bnb-8bit` against all other LLMs, where error bars correspond to 95% confidence intervals. Here, for each pair of empirical win-rates under coupled and independent generation, we conduct a two-tailed z-test, to test the null hypothesis that the empirical win-rates are the same; (∗∗∗∗, ∗∗∗) indicate p-values (< 0.0001, < 0.001). We present qualitatively similar results for other LLMs in Appendix F.3. Panel (b) shows the average win-rate of each LLM across all other LLMs (± 95% confidence intervals). To derive the rankings, for each LLM, we choose the lowest ranking provided by the method of Chatzi et al. (2024).

two approaches as a function of the available sample size. To this end, we first estimate the expected score difference using 1,000 samples and consider this as (a proxy of) the ground truth. Then, we compute the absolute estimation error achieved by independent and coupled generation while sub-sampling the original samples across various sample sizes. Figure 2c summarizes the results, which show that, as expected from our theoretical analysis, a lower variance of the difference in scores under coupled generation leads to a reduction in the number of samples required to achieve equivalent error in the estimation of the expected difference between the scores of the LLMs. Perhaps surprisingly, we find that this reduction can, in practice, be quite large. For example, to achieve an estimation error of ≈0.034, coupled generation needs 40% fewer samples than independent generation. In Appendix F.1, we find that this reduction is up to a striking 75% for sufficiently similar LLMs in agreement with Proposition 3. Finally, it is worth noting that, when the models under comparison are very different, there is essentially zero correlation between their evaluation scores and, hence, independent and coupled generation become effectively equivalent; see, for example, Figure 14 in Appendix F.2.

5.2 Evaluation on Pairwise Comparisons

Here, we compare the same LLMs as in the previous section using pairwise comparisons between their outputs by a strong LLM, when prompted with open-ended questions from the LMSYS Chatbot Arena platform (LMSYS, 2023). Our goal is to investigate to what extent the theoretical results derived in Section 4 generalize to a commonly used evaluation protocol (*i.e.*, LLM-as-a-judge), and not to what extent the rankings derived using a strong LLM are consistent with human preferences (Wang et al., 2024).

Experimental setup. We experiment with 500 questions from the LMSYS-Chat-1M dataset (Zheng et al., 2024). We provide the question itself as an input prompt to the LLMs, and instruct them to generate a concise response as an output through a system prompt. Further, similarly as elsewhere (Chatzi et al., 2024; Li et al., 2024c; Boyeau et al., 2024; Gera et al., 2024; Li et al., 2024a; Zheng et al., 2023), we use a strong LLM, namely, `GPT-4o-2024-11-20`, as a judge. More specifically, for each question and pair of outputs provided by two different LLMs, we prompt the judge to respond which of the two outputs it prefers, but allowing the judge to declare a tie—for the exact prompts we use, refer to Appendix E. Given these pairwise comparisons, to evaluate the outputs provided by each LLM, we use the win-rate achieved by each LLM against each other. To obtain reliable conclusions, similarly as in the previous section, we repeat each experiment 10 times, each time using a (different) random seed to generate the Gumbel noise variables used by the Gumbel-Max SCM.

Results. We find that the empirical win-rate of each LLM against any other LLM is generally lower under coupled generation than under independent generation, as shown in Figure 3a for `bnb-8bit` and Figure 16 in Appendix F.3 for other LLMs. Moreover, whenever the LLMs under comparison are *sufficiently* similar, the difference between win-rates is statistically significant, suggesting that our theoretical results may generalize beyond the canonical setting discussed in Section 4. We hypothesize that this is partially due to an increase

in the number of ties under coupled autoregressive generation. For example, for `bnb-8bit`, we observe a 24%, 11%, 15% increase in the number of ties in the pairwise comparisons against `8B`, `bnb-4bit`, and `AWQ-INT4`. Remarkably, the difference in empirical win-rates leads to differences in the rankings derived from the average win-rates, as shown in Figure 3b. Under independent generation, the average win-rates achieved by `8B` and `bnb-8bit` are statistically indistinguishable and they are both ranked at the top. However, under coupled generation, only `8B` is ranked at the top.

6 DISCUSSION AND LIMITATIONS

In this section, we discuss several aspects of our work which we believe are important to consider and may serve as a basis for future research.

Model assumptions. Our theoretical analysis of coupled autoregressive generation focuses on sampling mechanisms that satisfy counterfactual stability (Oberst and Sontag, 2019). Although counterfactual stability has been shown to be a desirable property for causal mechanisms in SCMs and, more specifically, for causal mechanisms used for sampling in LLMs (Chatzi et al., 2025), counterfactual stability may not always be appropriate and should be justified by domain specific knowledge (Haugh and Singal, 2023). In this context, it is worth mentioning that Chatzi et al. (2025) have empirically shown that, both under the counterfactually stable Gumbel-Max sampling mechanism and a non-counterfactually stable mechanism based on inverse transform sampling, controlling for the randomness in two autoregressive processes with sufficiently similar next-token distributions yields more similar outputs. Based on this evidence, we also expect that, even if a sampling scheme does not satisfy counterfactual stability, coupled generation may still reduce variance or alter model rankings as long as the LLMs under comparison are not very different, but to a lesser extent. It would be interesting to understand the sensitivity of coupled autoregressive generation to the specific choice of the Gumbel-Max SCM as well as extending our analysis to sampling mechanisms satisfying properties other than counterfactual stability (Vlontzos et al., 2023).

Practical considerations. Our experimental results and theoretical analysis suggest that coupled autoregressive generation is most advantageous over independent autoregressive generation whenever the LLMs under comparison are sufficiently close in terms of their next-token distributions. It would be important to identify which parts of the LLM development pipeline (*e.g.*, the LLMs' architectures, training data, or fine-tuning process) lead, in practice, to sufficiently small changes in the next-token distributions for coupled autoregressive generation to be most beneficial.

Throughout our paper, we have focused on comparisons of models sharing the same token vocabulary, which include important practical use cases. For example, during the rapid and continual development of a large language model, one may like to compare different model versions whose performance differences are not easily predictable, such as those resulting from small architectural changes, data preprocessing, and supervised fine-tuning. Nevertheless, there are also many practical use cases where one may want to compare models using different token vocabularies, for which our methodology could, in principle, be extended leveraging very recent methods in the field of tokenization (Vieira et al., 2025; Zheng et al., 2025). More specifically, one could transform two models with different token vocabularies into equivalent character-level models over a shared vocabulary of tokens corresponding exclusively to characters, and then apply coupled generation on the derived character-level models. However, this would bring additional challenges. For example, to compute the next-character distributions, the above methods use approximations to marginalize over all possible suffixes of a string and thus the relative performance of the original models might not be preserved.

Evaluation. We have experimented with (i) a single dataset of prompts for pairwise comparisons (*i.e.*, LMSYS Chatbot Arena), where we have focused on win-rate as an evaluation metric, and (ii) we have used a strong LLM as a judge (*i.e.*, `GPT-4o-2024-11-20`), which may introduce biases and lead to rankings that are inconsistent with (the distribution of) human preferences (Wang et al., 2024). To better understand the benefits of coupled autoregressive generation, it would be important to experiment with additional datasets, pairwise comparisons made by humans, and additional evaluation metrics based on, *e.g.*, the Elo rating system (Askell et al., 2021; Dettmers et al., 2024; Bai et al., 2022; Wu et al., 2023; Lin and Chen, 2023) and the B-T model (Chiang et al., 2025; Boyeau et al., 2024).

7 CONCLUSIONS

We have introduced a causal model of coupled autoregressive generation that enables the evaluation and comparison of different LLMs under the same source of randomness. We have theoretically and empirically shown that, in evaluations based on benchmark datasets, coupled autoregressive generation can reduce the number of samples required to reliably compare the performance of LLMs and, in evaluations based on pairwise comparisons, it can lead to different and, perhaps more intuitive, rankings of LLMs in comparison with independent autoregressive generation.

Acknowledgements

Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreements No. 945719 and 101169607). Tsirtsis acknowledges support from the Alexander von Humboldt Foundation in the framework of the Alexander von Humboldt Professorship (Humboldt Professor of Technology and Regulation awarded to Sandra Wachter) endowed by the Federal Ministry of Education and Research via the Hasso Plattner Institute. Straitouri acknowledges support from a Google PhD Fellowship.

References

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. Reading Between the Lines: Modeling User Behavior and Costs in AI-Assisted Programming. In *Proceedings of the Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024.

Claudia E Haupt and Mason Marks. AI-Generated Medical Advice—GPT and Beyond. *Journal of American Medical Association*, 329(16):1349–1350, 2023.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical Discoveries from Program Search with Large Language Models. *Nature*, 625(7995): 468–475, 2023.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts. In *Proceedings of the Association for Computational Linguistics: System Demonstrations*, pages 93–104. ACL, May 2022.

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 *Proceedings of the International Conference on Learning Representations*. ICLR, 2022.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158. ACL, 2019.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland, May 2022. ACL.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021.

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.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the International Conference on Machine Learning*, pages 22631–22648. PMLR, Jul 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *Proceedings of the International Conference on Learning Representations*. ICLR, 2021a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508. ACL, 2023a.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: an open platform for evaluat-

ing LLMs by human preference. In *Proceedings of the International Conference on Machine Learning*, 2025.

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 *Advances in Neural Information Processing Systems, data track*, pages 46595–46623. Curran Associates, Inc., 2023.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative Judge for Evaluating Alignment. In *Proceedings of the International Conference on Learning Representations*. ICLR, 2024a.

Ruosen Li, Teerth Patel, and Xinya Du. PRD: Peer Rank and Discussion Improve Large Language Model based Evaluations. *Transactions on Machine Learning Research*, 2024b.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Lovish Madaan, Aaditya K Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. *arXiv preprint arXiv:2406.10229*, 2024.

Abhimanyu Dubey et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354. ACL, 2024.

Pierre Boyeau, Anastasios N Angelopoulos, Nir Yosef, Jitendra Malik, and Michael I Jordan. AutoEval Done Right: Using Synthetic Data for Model Evaluation. *arXiv preprint arXiv:2403.07008*, 2024.

Ivi Chatzi, Eleni Straitouri, Suhas Thejaswi, and Manuel Gomez Rodriguez. Prediction-powered ranking of large language models. In *Advances in Neural Information Processing Systems*, 2024.

Florian E. Dorner, Vivian Y. Nastl, and Moritz Hardt. Limits to scalable evaluation at the frontier: LLM as judge won't beat twice the data. *arXiv preprint arXiv:2410.13341*, 2024.

Ariel Gera, Odellia Boni, Yotam Perlitz, Roy Bar-Haim, Lilach Eden, and Asaf Yehudai. Justrank: Benchmarking LLM judges for system ranking. *arXiv preprint arXiv:2412.09569*, 2024.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in Neural Information Processing Systems*, 13, 2000.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ivi Chatzi, Nina L Corvelo Benz, Eleni Straitouri, Stratis Tsirtsis, and Manuel Gomez-Rodriguez. Counterfactual token generation in large language models. In *Proceedings of the 4th Conference on Causal Learning and Reasoning*, volume 275 of *Proceedings of Machine Learning Research*, pages 1291–1315. PMLR, 2025.

Shauli Ravfogel, Anej Svete, Vésteinn Snæbjarnarson, and Ryan Cotterell. Counterfactual generation from language models. *arXiv preprint arXiv:2411.07180*, 2024.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with Gumbel-Max structural causal models. In *Proceedings of the International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

Stratis Tsirtsis, Abir De, and Manuel Rodriguez. Counterfactual explanations in sequential decision making under uncertainty. In *Advances in Neural Information Processing Systems*, 2021.

Kimia Noorbakhsh and Manuel Gomez-Rodriguez. Counterfactual temporal point processes. In *Advances in Neural Information Processing Systems*, 2022.

Nina L Corvelo Benz and Manuel Gomez Gomez-Rodriguez. Counterfactual inference of second opinions. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, 2022.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 2024.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations*, 2021b.

Louis L Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927. doi: 10.1037/h0070288.

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

R Duncan Luce. *Individual choice behavior*, volume 4. Wiley New York, 1959.

Judea Pearl. A probabilistic calculus of actions. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, 1994.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. Is temperature the creativity parameter of large language models? *arXiv preprint arXiv:2405.00492*, 2024.

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.

LMSYS. Chatbot Arena: Benchmarking LLMs in the Wild with Elo Ratings. `https://lmsys.org/`, 2023. Online; accessed 21 May 2024.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the Association for Computational Linguistics*, pages 9440–9450. ACL, 2024. doi: 10.18653/v1/2024.acl-long.511.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world LLM conversation dataset, 2024.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. LLMs-as-Judges: A comprehensive survey on LLM-based evaluation methods. *arXiv preprint arXiv:2412.05579*, 2024c.

Martin B Haugh and Raghav Singal. Counterfactual analysis in dynamic latent state models. In *Proceedings of the International Conference on Machine Learning*, 2023.

Athanasios Vlontzos, Bernhard Kainz, and Ciarán M Gilligan-Lee. Estimating categorical counterfactuals via deep twin networks. *Nature Machine Intelligence*, 5(2): 159–168, 2023.

Tim Vieira, Ben LeBrun, Mario Giulianelli, Juan Luis Gastaldi, Brian DuSell, John Terilla, Timothy J O'Donnell, and Ryan Cotterell. From language models over tokens to language models over characters. In *Proceedings of the International Conference on Machine Learning*, 2025.

Brian Siyuan Zheng, Alisa Liu, Orevaoghene Ahia, Jonathan Hayase, Yejin Choi, and Noah A Smith. Broken tokens? your language model can secretly handle non-canonical tokenizations. *arXiv preprint arXiv:2506.19004*, 2025.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A General Language Assistant as a Laboratory for Alignment. *arXiv preprint arXiv:2112.00861*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. In *Advances in Neural Information Processing Systems*, pages 10088–10115. Curran Associates, Inc., 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Yuxiang Wu, Zhengyao Jiang, Akbir Khan, Yao Fu, Laura Ruis, Edward Grefenstette, and Tim Rocktäschel. ChatArena: Multi-Agent Language Game Environments for Large Language Models. `https://github.com/chatarena/chatarena`, 2023.

Yen-Ting Lin and Yun-Nung Chen. LLM-Eval: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models. In *Proceedings of the Workshop on NLP for Conversational AI*, pages 47–58. ACL, July 2023.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning Large Language Models with Human: A Survey. *arXiv preprint arXiv:2307.12966*, 2023b.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, 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 *Advances in Neural Information Processing Systems*, pages 27730–27744. Curran Associates, Inc., 2022.

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. Online; accessed 08 Aug 2025.

Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. Elo uncovered: Robustness and best practices in language model evaluation. In *Advances in Neural Information Processing Systems*, 2024.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large Language Models Encode Clinical Knowledge. *Nature*, 620(7972):172–180, July 2023.

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. `https://vicuna.lmsys.org`, 2023. Online; accessed 21 May 2024.

Arpad E. Elo. *The USCF Rating System: Its Development, Theory, and Applications.* United States Chess Federation, 1966.

Quentin Bertrand, Wojciech Marian Czarnecki, and Gauthier Gidel. On the Limitations of the Elo, Real-World Games are Transitive, Not Additive. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 2905–2921. PMLR, 2023.

Iris A. M. Huijben, Wouter Kool, Max B. Paulus, and Ruud J. G. van Sloun. A review of the Gumbel-Max trick and its extensions for discrete stochasticity in machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1353–1371, 2023. doi: 10.1109/TPAMI.2022.3157042.

Bits and Bytes Foundation. Bits and bytes quantisation library. `https://huggingface.co/docs/bitsandbytes/main/en/index`, 2024. Online; accessed 08 Aug 2025.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Evaluation of Large Language Models via Coupled Token Generation: Supplementary Materials

A FURTHER RELATED WORK

Our work also builds upon the rapidly increasing literature on evaluation and comparison of LLMs (Chang et al., 2024). Within this literature, LLMs are evaluated and compared using: (i) benchmark datasets with manually hand-crafted inputs and ground-truth outputs (Bach et al., 2022; Wei et al., 2022; Talmor et al., 2019; Mishra et al., 2022; Chen et al., 2021; Liang et al., 2023; Longpre et al., 2023) and (ii) the level of alignment with human preferences, as elicited by means of pairwise comparisons (Wang et al., 2023b; Ouyang et al., 2022; Taori et al., 2023; Zheng et al., 2023; Li et al., 2024a,b; Boubdir et al., 2024; Singhal et al., 2023; Chiang et al., 2025). However, it has become clear that oftentimes rankings derived from benchmark datasets do not match those derived from human preferences (Zheng et al., 2023; Li et al., 2024a,b; Chiang et al., 2023, 2025). Within the literature on ranking LLMs from pairwise comparisons, most studies use the Elo rating system (Askell et al., 2021; Dettmers et al., 2024; Bai et al., 2022; Wu et al., 2023; Lin and Chen, 2023), originally introduced for chess tournaments (Elo, 1966). However, Elo-based rankings are sensitive to the order of pairwise comparisons, as newer comparisons have more weight than older ones, which leads to unstable rankings (Boubdir et al., 2024). To address this limitation, several studies have instead used the Bradley-Terry (B-T) model (Chiang et al., 2025; Boyeau et al., 2024), which weighs pairwise comparisons equally regardless of their order. Nevertheless, both the Elo rating system and the B-T model have faced criticism, as pairwise comparisons often fail to satisfy the fundamental axiom of transitivity, upon which both approaches rely (Boubdir et al., 2024; Bertrand et al., 2023). Recently, several studies have used the win-rate (Zheng et al., 2023; Chiang et al., 2025; Boyeau et al., 2024), which weighs comparisons equally regardless of their order and does not require the transitivity assumption. In our work, we focus on win-rates. However, it may be possible to extend our theoretical and empirical results to rankings based on Elo ratings and the B-T model.

B FORMAL DEFINITION OF COUNTERFACTUAL STABILITY

Counterfactual stability is a desirable property of SCMs (Oberst and Sontag, 2019) that has previously been used in the context of autoregressive generation of LLMs (Chatzi et al., 2025). In the following, we provide its formal definition along with a simple example to explain the intuition behind it. Throughout this section, $P^{\mathcal{C}\,;\,do(\cdot)}$ denotes the probability of the interventional distribution entailed by an SCM $\mathcal{C}$ under an intervention $do(\cdot)$. Moreover, $P^{\mathcal{C}\,|\,\star\,;\,do(\cdot)}$ denotes the probability of the counterfactual distribution entailed by an SCM $\mathcal{C}$ under an intervention $do(\cdot)$ given that an observed event $\star$ has already occurred.

Definition 1 *A sampling mechanism defined by f_T and P_U satisfies counterfactual stability if for all LLMs $m, m' \in \mathcal{M}$, $i \in \{1, 2, \ldots, K\}$ and tokens $t_1, t_2 \in V$ with $t_1 \neq t_2$, the condition*

$$\frac{P^{\mathcal{C}\,;\,do(M=m')}[T_i = t_1 \mid D_i]}{P^{\mathcal{C}\,;\,do(M=m)}[T_i = t_1 \mid D_i]} \geq \frac{P^{\mathcal{C}\,;\,do(M=m')}[T_i = t_2 \mid D_i]}{P^{\mathcal{C}\,;\,do(M=m)}[T_i = t_2 \mid D_i]} \tag{7}$$

implies that $P^{\mathcal{C}\,|\,D_i, M=m, T_i=t_1\,;\,do(M=m')}[T_i = t_2] = 0$.

The property of counterfactual stability has an intuitive interpretation that can be best understood via a simple example. Assume that the vocabulary contains 2 tokens "A" and "B" and, using LLM m, the next-token distribution at a time step i assigns values 0.6, 0.4 to the two tokens, respectively. Moreover, the realized noise value $\mathbf{u}_i$ is such that the token "A" is sampled. Now, consider that, while keeping the noise value $\mathbf{u}_i$ fixed, we change the LLM to m', resulting in a next-token distribution that assigns values 0.7, 0.3 to the two tokens, respectively. Counterfactual stability ensures that, since the noise value $\mathbf{u}_i$ led to "A" being sampled under m at 0.6 to 0.4 odds, the same value cannot lead to "B" being sampled under m' where its relative odds are lower (*i.e.*, 0.3 to 0.7).

C ADDITIONAL THEORETICAL RESULTS

C.1 Evaluation based on Pairwise Comparison

In this Section, we consider another canonical setting in which, for each prompt, the response can be one of any single-token sequences, and each of the sequences may provide a different level of user's satisfaction (*i.e.*, achieve a different score). Further, the LLMs under comparison always output one of them as a response and the sampling mechanism used by the LLMs is given by the Gumbel-Max SCM as defined in Footnote 4 in Section 3. The following proposition shows that the number of ties between an LLM m and any other *sufficiently similar* LLM $m' \neq m$ are higher under coupled autoregressive generation than under independent autoregressive generation:

Proposition 5 *Given a fixed prompt $s_q \sim P_Q$, assume, without loss of generality, that $f_R(s_q \circ t_1) \geq f_R(s_q \circ t_2) \geq \ldots \geq f_R(s_q \circ t_{|V|})$. Let m be an LLM that assigns positive probability to every single-token sequence and zero probability to any other sequence. If the sampling mechanism defined by f_T and P_U is given by the Gumbel-Max SCM, then, there exists a constant $\varepsilon(m) > 0$ such that, for every LLM m' that assigns positive probability to every single-token sequence and zero probability to any other sequence and satisfies $d(m, m') = \sup_{s_q} \|f_D(s_q, m) - f_D(s_q, m')\|_\infty < \varepsilon(m)$, it holds that*

$$\mathbb{E}_{\boldsymbol{U} \sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}, s_q)\}] > \mathbb{E}_{\boldsymbol{U}, \boldsymbol{U}' \sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}', s_q)\}].$$

The above proposition implies that the win-rates under independent and coupled autoregressive generation are different and, similarly as in the first canonical setting, rankings derived from the win-rates may differ under independent and coupled autoregressive generation. We investigate this further in our experiments in Section 5.

D PROOFS

D.1 Proof of Proposition 1

We can rewrite the variance of the difference in scores under independent generation in terms of the variance of the difference in scores under coupled generation as follows:

$$\begin{aligned}
\mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)]] &= \mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q) + R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)]] \\
&= \mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)] + \mathrm{Var}[R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)] \\
&\quad + 2 \cdot \mathrm{Cov}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)].
\end{aligned}$$

For the variance of the difference in scores for the same LLM under independent noise values, we have that

$$\begin{aligned}
\mathrm{Var}[R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)] &\overset{(a)}{=} \mathbb{E}[(R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q))^2] - \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)]^2 \\
&\overset{(b)}{=} \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q)^2 - 2 \cdot R_{m'}(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}', S_q) + R_{m'}(\boldsymbol{U}', S_q)^2] \\
&\overset{(c)}{=} 2 \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q)^2] - 2 \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}', S_q)],
\end{aligned}$$

where (a) holds by the definition of variance, (b) is due to the subtraction term being 0, and (c) is due to the linearity of expectation. Further, for the covariance of the difference in scores under independent generation and the difference in scores under coupled generation, we have that

$$\begin{aligned}
&\mathrm{Cov}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)] \\
&\quad \overset{(a)}{=} \mathbb{E}[(R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)) \cdot (R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q))] \\
&\qquad - \mathbb{E}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)] \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)] \\
&\quad \overset{(b)}{=} \mathbb{E}[R_m(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}, S_q)] - \mathbb{E}[R_m(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}', S_q)] \\
&\qquad - \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}, S_q)] + \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}', S_q)] \\
&\quad \overset{(c)}{=} \mathrm{Cov}[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)] - \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q)^2] + \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}', S_q)]]
\end{aligned}$$

where (a) and (c) hold by the definition of covariance and (b) is due to the last term being zero and by the expansion of the first term.

Putting all the above results together, it follows that

$$\begin{aligned}\mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}', S_q)]] &= \mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)] + 2 \cdot \mathrm{Cov}\left[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)\right] \\ &\quad - 2 \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q)^2] + 2 \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}', S_q)] \\ &\quad + 2 \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q)^2] - 2 \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q) R_{m'}(\boldsymbol{U}', S_q)] \\ &= \mathrm{Var}[R_m(\boldsymbol{U}, S_q) - R_{m'}(\boldsymbol{U}, S_q)] + 2 \cdot \mathrm{Cov}[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)]\end{aligned}$$

which concludes the proof.

D.2 Proof of Proposition 2

Due to Proposition 1, to show that Eq. 4 holds, it suffices to show that the covariance between the scores of the different LLMs under coupled generation is non-negative, *i.e.*, $\mathrm{Cov}[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)] > 0$.

To this end, we first rewrite the covariance as

$$\begin{aligned}&\mathrm{Cov}[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)] \qquad (8)\\ &\quad = P[R_m(\boldsymbol{U}, S_q) = 1, R_{m'}(\boldsymbol{U}, S_q) = 1] - P[R_m(\boldsymbol{U}, S_q) = 1] \cdot P[R_{m'}(\boldsymbol{U}, S_q) = 1] \\ &\quad = \sum_{s_q} P[S_q = s_q] \cdot (P[R_m(\boldsymbol{U}, s_q) = 1, R_{m'}(\boldsymbol{U}, s_q) = 1] \; - \; P[R_m(\boldsymbol{U}, s_q) = 1] \cdot P[R_{m'}(\boldsymbol{U}, s_q) = 1]) \qquad (9)\end{aligned}$$

Next, we note that the event $R_m(\boldsymbol{U}, s_q) = 1$ is equivalent to LLM m sampling the ground truth token for prompt s_q. Without loss of generality, assume t_1 is the ground truth token, *i.e.*, $\mathrm{C}(s_q) = t_1$. Then, since only tokens $\{t_1, t_2\}$ have positive probability under m and m', it must hold that either (i) one LLM assigns a greater probability to t_1 and the other LLM assigns a greater probability to t_2, or (ii) both LLMs assign the same probabilities. Further, since the sampling mechanism defined by f_T and P_U satisfies counterfactual stability, we have that the condition in Eq. 7 holds in both (i) and (ii) and, under coupled generation, the LLM with greater (or equal) probability for t_1 will always sample t_1 when the LLM with lower (or equal) probability does. This implies that

$$P[R_m(\boldsymbol{U}, s_q) = 1, R_{m'}(\boldsymbol{U}, s_q) = 1] = \min\{P[R_m(\boldsymbol{U}, s_q) = 1], P[R_{m'}(\boldsymbol{U}, s_q) = 1]\} \qquad (10)$$

Finally, since it holds that

$$\min\{P[R_m(\boldsymbol{U}, s_q) = 1], P[R_{m'}(\boldsymbol{U}, s_q) = 1]\} > P[R_m(\boldsymbol{U}, s_q) = 1] P[R_{m'}(\boldsymbol{U}, s_q) = 1] \qquad (11)$$

because $P[R_m(\boldsymbol{U}, s_q) = 1] \in (0, 1)$ and $P[R_{m'}(\boldsymbol{U}, s_q) = 1] \in (0, 1)$ by assumption, we can conclude from Eq. 9 that

$$\mathrm{Cov}[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)] > 0. \qquad (12)$$

D.3 Proof of Proposition 3

Using Proposition 1, we have that

$$\begin{aligned}&\mathrm{Cov}[R_m(\boldsymbol{U}, S_q), R_{m'}(\boldsymbol{U}, S_q)] \\ &\quad = \mathbb{E}[R_m(\boldsymbol{U}, S_q) \cdot R_{m'}(\boldsymbol{U}, S_q)] - \mathbb{E}[R_m(\boldsymbol{U}, S_q)] \cdot \mathbb{E}[R_{m'}(\boldsymbol{U}, S_q)] \\ &\quad = \underbrace{P[R_m(\boldsymbol{U}, S_q) = 1, R_{m'}(\boldsymbol{U}, S_q) = 1]}_{(i)} - \underbrace{P[R_m(\boldsymbol{U}, S_q) = 1) \cdot P[R_{m'}(\boldsymbol{U}, S_q) = 1]}_{(ii)}.\end{aligned}$$

In the remainder of the proof, we will bound each term (i) and (ii) separately and, since $|\mathrm{C}(s_q)| = 1$ for all $s_q \sim P_Q$, assume without loss of generality that the correct token is single-token sequence t_1.

To bound the term (ii), first note that, using the definition of the Gumbel-Max SCM, we have that, for each $k \in \{2, \ldots, |V|\}$, it holds that

$$R_m(\boldsymbol{U}, s_q) = 1 \iff U_1 + \log([f_D(s_q, m)]_{t_1}) \geq U_k + \log([f_D(s_q, m)]_{t_k}),$$
$$R_{m'}(\boldsymbol{U}, s_q) = 1 \iff U_1 + \log([f_D(s_q, m')]_{t_1}) \geq U_k + \log([f_D(s_q, m')]_{t_k}).$$

Next, let $\varepsilon^* > 0$ be an arbitrary constant that we will determine later such that

$$|\log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m')]_{t_k})| \leq \varepsilon^*, \tag{13}$$

and note that since, by assumption, $D_{t_k} > 0$ for all $k \in \{1, \ldots, |V|\}$, any bound on the absolute difference of log-probabilities $|\log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m')]_{t_k})|$ uniformly implies a bound on the difference of probabilities $|[f_D(S_q, m)]_{t_k} - [f_D(S_q, m')]_{t_k}|$ and vice versa. For simplicity, we prove the result in the log-domain.

Now, using the bound defined by Eq. 13, we have that

$$\bigcap_{k \neq 1} \{U_1 + \log([f_D(S_q, m')]_{t_1}) \geq U_k + \log([f_D(S_q, m')]_{t_k})\}$$
$$\subset \bigcap_{k \neq 1} \{U_1 + \log([f_D(S_q, m)]_{t_1}) + \varepsilon^* \geq U_k + \log([f_D(S_q, m)]_{t_k}) - \varepsilon^*\},$$

and we can then bound the term (ii) as follows:

$$\begin{aligned}
&P[R_m(\boldsymbol{U}, S_q) = 1] \cdot P[R_{m'}(\boldsymbol{U}, S_q) = 1] \\
&\quad = P[\cap_{k \neq 1}\{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k})\}] \\
&\qquad \times P[\cap_{k \neq 1}\{U_1 + \log([f_D(S_q, m')]_{t_1}) \geq U_k + \log([f_D(S_q, m')]_{t_k})\}] \\
&\quad \leq P[\cap_{k \neq 1}\{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k})\}] \\
&\qquad \times P[\cap_{k \neq 1}\{U_1 + \log([f_D(S_q, m)]_{t_1}) + \varepsilon^* \geq U_k + \log([f_D(S_q, m)]_{t_k}) - \varepsilon^*\}].
\end{aligned}$$

To bound the term (i), first note that, using the bound defined by Eq. 13, we have that

$$\bigcap_{k \neq 1} \{U_1 + \log([f_D(S_q, m')]_{t_1}) \geq U_k + \log([f_D(S_q, m')]_{t_k})\}$$
$$\supset \bigcap_{k \neq 1} \{U_1 + \log([f_D(S_q, m)]_{t_1}) - \varepsilon^* \geq U_k + \log([f_D(S_q, m)]_{t_k}) + \varepsilon^*\}.$$

Thus, we can bound the term (i) as follows:

$$\begin{aligned}
&P[R_m(\boldsymbol{U}, S_q) = 1, R_{m'}(\boldsymbol{U}, S_q) = 1] \\
&\quad = P\Big[\cap_{k \neq 1} \{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k})\} \\
&\qquad \cap \{U_1 + \log([f_D(S_q, m')]_{t_1}) \geq U_k + \log([f_D(S_q, m')]_{t_k})\}\Big] \\
&\quad \geq P\Big[\cap_{k \neq 1} \{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k})\} \\
&\qquad \cap \{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k}) + 2\varepsilon^*\}\Big] \\
&\quad \overset{(a)}{=} \sum_{s_q} P[S_q = s_q] \cdot P[\cap_{k \neq 1}\{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k}) + 2\varepsilon^*\}],
\end{aligned}$$

where (a) follows from the fact that

$$\{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k}) + 2\varepsilon^*\}$$
$$\subset \{U_1 + \log([f_D(S_q, m)]_{t_1}) \geq U_k + \log([f_D(S_q, m)]_{t_k})\}.$$

Now, note that, for $k \in \{2, \dots, |V|\}$, the variable $X_k \equiv U_1 - U_k \sim \text{Logistic}(0,1)$ (for $k = 1$, define $X_k \equiv 0$). Therefore, we can rewrite the bound for (i) as

$$P[R_m(\boldsymbol{U}, S_q) = 1, R_{m'}(\boldsymbol{U}, S_q) = 1] \geq \sum_{s_q} P[S_q = s_q] \cdot \prod_{k \neq 1} \cdot P[\{X_k \geq \log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m)]_{t_1}) + 2\varepsilon^*\}]$$

and we can rewrite the bound for (ii) as

$$P[R_m(\boldsymbol{U}, S_q) = 1]P[R_{m'}(\boldsymbol{U}, S_q) = 1] \leq \sum_{s_q} P[S_q = s_q] \cdot \left\{ \prod_{k \neq 1} P[\{X_k \geq \log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m)]_{t_k}) - 2\varepsilon^*\}] \right\} \times P[\cap_{k \neq 1} \{X_k \geq \log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m)]_{t_k})\}].$$

As a consequence, to prove that $P[R_m(\boldsymbol{U}, S_q) = 1, R_{m'}(\boldsymbol{U}, S_q) = 1] > P[R_m(\boldsymbol{U}, S_q) = 1]P[R_{m'}(\boldsymbol{U}, S_q) = 1]$, it suffices to show that

$$\sum_{s_q} P[S_q = s_q] \prod_{k \neq 1} \cdot P[\{X_k \geq \log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m)]_{t_1}) + 2\varepsilon^*\}] > \sum_{s_q} P[S_q = s_q] \prod_{k \neq 1} \cdot P[\{X_k \geq \log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m)]_{t_1}) - 2\varepsilon^*\}] \times P[\cap_{k \neq 1} \{X_k \geq \log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m)]_{t_1})\}] \quad (14)$$

To do so, note that Eq. 14 holds trivially for $\varepsilon^* = 0$ since

$$P[\cap_{k \neq 1} \{X_k \geq \log([f_D(S_q, m)]_{t_k}) - \log([f_D(S_q, m)]_{t_1})\}] < 1,$$

which is a fixed term independent of m'. Since all terms in Eq. 14 are continuous in ε^*, there exists $\varepsilon^*(m) > 0$, possibly dependent of m but independent of m', such that Eq. 14 holds if

$$\sup_{s_q} \|\log(f_D(s_q, m)) - \log(f_D(s_q, m'))\|_\infty < \varepsilon^*(m).$$

Since by assumption $D_t > 0$ for all $t \in V$, there exists $\varepsilon(m) > 0$ in probability space such that Eq. 14 holds if

$$\sup_{s_q} \|f_D(s_q, m) - f_D(s_q, m')\|_\infty < \varepsilon(m).$$

This concludes the proof.

D.4 Proof of Proposition 4

Under coupled autoregressive generation, if the LLM m samples the preferred token t_+, then the LLM m' must also sample t_+ because t_+ is more likely under m' than under m and the sampling mechanism defined by f_T and P_U satisfies counterfactual stability. This implies that the win-rate achieved by m against m' is

$$\mathbb{E}_{\boldsymbol{U} \sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) > R_{m'}(\boldsymbol{U}, s_q)\}] = P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_+, f_T(f_D(s_q, m'), \boldsymbol{U}) = t_-] = 0 \quad (15)$$

and that

$$P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_+, f_T(f_D(s_q, m'), \boldsymbol{U}) = t_+] = P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_+] = p_m. \quad (16)$$

Using the same reasoning, if the LLM m' samples the non-preferred token t_-, then, m must also sample t_- because t_- is more likely under m than under m'. This implies that

$$P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_-, f_T(f_D(s_q, m'), \boldsymbol{U}) = t_-] = P[f_T(f_D(s_q, m'), \boldsymbol{U}) = t_-] = 1 - p_{m'} \quad (17)$$

Then, from Eq. 16 and Eq. 17, we can conclude that

$$\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}, s_q)\}] = p_m + (1 - p_{m'}) \tag{18}$$

Finally, from Eq. 15 and Eq. 18, we can conclude that the win-rate achieved by m' against m is

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) < R_{m'}(\boldsymbol{U}, s_q)\}] \\
&\quad = 1 - \mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) > R_{m'}(\boldsymbol{U}, s_q)\}] - \mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}, s_q)\}] \\
&\quad = p_{m'} - p_m.
\end{aligned}$$

Under independent autoregressive generation, the LLMs m and m' sample tokens independently from each other, *i.e.*, $f_T(f_D(s_q, m), \boldsymbol{U}) \perp f_T(f_D(s_q, m'), \boldsymbol{U}')$. Thus, we can factorize all joint probabilities when computing the win-rates and obtain

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'\sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) > R_{m'}(\boldsymbol{U}', s_q)\}] &= P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_+] \cdot P[f_T(f_D(s_q, m'), \boldsymbol{U}') = t_-] \\
&= p_m \cdot (1 - p_{m'})
\end{aligned}$$

and

$$\mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'\sim P_{\boldsymbol{U}}}[\mathbf{1}\{R_m(\boldsymbol{U}, s_q) < R_{m'}(\boldsymbol{U}', s_q)\}] = p_{m'} \cdot (1 - p_m).$$

D.5 Proof of Proposition 5

We follow the notations and technique of Proposition 3. Fix query s_q and consider first the case of independent autoregressive generation. Since each LLM can only assign a non-zero probability to single-token sequences, we have:

$$\begin{aligned}
P[R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}', s_q)] &= \sum_{k=1}^{|V|} P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_k] \cdot P[f_T(f_D(s_q, m'), \boldsymbol{U}) = t_k] \\
&< \sum_{k=1}^{|V|} P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_k],
\end{aligned}$$

In the case of coupled autoregressive generation, since

$$P\left[\{f_T(f_D(s_q, m), \boldsymbol{U}) = t_k\} \cap \{f_T(f_D(s_q, m), \boldsymbol{U}) = t_j\}\right] = 0,\ i \neq j,$$

we obtain:

$$\begin{aligned}
&P[R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}, s_q)] \\
&\quad = P\left[\cup_i \{f_T(f_D(s_q, m), \boldsymbol{U}) = t_k, f_T(f_D(s_q, m'), \boldsymbol{U}) = t_k\}\right] \\
&\quad = \sum_k P[\{f_T(f_D(s_q, m), \boldsymbol{U}) = t_k, f_T(f_D(s_q, m'), \boldsymbol{U}) = t_k\}] \\
&\quad = \sum_k P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_k] \cdot P[f_T(f_D(s_q, m'), \boldsymbol{U}) = t_k | f_T(f_D(s_q, m), \boldsymbol{U}) = t_k].
\end{aligned}$$

We now follow Huijben et al. (2023) and expand the posterior Gumbels, $P[f_T(f_D(s_q, m'), \boldsymbol{U}) = t_k \mid f_T(f_D(s_q, m), \boldsymbol{U}) = t_k]$, as truncated Gumbel distributions. In particular, we leverage the fact that

$$\max_{t\in V}\{U_t + \log([f_D(s_q, \bullet)]_t)\} \sim \text{Gumbel}(0, 1), \tag{19}$$

and that a Gumbel distribution, with parameter $\log(\theta)$, truncated at $b \sim \text{Gumbel}(0, 1)$ can be sampled as

$$-\log(\exp(-b) - \log(\eta)/\theta),\ \eta \sim U(0, 1). \tag{20}$$

Furthermore, by assumption, $D_{t_k} > 0$ for all $k \in \{1, \dots, |V|\}$, so that any bound on the absolute difference of log-probabilities $|\log([f_D(s_q, m)]_{t_k}) - \log([f_D(s_q, m')]_{t_k})|$ uniformly implies a bound on the difference of probabilities $|[f_D(s_q, m)]_{t_k} - [f_D(s_q, m')]_{t_k}|$ and vice versa. Using the bound

$$|\log([f_D(s_q, m)]_{t_k}) - \log([f_D(s_q, m')]_{t_k})| \leq \varepsilon^*$$

and the Gumbel properties in Eq. 19 and Eq. 20, we obtain:

$$\begin{aligned}
&P[R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}, s_q)] \\
&= \sum_k P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_k] \\
&\qquad \times P\Bigg[\bigcap_k \Big\{ \log([f_D(s_1, m')]_{t_k}) - \log([f_D(s_1, m)]_{t_k}) - \log(-\log(\eta_k)) \\
&\qquad\qquad \geq \log([f_D(s_1, m')]_{t_j}) - \log([f_D(s_1, m)]_{t_j}) - \log(-\log(\eta_k) - \log(\eta_j)/[f_D(s_1, m')]_{t_j}) \Big\}\Bigg] \\
&\geq \sum_k P[f_T(f_D(s_q, m), \boldsymbol{U}) = t_k] \\
&\qquad \times P\left[\cap_k \{-\log(-\log(\eta_k)) \geq -2\varepsilon^* - \log(-\log(\eta_k) - \log(\eta_j)/[f_D(s_1, m')]_{t_j})\}\right]
\end{aligned} \tag{21}$$

where $\eta_k \sim \mathrm{U}(0, 1)$ are independently distributed uniform random variables. Now, note that the claim holds for $\varepsilon^* = 0$ since, in that case, we have that

$$P\left[\bigcap_k \{-\log(-\log(\eta_k)) \geq -\log(-\log(\eta_k) - \log(\eta_k)/[f_D(s_1, m')]_{t_k})\}\right] = 1,$$

using that $x \mapsto -\log(x)$ is strictly decreasing. Since all terms in Eq. 21 are continuous in ε^*, there exists $\varepsilon^*(m) > 0$, possibly dependent on m but independent of m', such that

$$P[R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}, s_q)] > P[R_m(\boldsymbol{U}, s_q) = R_{m'}(\boldsymbol{U}', s_q)] \tag{22}$$

holds if

$$\sup_{s_q} \|\log(f_D(s_q, m)) - \log(f_D(s_q, m'))\|_\infty < \varepsilon^*(m).$$

Since by assumption $D_t > 0$ for all $t \in V$, there exists $\varepsilon(m) > 0$ in probability space such that Eq. 22 holds if

$$\sup_{s_q} \|f_D(s_q, m) - f_D(s_q, m')\|_\infty < \varepsilon(m).$$

This concludes the proof.

D.6 Calculation of average win-rates in the example used in Sections 1 and 4

In this section, we provide detailed calculations of the win-rates for the example in Sections 1 and 4. Recall that in this example, we are given three LLMs m_1, m_2 and m_3, and we need to rank them according to their ability to answer correctly two types of input prompts, q and q', picked uniformly at random. We assume that the true probability that each LLM answers correctly each type of input prompt is given by:

	m_1	m_2	m_3
q	$p_1 = 0.4$	$p_2 = 0.48$	$p_3 = 0.5$
q'	$p'_1 = 1$	$p'_2 = 0.9$	$p'_3 = 0.89$

Using Proposition 4, the win-rates under independent autoregressive generation are given, for each LLM m_k, by:

$$\frac{1}{2} \sum_{j \neq k} \mathbb{E}_{\boldsymbol{U}, \boldsymbol{U}' \sim P_{\boldsymbol{U}}, S_q \sim P_Q}[\mathbf{1}\{R_{m_k}(\boldsymbol{U}, S_q) > R_{m_j}(\boldsymbol{U}', S_q)\}] = \frac{\sum_{j \neq k} p_k(1 - p_j) + \sum_{j \neq k} p'_k(1 - p'_j)}{4}. \tag{23}$$

Substituting the numerical values we obtain:

$$\begin{aligned}
&\frac{1}{2}\sum_{j\neq 1}\mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_{m_1}(\boldsymbol{U},S_q)>R_{m_j}(\boldsymbol{U}',S_q)\}]=0.1545,\\
&\frac{1}{2}\sum_{j\neq 2}\mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_{m_2}(\boldsymbol{U},S_q)>R_{m_j}(\boldsymbol{U}',S_q)\}]=0.15675,\\
&\frac{1}{2}\sum_{j\neq 3}\mathbb{E}_{\boldsymbol{U},\boldsymbol{U}'\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_{m_3}(\boldsymbol{U},S_q)>R_{m_j}(\boldsymbol{U}',S_q)\}]=0.16225
\end{aligned} \tag{24}$$

Similarly, using Proposition 4, the win-rates using coupled autoregressive generation can be written, for each LLM m_k, as:

$$\frac{1}{2}\sum_{j\neq k}\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_{m_k}(\boldsymbol{U},S_q)>R_{m_j}(\boldsymbol{U},S_q)\}]=\frac{\sum_{j\neq k}(p_k-p_j)_+ + \sum_{j\neq k}(p'_k-p'_j)_+}{4}, \tag{25}$$

where $(\bullet)_+ = \max(0,\bullet)$ denotes the positive part. Substituting the numerical values we obtain:

$$\begin{aligned}
&\frac{1}{2}\sum_{j\neq 1}\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_{m_1}(\boldsymbol{U},S_q)>R_{m_j}(\boldsymbol{U},S_q)\}]=0.0525,\\
&\frac{1}{2}\sum_{j\neq 2}\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_{m_2}(\boldsymbol{U},S_q)>R_{m_j}(\boldsymbol{U},S_q)\}]=0.0225,\\
&\frac{1}{2}\sum_{j\neq 3}\mathbb{E}_{\boldsymbol{U}\sim P_{\boldsymbol{U}},S_q\sim P_Q}[\mathbf{1}\{R_{m_3}(\boldsymbol{U},S_q)>R_{m_j}(\boldsymbol{U},S_q)\}]=0.03.
\end{aligned}$$

E ADDITIONAL EXPERIMENTAL DETAILS

Hardware setup. Our experiments are executed on a compute server equipped with 2 × Intel Xeon Gold 5317 CPU, 1,024 GB main memory, and 2 × A100 Nvidia Tesla GPU (80 GB, Ampere Architecture). In each experiment a single Nvidia A100 GPU is used.

Datasets. As benchmark datasets, we use (i) the Measuring Massive Multitask Language Understanding dataset (MMLU) (Hendrycks et al., 2021b), (ii) the validation data split of the Grade School Math 8K dataset (GSM8K) (Cobbe et al., 2021), and (iii) the HumanEval dataset (Chen et al., 2021). MMLU consists of 14,042 questions covering 52 diverse knowledge areas with each question offering four possible choices indexed from A to D, and a ground-truth answer. The validation data split of the GSM8K dataset comprises 1,319 grade school math problems along with their full solution as a string. The HumanEval dataset includes 164 programming problems with a function signature, docstring, body, and several unit tests. For pairwise comparison tasks, we use the first 500 questions from the LMSYS-Chat-1M dataset (Zheng et al., 2024).

Models. In our experiments with the `Llama` family of models, we use `Llama-3.1-8B-Instruct`, its quantized variants `Llama-3.1-8B-Instruct-{AWQ-INT4, bnb-4bit, bnb-8bit}` and `Llama-3.2-{1B, 3B}-Instruct` models. In our experiments with the `Qwen` family of models, we use `Qwen2.5-7B-Instruct`, its quantized variants `Qwen2.5-7B-Instruct-{AWQ-INT4, bnb-4bit, bnb-8bit}`, `Qwen2.5-3B-Instruct`, a distilled variant `DistilQwen2.5-3B-Instruct`, `Qwen2.5-1B-Instruct` and `Qwen3-8B`. In our experiments with the `Mistral` family of models, we use `Mistral-7B-Instruct-v0.3`, its quantized variants `Mistral-7B-Instruct-v0.3-{bnb-4bit, bnb-8bit}` and `Mistral-7B-Instruct-{v0.1,v0.2}`. The models are obtained from `Hugging Face`, and the quantized `bnb-4bit, bnb-8bit` LLM variants are built using the `bitsandbytes` library (Bits and Bytes Foundation, 2024). The token vocabulary of `Mistral-7B-Instruct-v0.3` contains 768 additional tokens compared to `Mistral-7B-Instruct-{v0.1,v0.2}`, and `Qwen3-8B`'s vocabulary contains 4 additional tokens compared to the rest of the models in the `Qwen` family. Such differences correspond to additional control tokens used to, for example, allow models to access external information retrieval tools, and are therefore neither sampled nor relevant in our experiments. In our experiments, within each model family, we sample from the larger vocabulary and set the probability of tokens that are not included in the smaller vocabulary to zero when using the respective models.

Parameters. For the benchmark tasks, we set the temperature to 0.7 unless specified otherwise (Figure 8), and, for the pairwise comparison tasks, we set it to 1. When generating with `Qwen3-8B`, we disable thinking mode.

Prompts. To instruct LLMs for generating output, we use the system prompt in Table 1 for the MMLU dataset, the system prompt in Table 2 for the GSM8K data, and the system prompt in Table 3 for the LMSYS-Chat-1M dataset. For the HumanEval dataset we use no system prompt, but rather the (input) prompt and response prefixes shown in Table 4.[9] Further, to perform pairwise comparisons of outputs of different LLMs, we use the system prompt in Table 5, which is adapted from Chiang et al. (2025), to prompt the strong LLM.

Licenses. The MMLU, GSM8K and HumanEval datasets are licensed under MIT, and the LMSYS-Chat-1M dataset is licensed under the LMSYS-Chat-1M Dataset License Agreement.[10] The `Llama-3.1-8B-Instruct` model, its quantized version `Llama-3.1-8B-Instruct-AWQ-INT4` and the prompt used for the HumanEval dataset are licensed under the LLAMA 3.1 COMMUNITY LICENSE AGREEMENT.[11]. The `Llama-3.2-{1B, 3B}-Instruct` models are licensed under the LLAMA 3.2 COMMUNITY LICENSE AGREEMENT.[12]. The models in the `Qwen` and `Mistral` families are licensed under Apache 2.0.

[9] We chose the prefixes following the evaluation details on the HumanEval dataset for `8B` available in `https://huggingface.co/datasets/meta-llama/Llama-3.1-8B-Instruct-evals`

[10] `https://huggingface.co/datasets/lmsys/lmsys-chat-1m`

[11] `https://www.llama.com/llama3_1/license/`

[12] `https://www.llama.com/llama3_2/license/`

System: You will be given multiple choice questions. Please reply with a single character 'A', 'B', 'C', or 'D' only. DO NOT explain your reply.

Table 1: System prompt used for the MMLU dataset.

System: You will be given a mathematical problem. Please reply only with a single number as your answer.

Table 2: System prompt used for the GSM8K dataset.

System: Keep your responses short and to the point.

Table 3: System prompt used for the LMSYS Chatbot Arena dataset.

Prompt Prefix:

Write a solution to the following problem and make sure that it passes the tests:
"'python

Response Prefix:

Here is the completed function:
"'python

Table 4: Prompt and response prefix for the HumanEval dataset.

System: Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user prompt displayed below. Your job is to evaluate which assistant's answer is better. When evaluating the assistants' answers, compare both assistants' answers. You must identify and correct any mistakes or inaccurate information. Then consider if the assistant's answers are helpful, relevant, and concise. Helpful means the answer correctly responds to the prompt or follows the instructions. Note when user prompt has any ambiguity or more than one interpretation, it is more helpful and appropriate to ask for clarifications or more information from the user than providing an answer based on assumptions. Relevant means all parts of the response closely connect or are appropriate to what is being asked. Concise means the response is clear and not verbose or excessive. Then consider the creativity and novelty of the assistant's answers when needed. Finally, identify any missing important information in the assistants' answers that would be beneficial to include when responding to the user prompt. do not provide any justification or explanation for your response. You must output only one of the following choices as your final verdict:

'A' if the response of assistant A is better

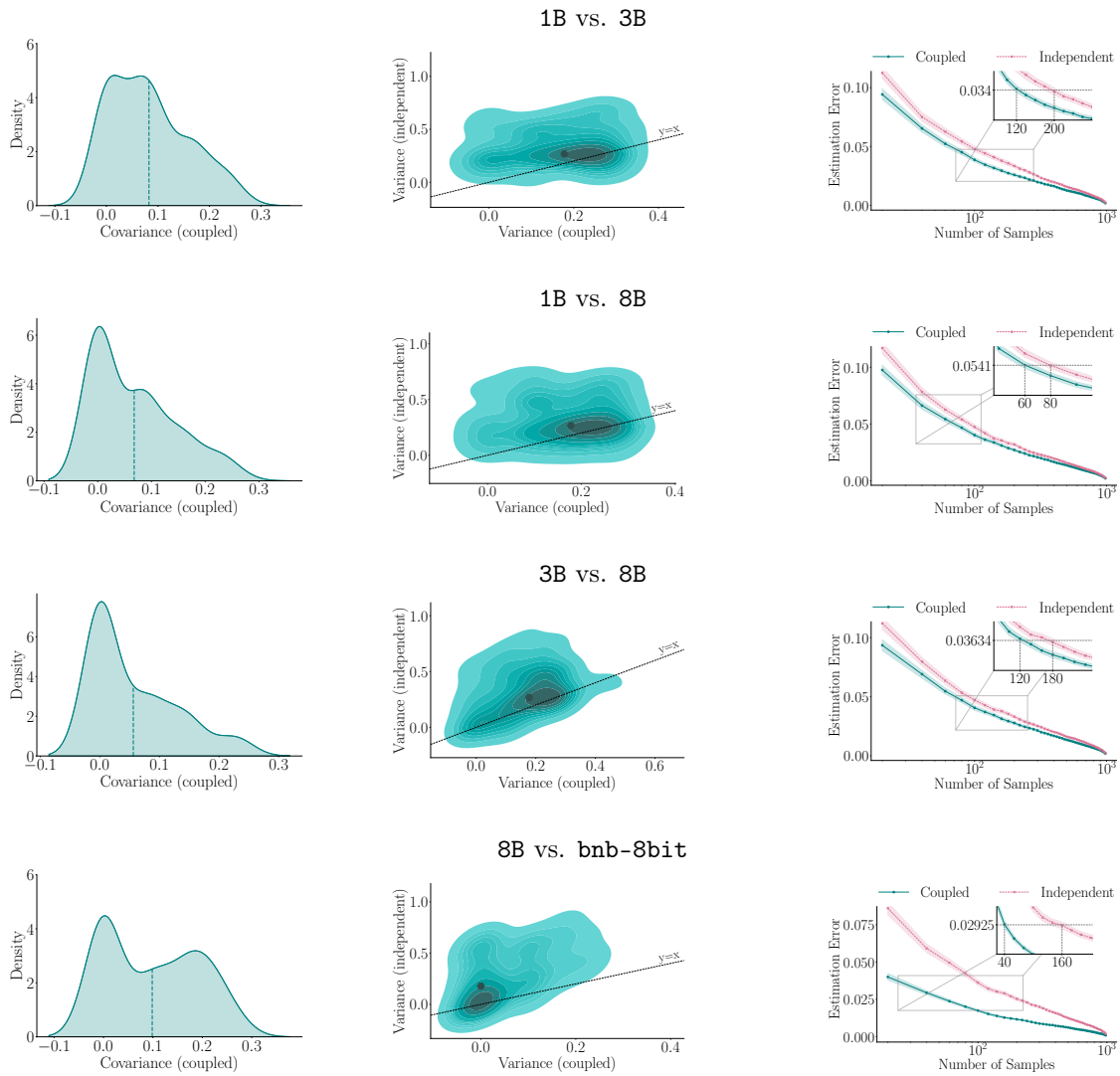
'B' if the response of assistant B is better

'Tie' if the responses are tied

Table 5: System prompt used for obtaining pairwise preferences using `GPT-4o-2024-11-20` as the judge.

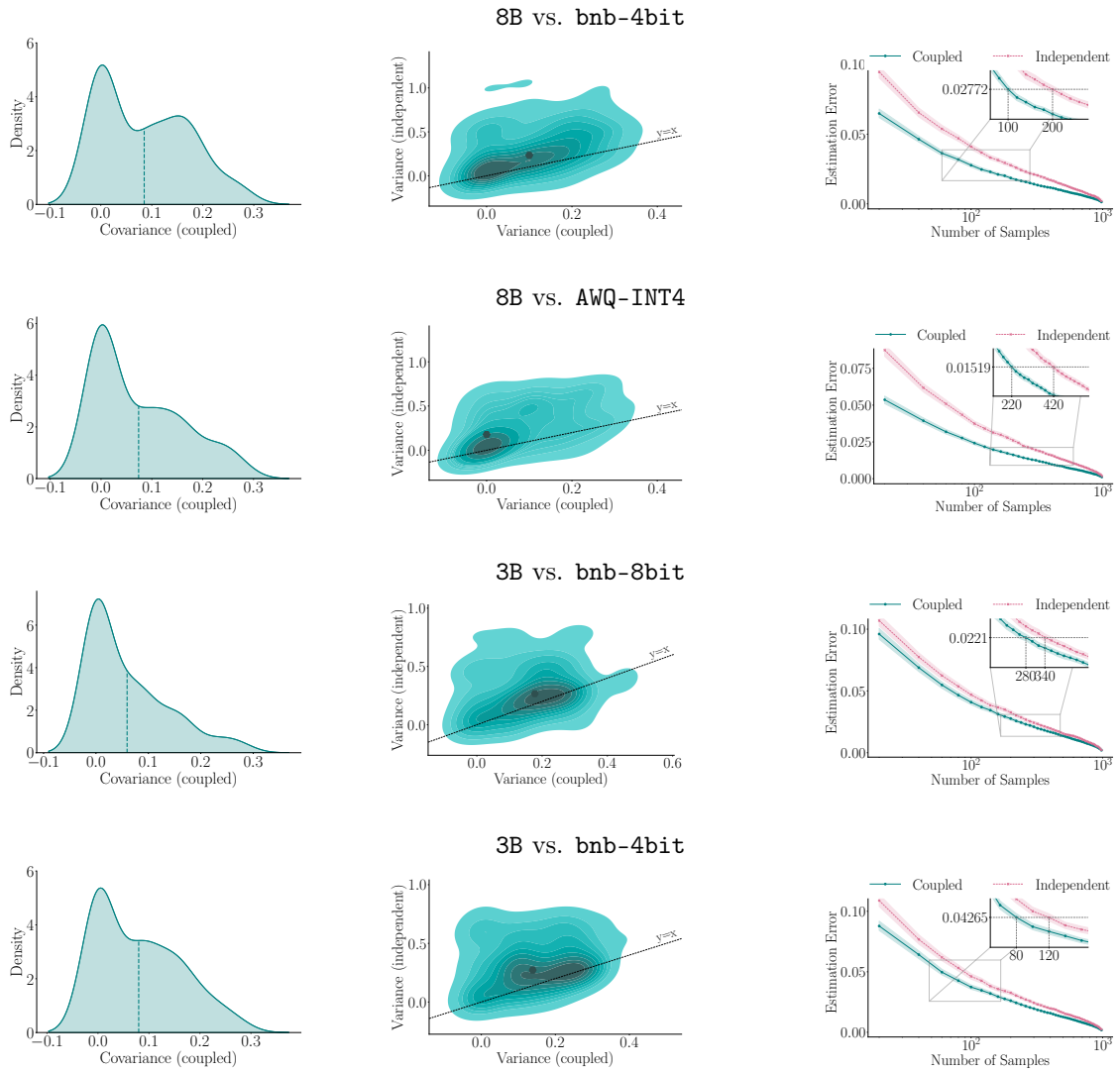
F ADDITIONAL EXPERIMENTS WITH LLMS IN THE `Llama` FAMILY

F.1 MMLU Dataset



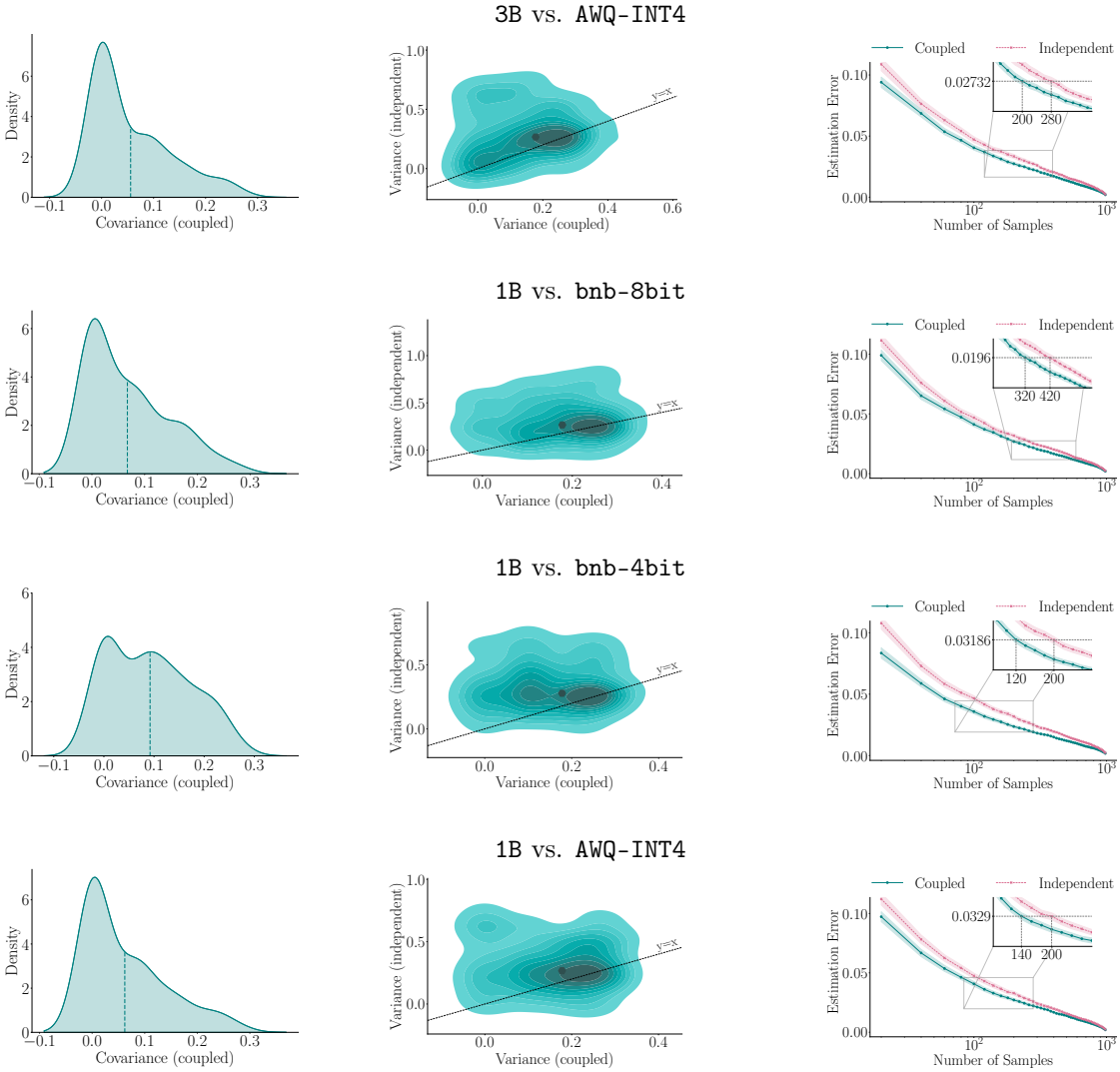
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 4: **Comparison between four pairs of LLMs in the `Llama` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



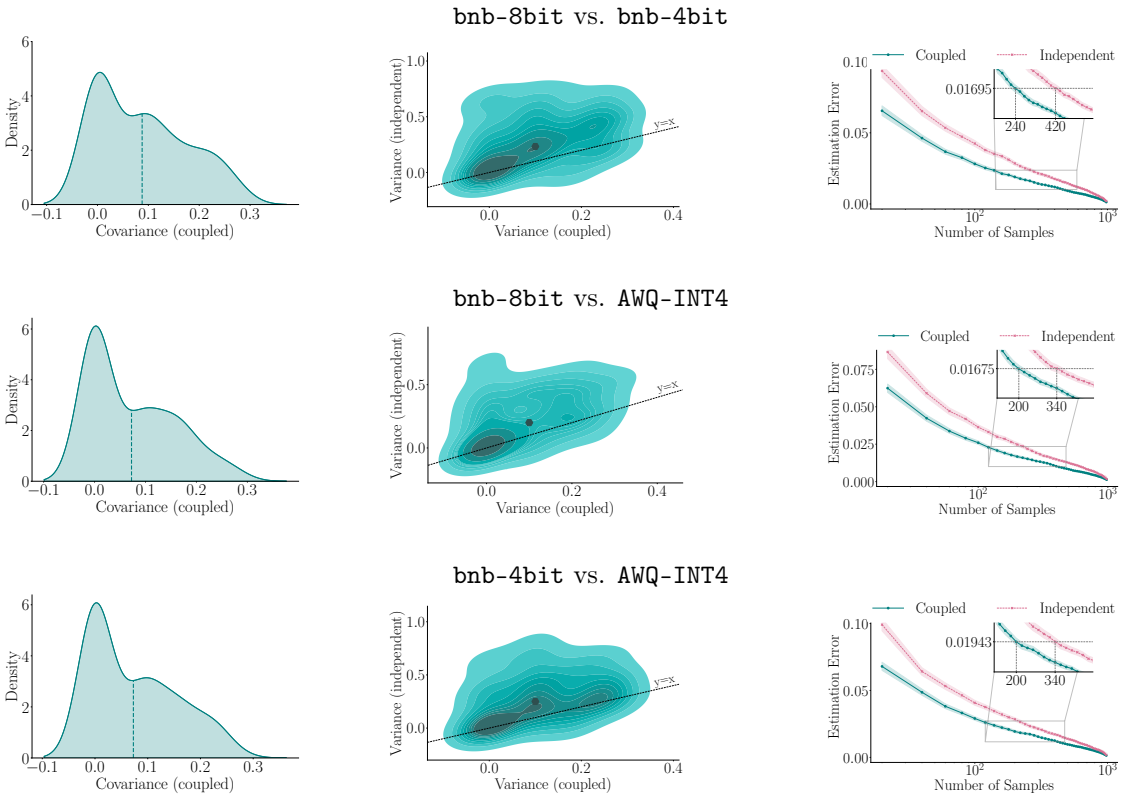
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 5: **Comparison between four pairs of LLMs in the `Llama` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



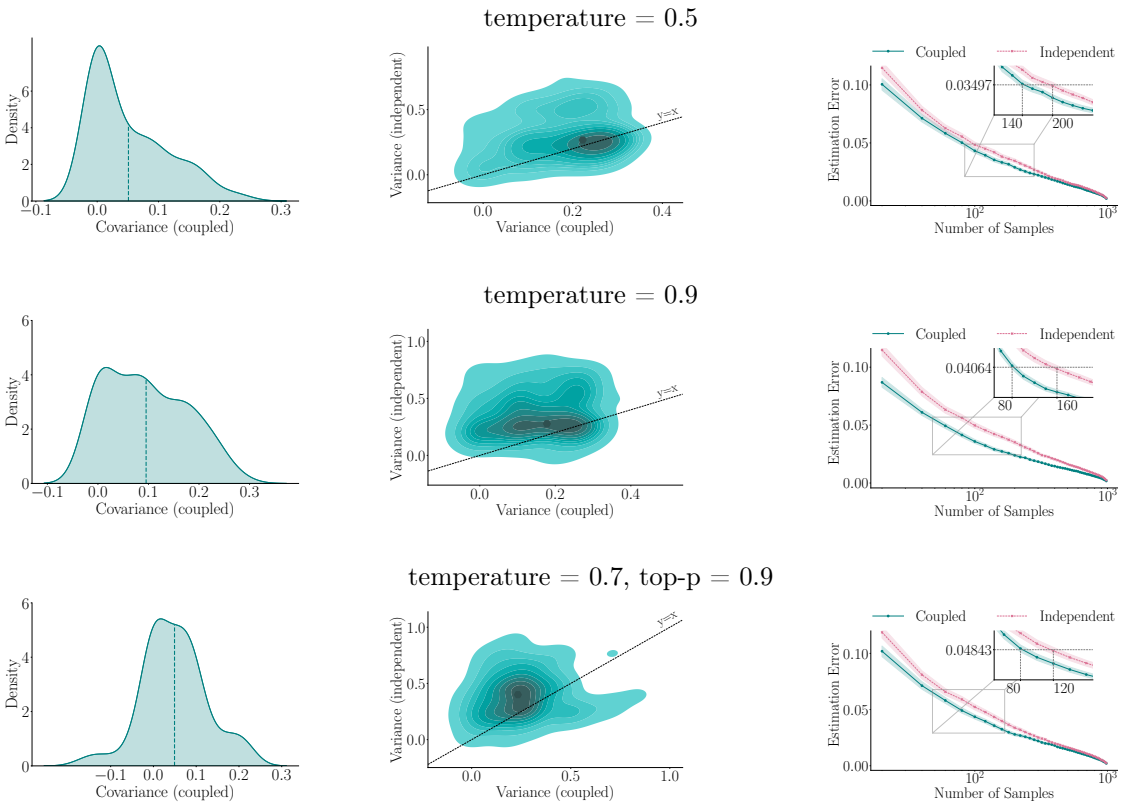
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 6: **Comparison between four pairs of LLMs in the `Llama` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



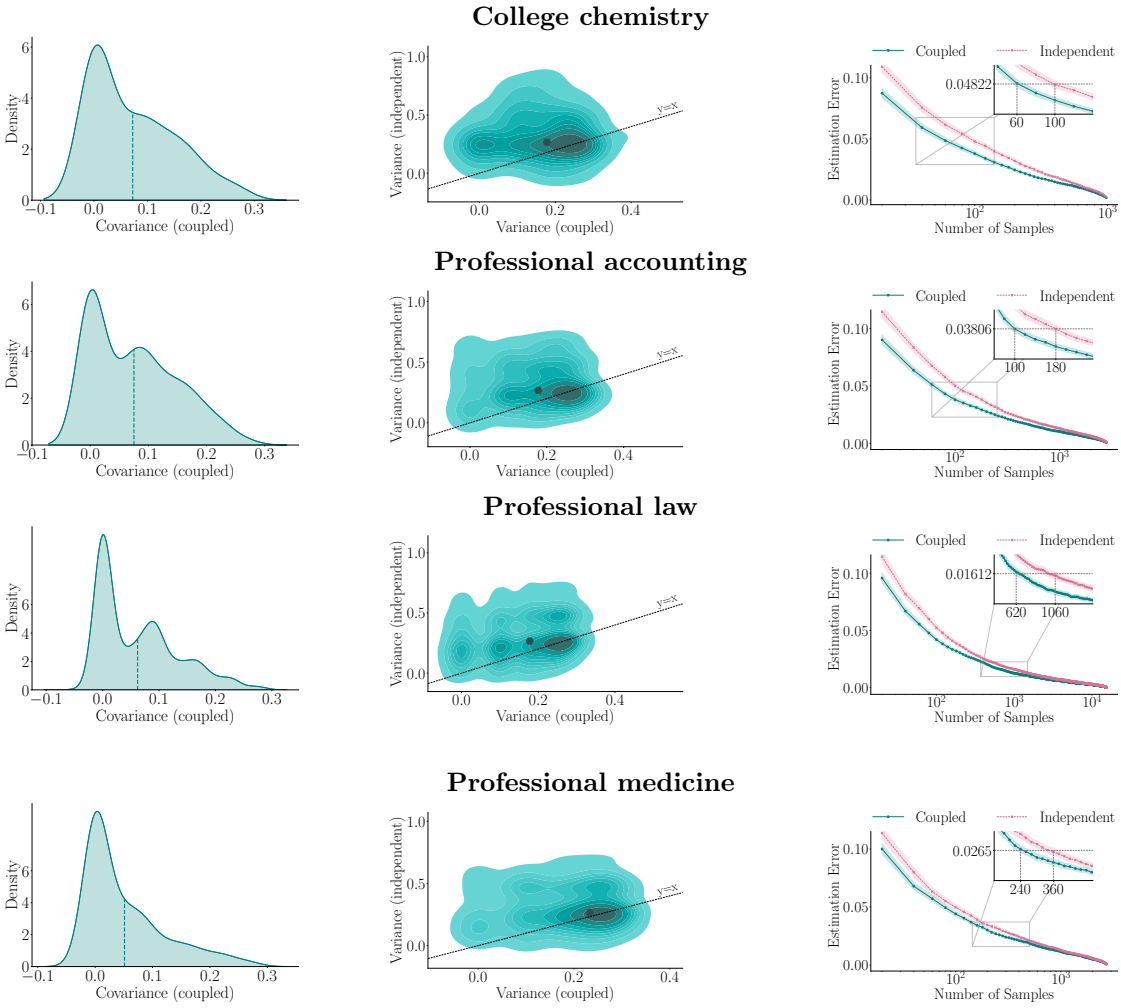
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 7: **Comparison between three pairs of LLMs in the `Llama` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 8: **Comparison Comparison between `1B` and `3B` from the `Llama` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset under different sampling parameters.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 9: **Comparison between `1B` and `3B` from the `Llama` family on multiple-choice questions from four knowledge areas of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals. We observe qualitatively similar results for other knowledge areas.

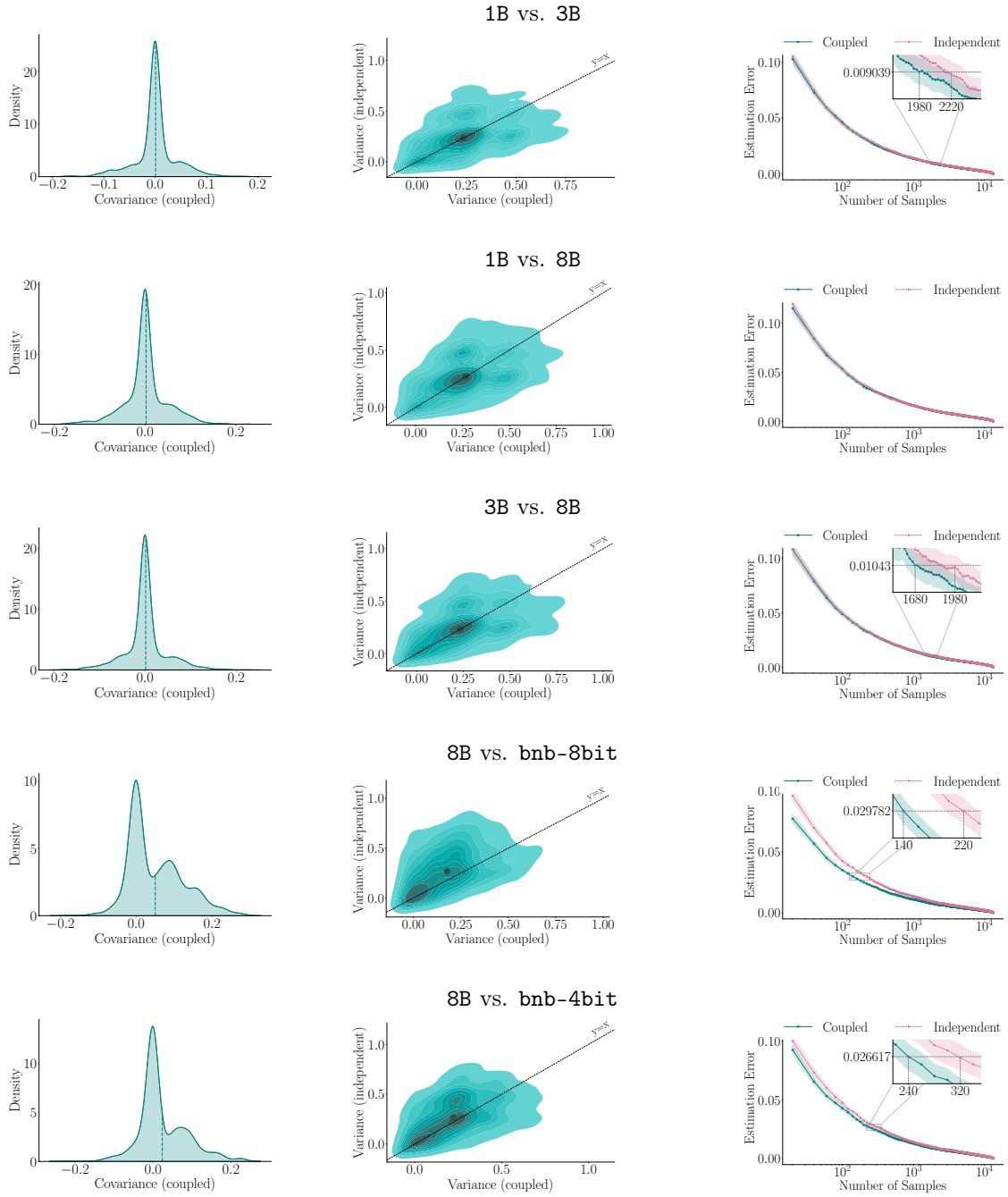
F.2 GSM8K and HumanEval Datasets

Experimental Setup for GSM8K. We provide each math problem in the GSM8K validation set as an input prompt to the LLMs. Further, we instruct the LLMs through a system prompt to reply with a single numerical value—see Table 2 for the exact prompt. To evaluate the outputs provided by each LLM, we use a binary score $R \in \{0, 1\}$, which indicates whether the last numerical value in the LLM output is the correct ($R = 1$) or incorrect ($R = 0$) solution to the math problem. To obtain reliable conclusions, we experiment with each math problem 10 times, each time using a (different) random seed to generate the noise variables used by the sampler. In what follows, we compare all pairs of LLMs described in Section 5.

Experimental Setup for HumanEval. We prepend the input prefix in Table 4 to each programming problem in HumanEval dataset and provide it as an input to the LLMs. Further, we append each programming problem to the response prefix in Table 4 and provide it as an output prefix, so that the LLMs generate responses that continue the given output prefix (instead of generating a new response). To evaluate the outputs provided by each LLM, we use a binary score $R \in \{0, 1\}$, which indicates whether the program given in each LLM output is the correct ($R = 1$) or incorrect ($R = 0$) solution to the programming problem. To parse and evaluate the program in each LLM output, we use the evaluation harness by Chen et al. (2021). To obtain reliable conclusions, we experiment with each programming problem 10 times, each time using a (different) random seed to generate the noise variables used by the sampler. In what follows, we compare all pairs of LLMs described in Section 5.

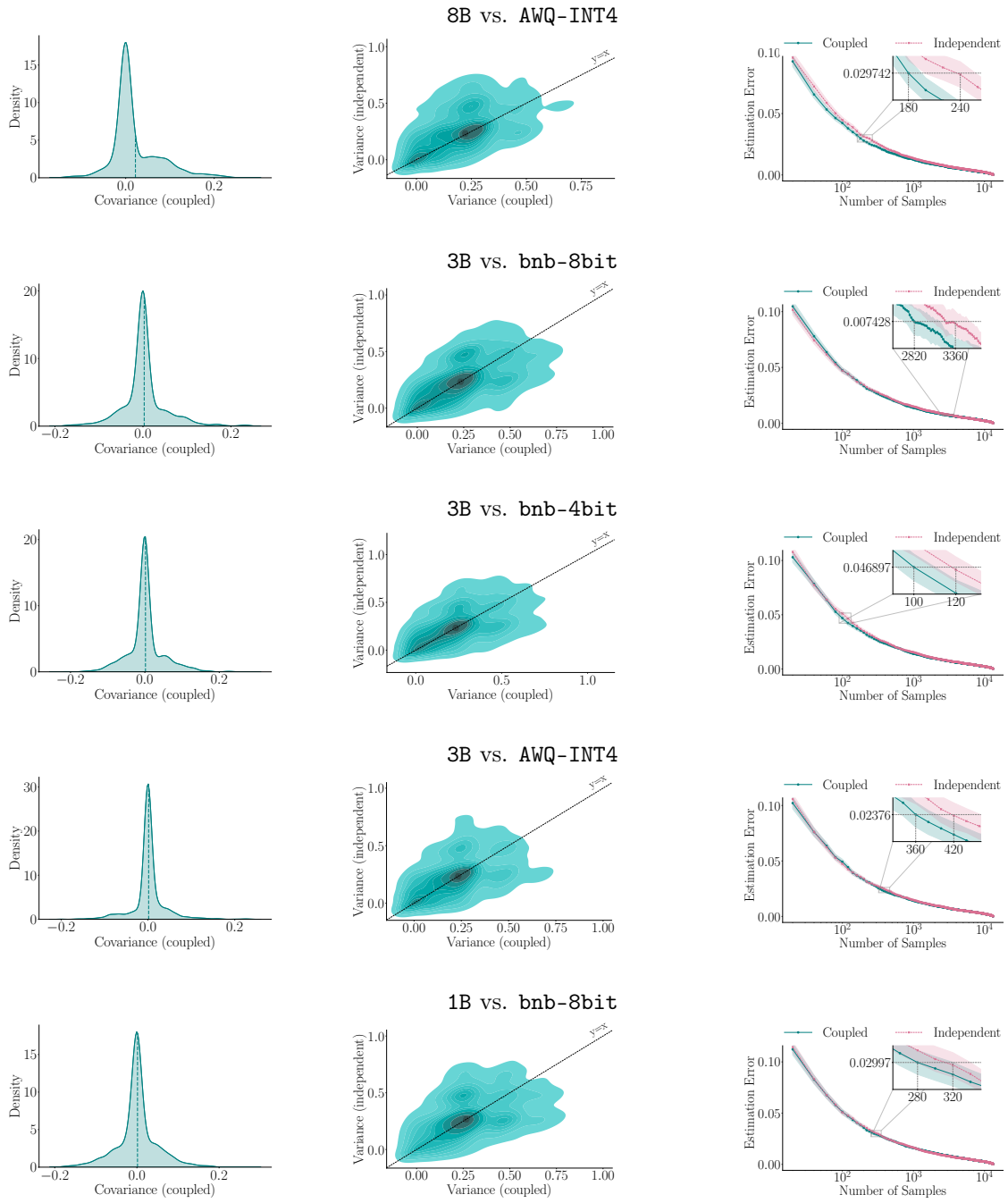
Results. Figures 10- 15 show the results for models in the `Llama` family on the GSM8K and HumanEval datasets. On both datasets, we find that sufficiently similar LLMs, for example, `8B` and `bnb-8bit`, have positively correlated scores under coupled generation and thus the variance of the difference in scores is lower under coupled generation than under independent, in agreement with Proposition 1 and 3. These pairs of LLMs also require fewer samples under coupled generation than under independent to achieve equivalent error in the estimation of the expected difference between the scores of the LLMs.[13] Other pairs of LLMs have non-positively correlated scores, resulting in similar variance under coupled and independent generation, and requiring the same number of samples to achieve equivalent error in the estimation of the expected difference between the scores of the LLMs.

[13]For each dataset, we compute the error in the estimation of the expected difference in scores as a function of the available sample size as in Section 5.1; here, we use 13,190 samples from GSM8K and 1,680 samples from HumanEval to estimate (proxies of) the ground truth expected score difference.



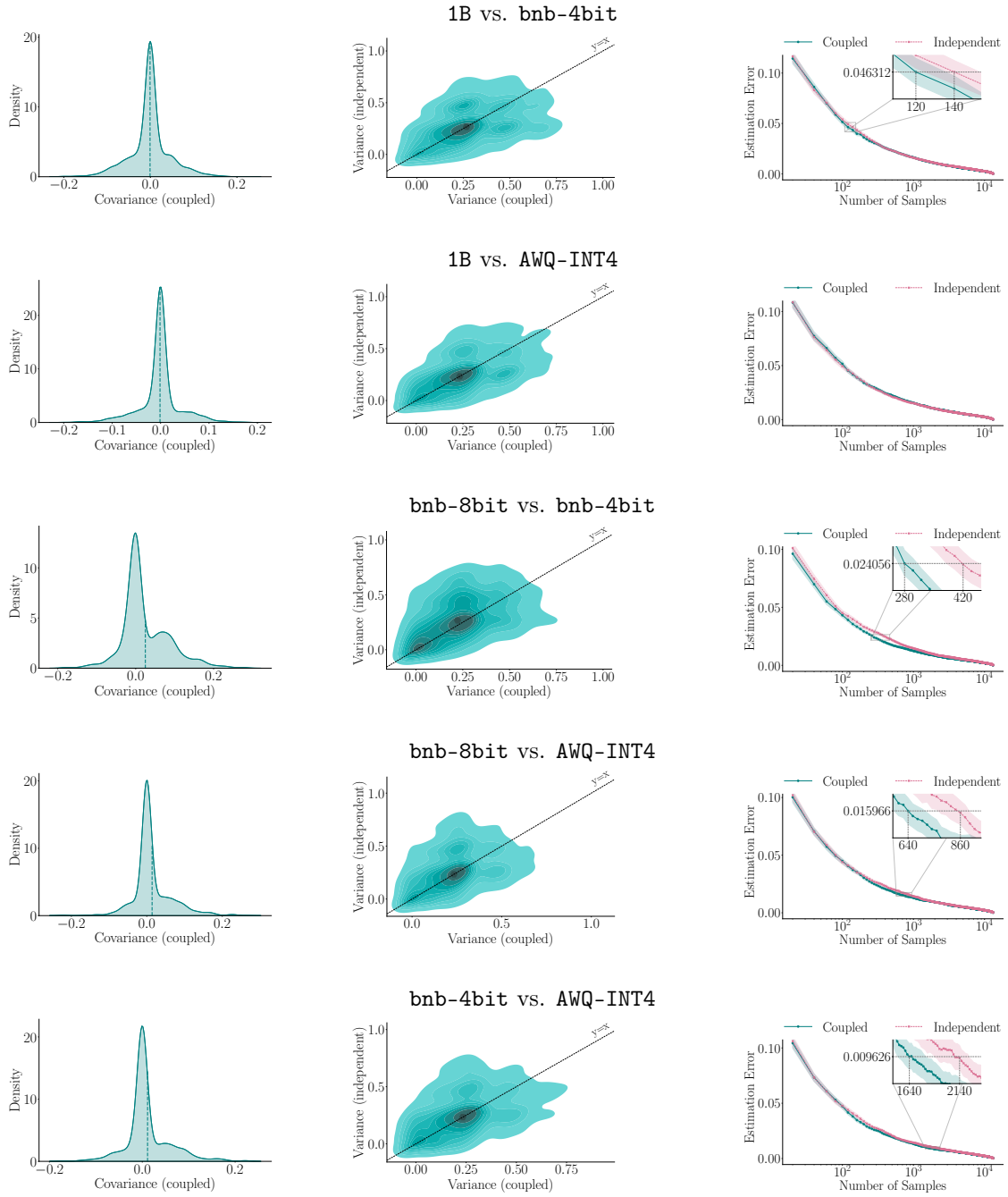
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 10: **Comparison between several pairs of LLMs in the `Llama` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



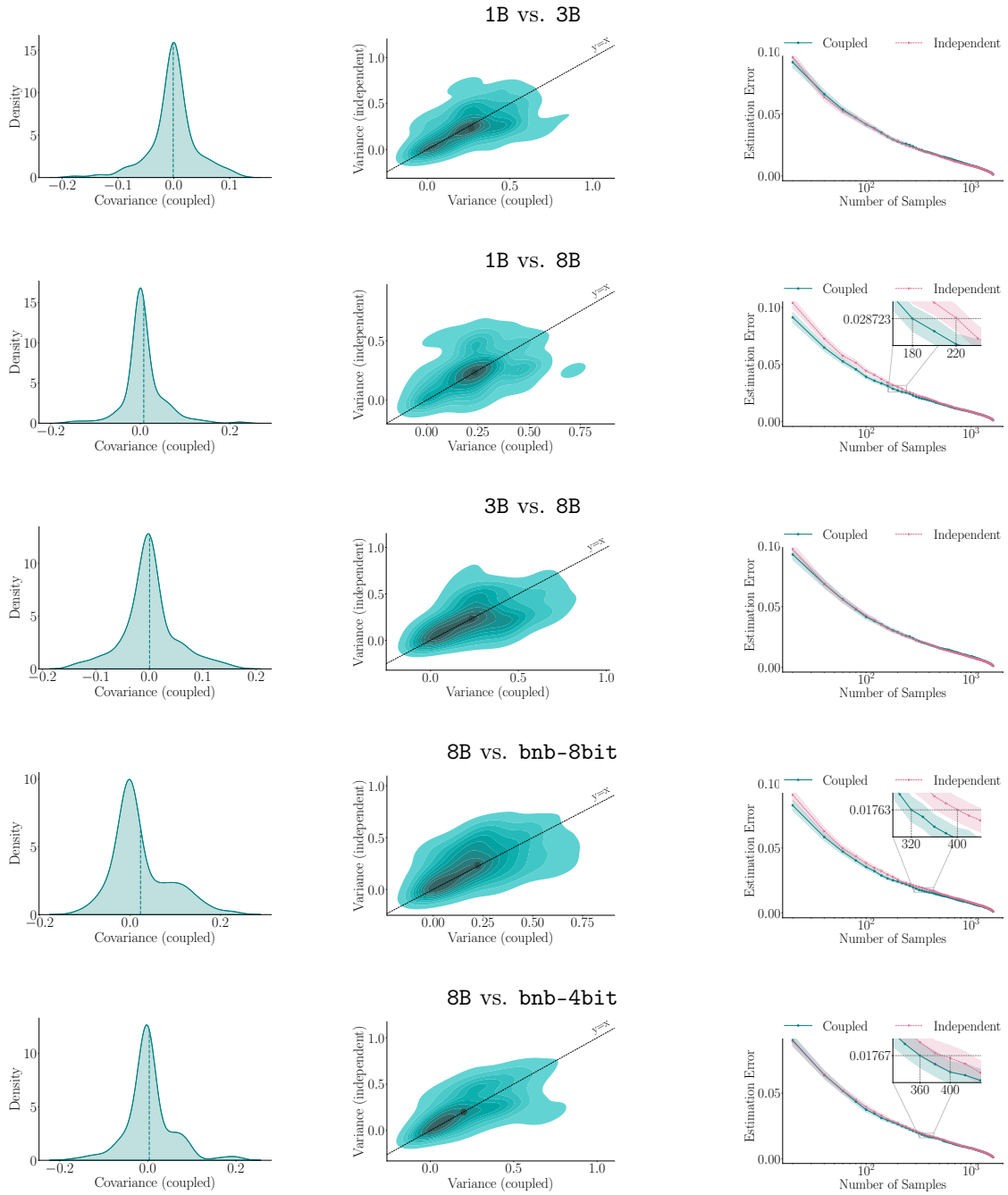
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 11: **Comparison between several pairs of LLMs in the `Llama` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



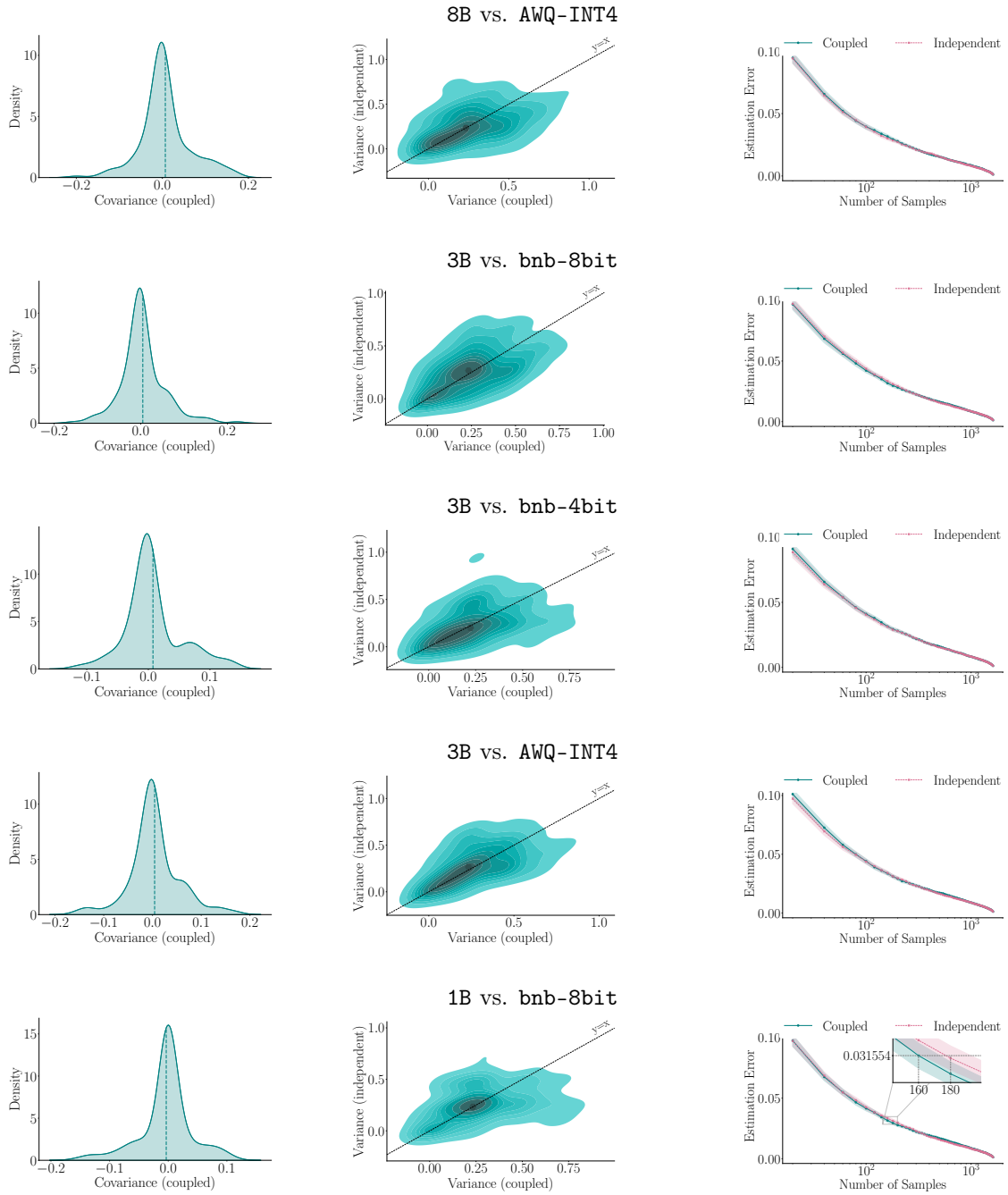
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 12: **Comparison between several pairs of LLMs in the `Llama` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



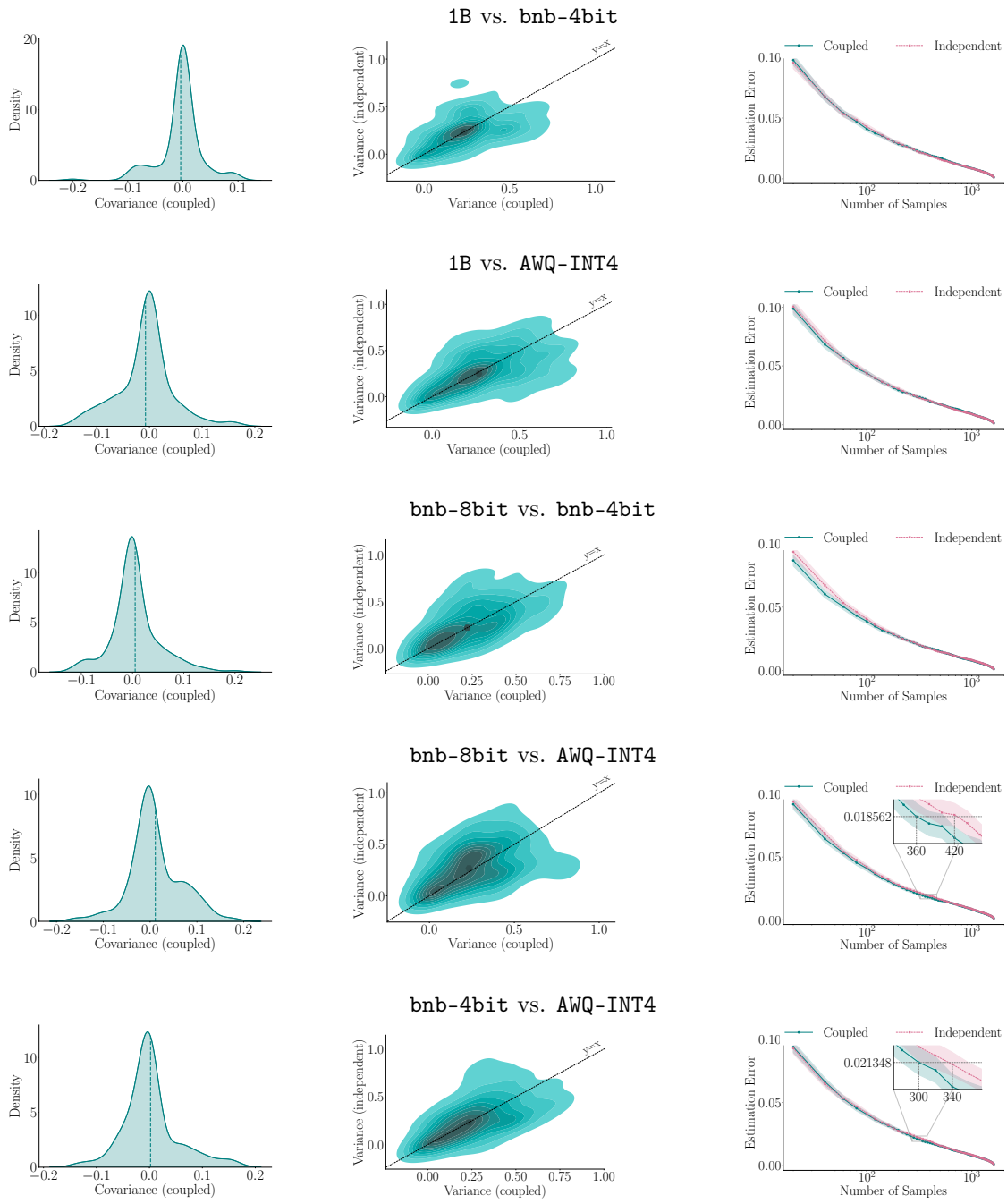
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 13: **Comparison between several pairs of LLMs in the `Llama` family on programming problems from the HumanEval dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 14: **Comparison between several pairs of LLMs in the `Llama` family on programming problems from the HumanEval dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 15: **Comparison between several pairs of LLMs in the `Llama` family on programming problems from the HumanEval dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.

F.3 Pairwise Comparisons

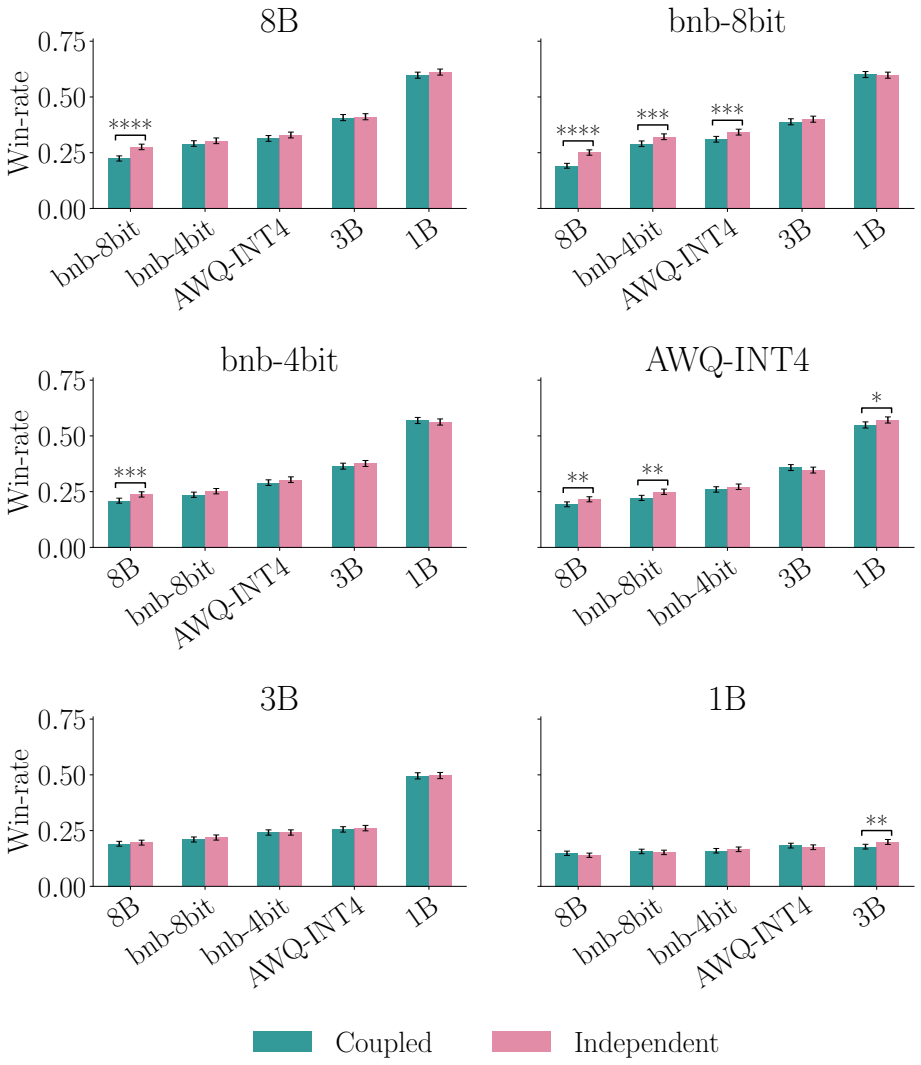


Figure 16: **Empirical win-rate of each LLM against other LLMs in the `Llama` family on questions from the LMSYS-Chat-1M dataset.** Empirical estimate of the win-rate under coupled autoregressive generation as given by Eq. 6 and under independent generation generation as given by Eq. 5. Each empirical win-rate is computed using pairwise comparisons between the outputs of each LLM and any other LLM over 500 questions with 10 (different) random seeds. The error bars correspond to 95% confidence intervals. For each pair of empirical win-rates, we conduct a two-tailed test, to test the hypothesis that the empirical win-rates are the same; (****, ***, **, *) indicate p-values (< 0.0001, < 0.001, < 0.01, < 0.05), respectively.

G EXPERIMENTS WITH LLMS IN THE `Qwen` FAMILY

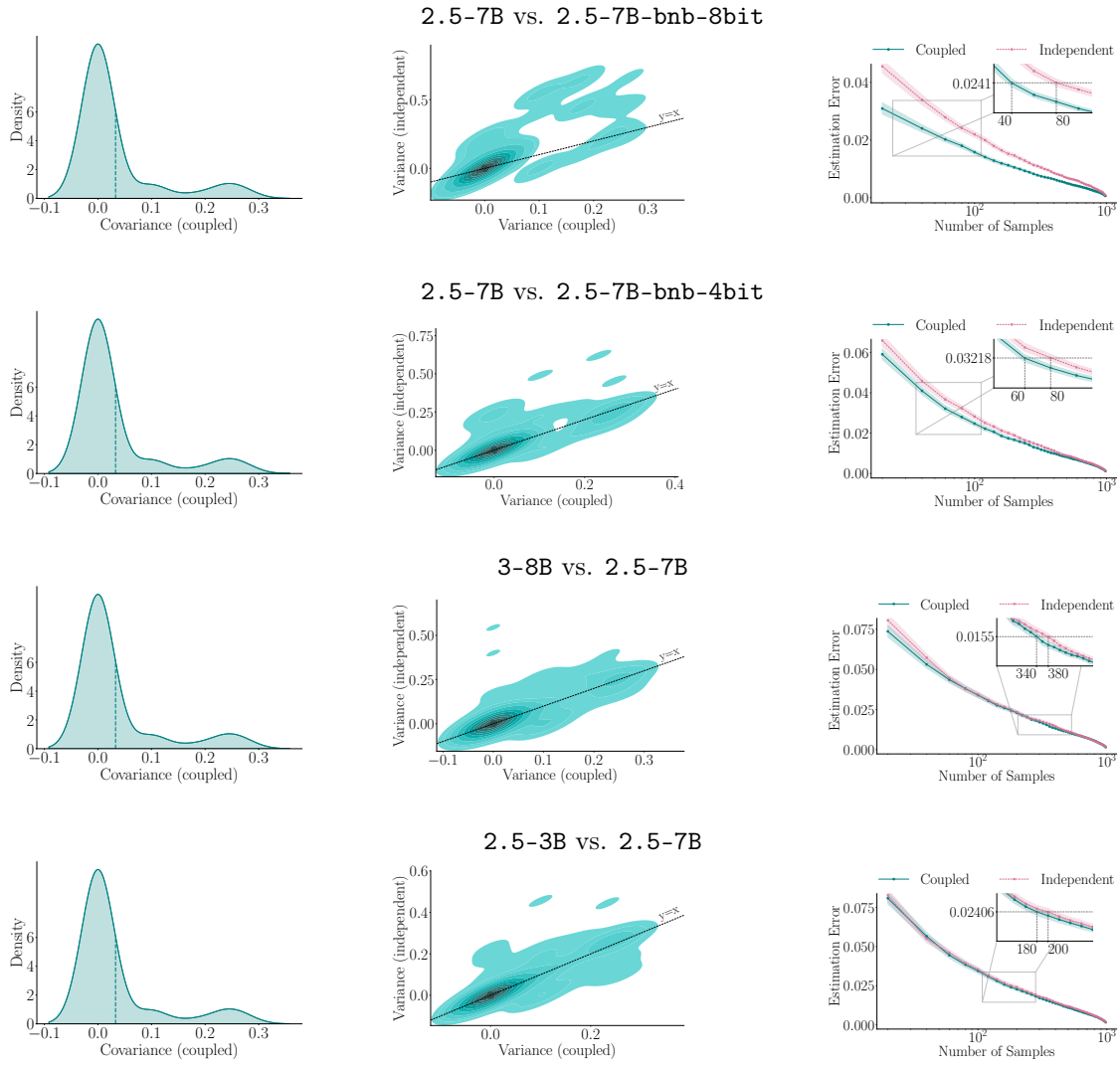
In this section, we experiment with LLMs from the `Qwen` family. For brevity, we shorten the names of the LLMs, as listed in Table 6.

Full name	Shortened name
`Qwen2.5-7B-Instruct`	`2.5-7B`
`Qwen2.5-7B-Instruct-AWQ-INT4`	`2.5-7B-AWQ-INT4`
`Qwen2.5-7B-Instruct-bnb-4bit`	`2.5-7B-bnb-4bit`
`Qwen2.5-7B-Instruct-bnb-8bit`	`2.5-7B-bnb-8bit`
`Qwen2.5-3B-Instruct`	`2.5-3B`
`DistilQwen2.5-3B-Instruct`	`2.5-3B-distil`
`Qwen2.5-1.5B-Instruct`	`2.5-1.5B`
`Qwen3-8B`	`3-8B`

Table 6: Full and shortened names of LLMs in the `Qwen` family.

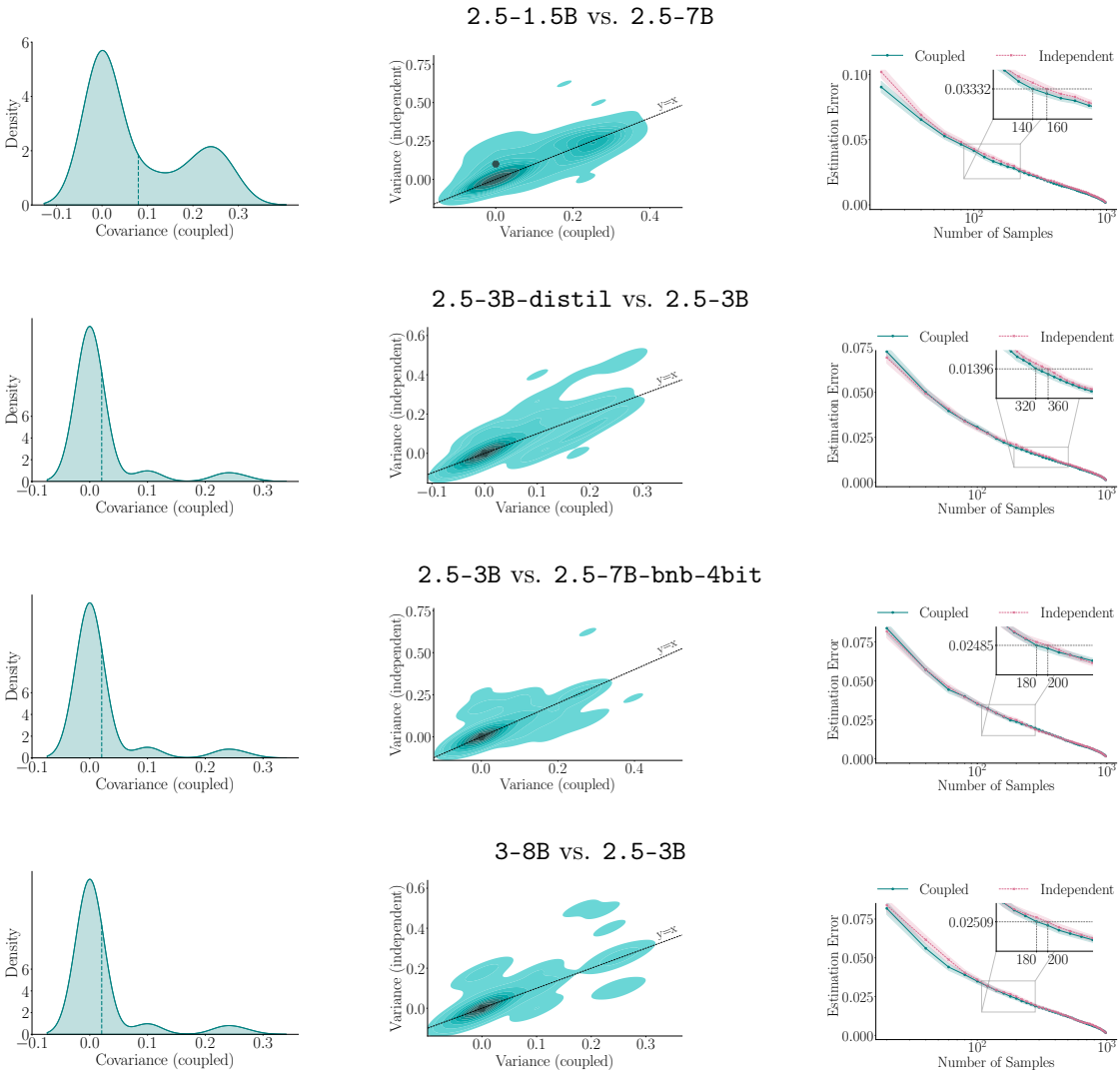
G.1 MMLU Dataset

Here, we experiment with models from the `Qwen` family on the MMLU dataset following the setup described in Section 5.1. We find that, for $\sim$40% of the pairs of models shown in Table 6, coupled generation leads to at least a 10% reduction in the number of samples required to achieve equivalent error in the estimation of the expected difference between the scores of the LLMs on the knowledge area `college computer science`. For brevity, we only show the results for the above mentioned pairs in Figures 17–19. Further, Figure 20 shows the results for `2.5-7B` against `2.5-7B-bnb-8bit` on different knowledge areas. Overall, the results are qualitatively similar to those in Section 5.1.



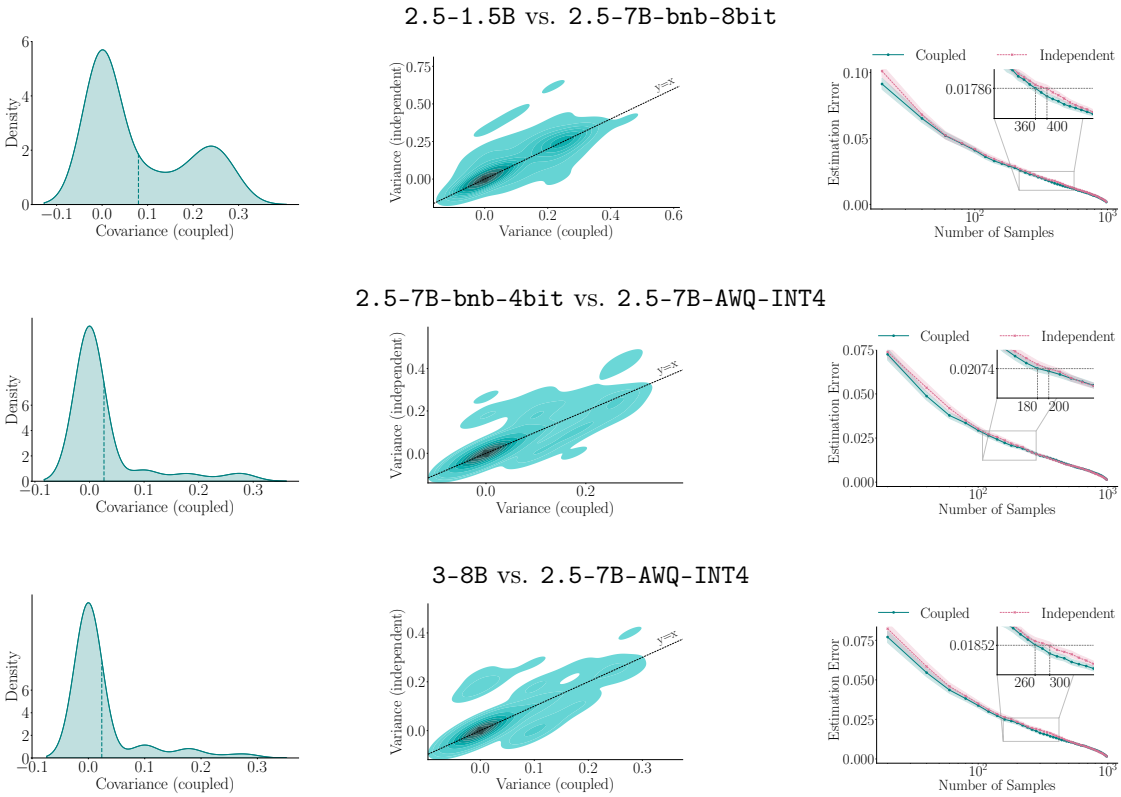
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 17: **Comparison between four pairs of LLMs in the `Qwen` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



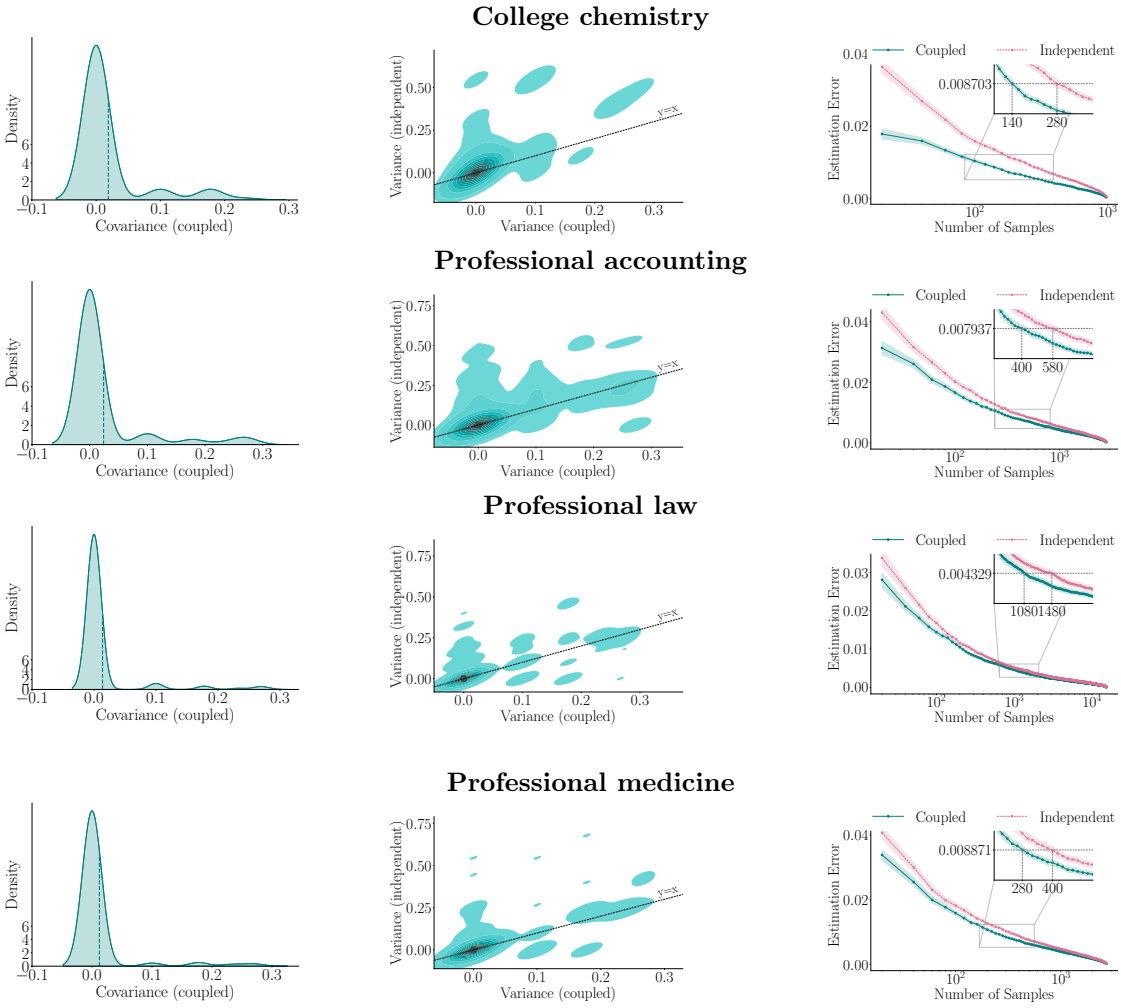
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 18: **Comparison between four pairs of LLMs in the `Qwen` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 19: **Comparison between three pairs of LLMs in the `Qwen` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.

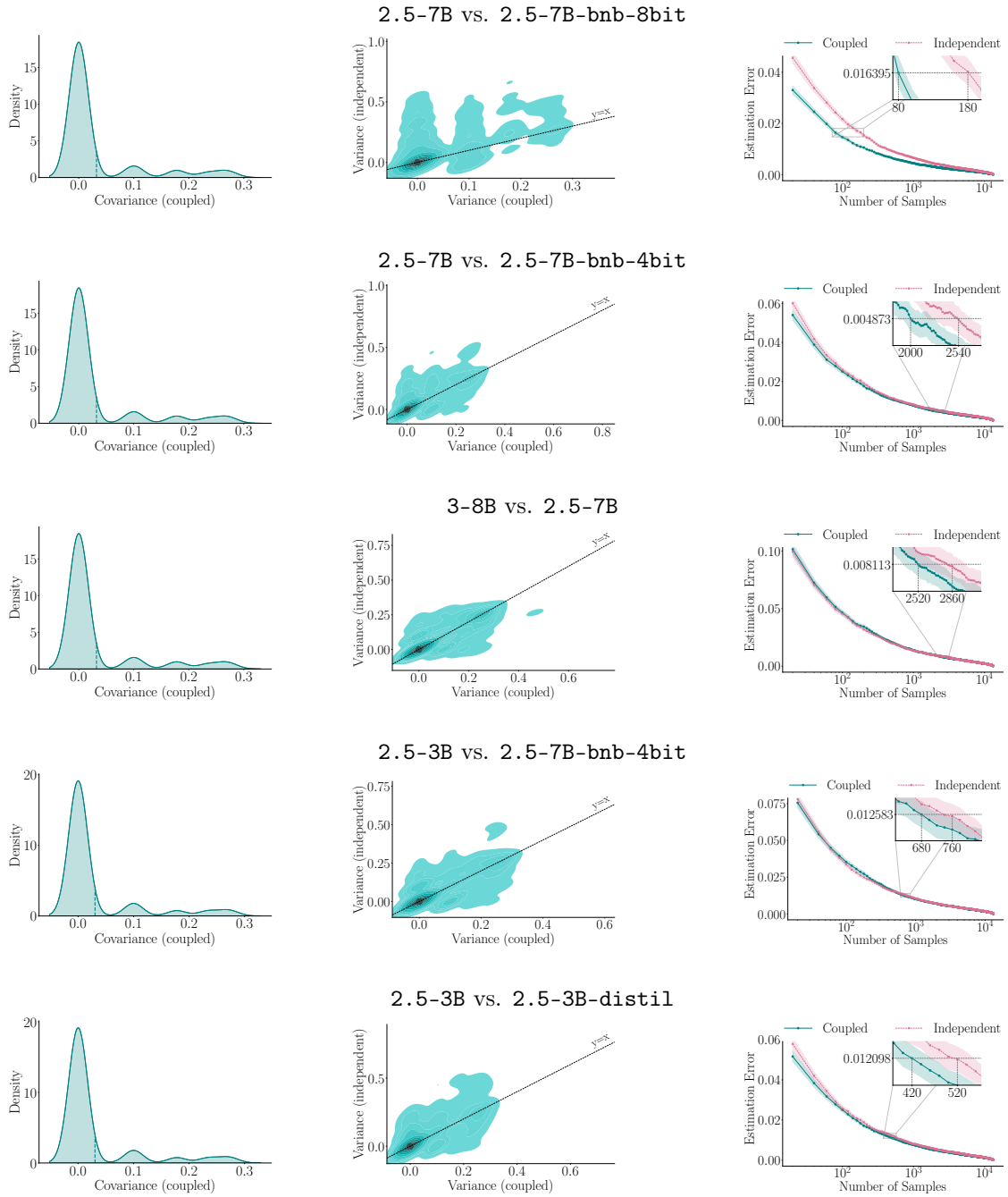


(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 20: **Comparison between `2.5-7B` and `2.5-7B-bnb-8bit` from the `Qwen` family on multiple-choice questions from four knowledge areas of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals. We observe qualitatively similar results for other knowledge areas.

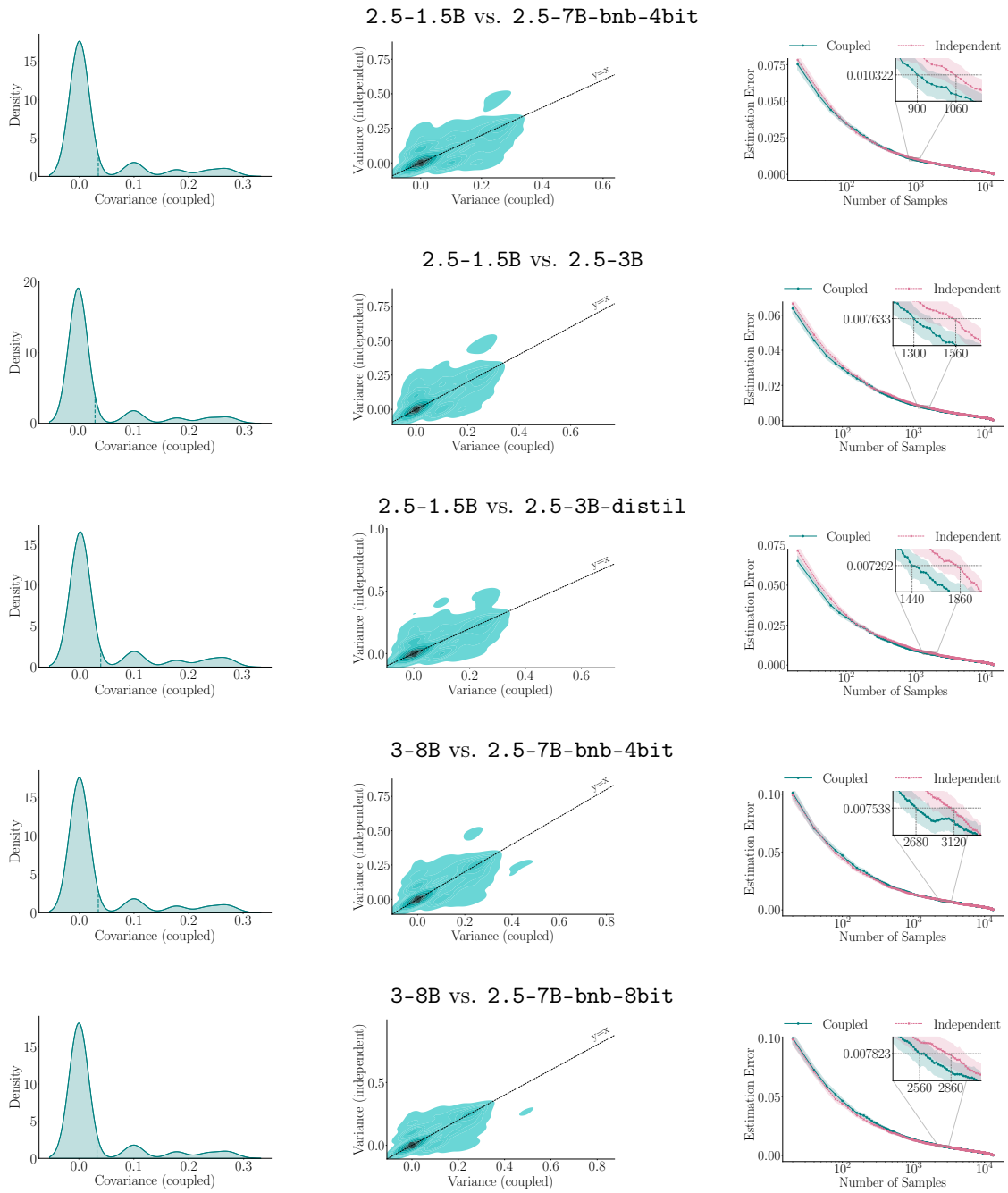
G.2 GSM8K and HumanEval Datasets

Here, we experiment with models in the `Qwen` family on the GSM8K and HumanEval datasets following the setup described in Appendix F.2. We find that, for ∼61% and ∼32% of the pairs of models shown in Table 6, coupled generation leads to at least a 10% reduction in the number of samples required to achieve equivalent error in the estimation of the expected difference between the scores of the LLMs on the GSM8K dataset and the HumanEval dataset, respectively. For brevity, we only show the results for the above mentioned pairs in Figures 21–26. Overall, the results are qualitatively similar to those in Appendix F.2.



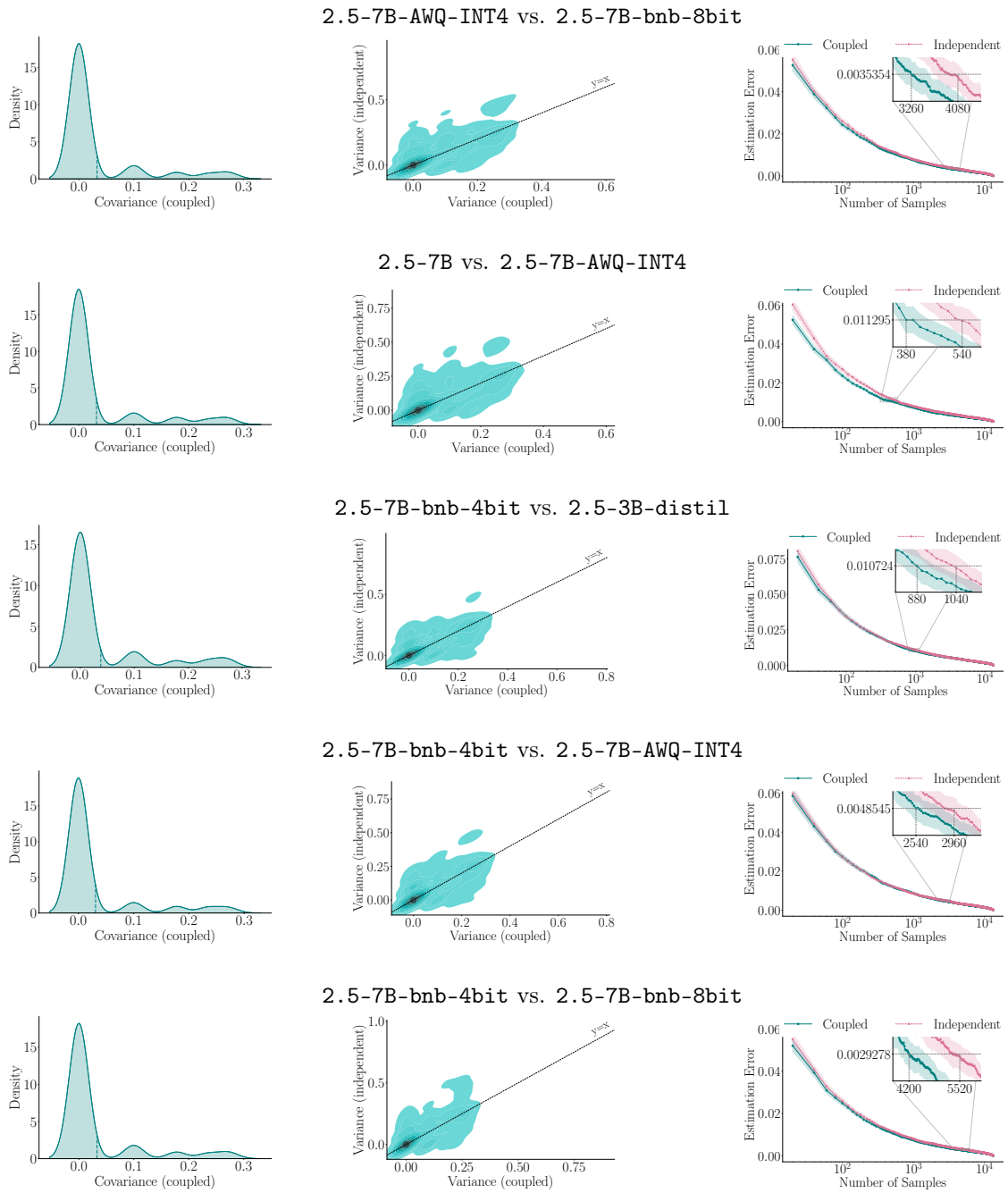
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 21: **Comparison between several pairs of LLMs in the `Qwen` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



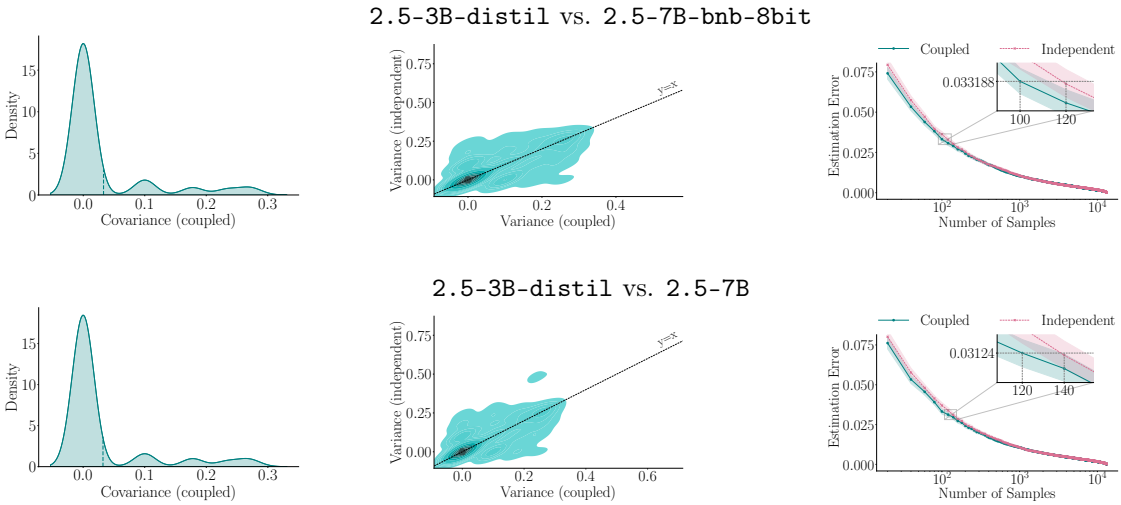
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 22: **Comparison between several pairs of LLMs in the `Qwen` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



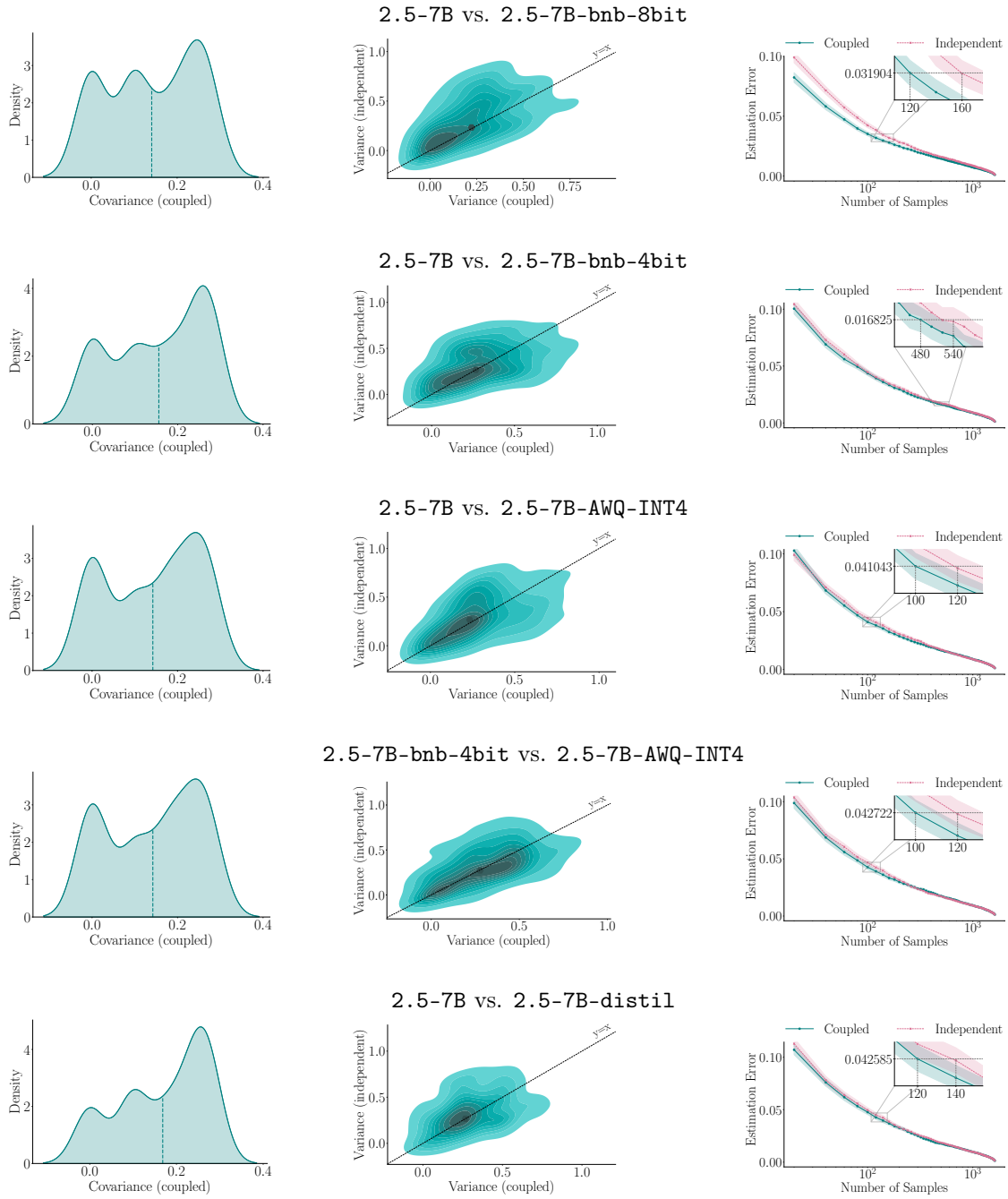
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 23: **Comparison between several pairs of LLMs in the `Qwen` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



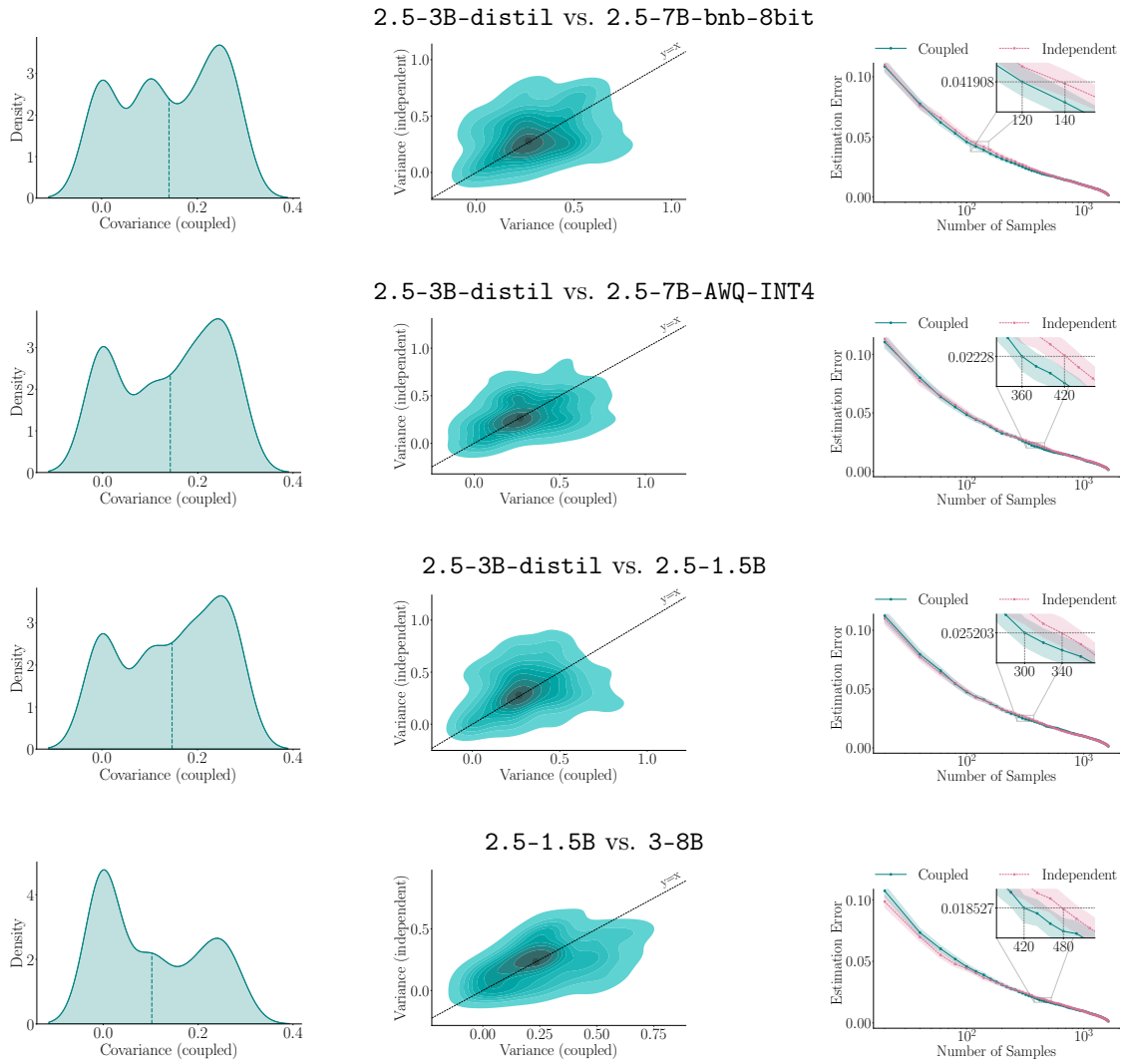
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 24: **Comparison between several pairs of LLMs in the `Qwen` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 25: **Comparison between several pairs of LLMs in the `Qwen` family on programming problems from the HumanEval dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 26: **Comparison between several pairs of LLMs in the `Qwen` family on programming problems from the HumanEval dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.

G.3 Pairwise Comparisons

Here, we experiment with models from the `Qwen` family using pairwise comparisons between their outputs by a strong LLM, when prompted with open-ended questions from the LMSYS Chatbot Arena platform, following the setup described in Section 4. Table 7 and Figure 27 summarizes the results, which are qualitatively similar to those in Section 4.

	Coupled		Independent	
LLM	Rank	Avg. win-rate	Rank	Avg. win-rate
`3-8B`	1	0.4486 ± 0.0013	1	0.4524 ± 0.0013
`2.5-7B-bnb-8bit`	2	0.3211 ± 0.0012	2	0.3264 ± 0.0012
`2.5-7B`	3	0.3148 ± 0.0012	2	0.3279 ± 0.0012
`2.5-7B-bnb-4bit`	4	0.2892 ± 0.0012	4	0.2862 ± 0.0012
`2.5-7B-AWQ-INT4`	5	0.2561 ± 0.0011	5	0.2591 ± 0.0012
`2.5-3B`	6	0.2072 ± 0.0011	6	0.2121 ± 0.0011
`2.5-1.5B`	7	0.1834 ± 0.0010	7	0.1875 ± 0.0010
`2.5-3B-distil`	8	0.1780 ± 0.0010	8	0.1850 ± 0.0010

Table 7: Average win-rate of each LLM across all other LLMs in the `Qwen` family (± 95% confidence intervals). To derive the rankings, for each LLM, we choose the lowest ranking provided by the method of Chatzi et al. (2024).

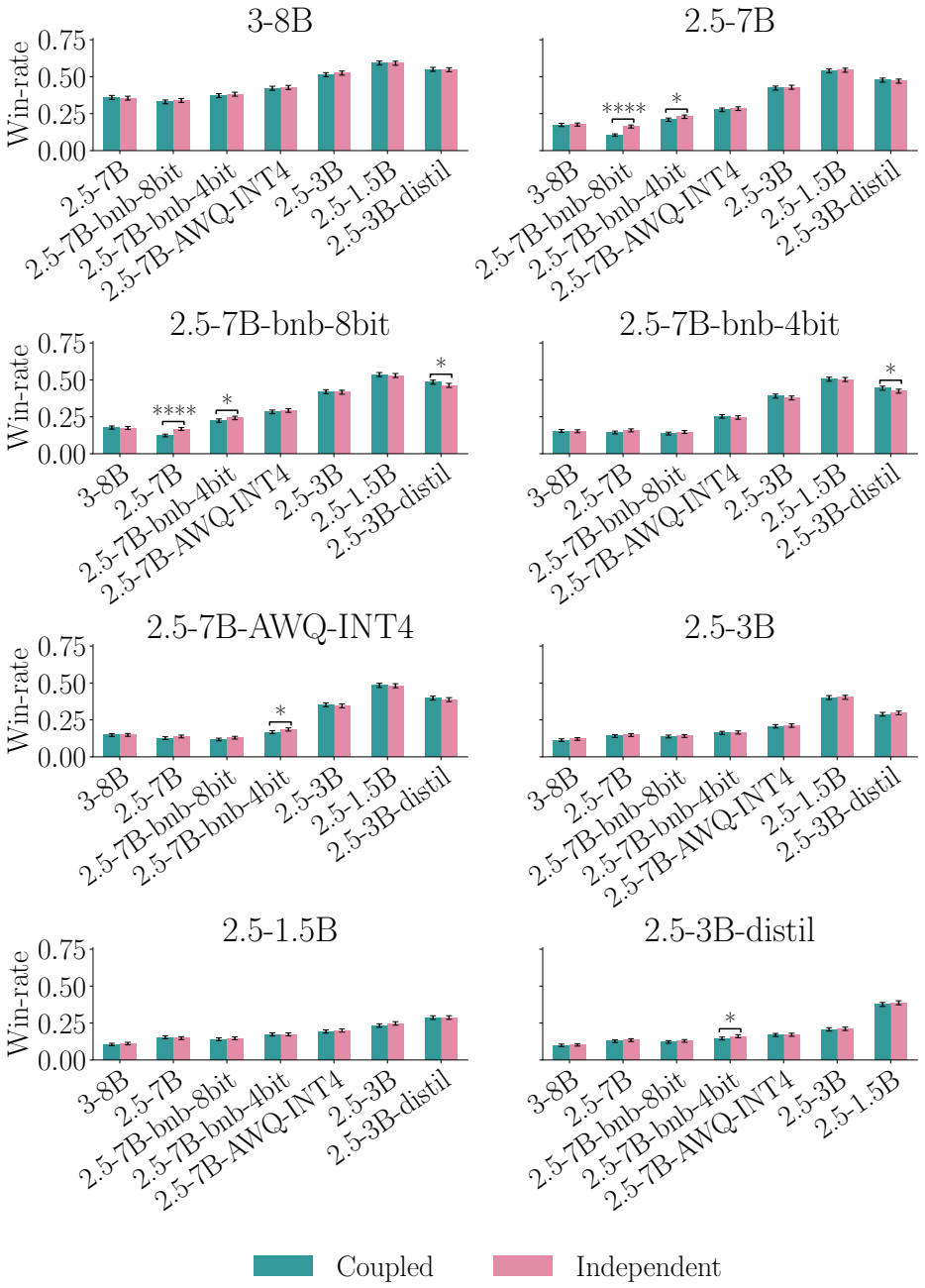


Figure 27: **Empirical win-rate of each LLM against other LLMs in the `Qwen` family on questions from the LMSYS-Chat-1M dataset.** Empirical estimate of the win-rate under coupled autoregressive generation as given by Eq. 6 and under independent generation generation as given by Eq. 5. Each empirical win-rate is computed using pairwise comparisons between the outputs of each LLM and any other LLM over 500 questions with 10 (different) random seeds. The error bars correspond to 95% confidence intervals. For each pair of empirical win-rates, we conduct a two-tailed test, to test the hypothesis that the empirical win-rates are the same; (****, ***, **, *) indicate p-values (< 0.0001, < 0.001, < 0.01, < 0.05), respectively.

H EXPERIMENTS WITH LLMS IN THE `Mistral` FAMILY

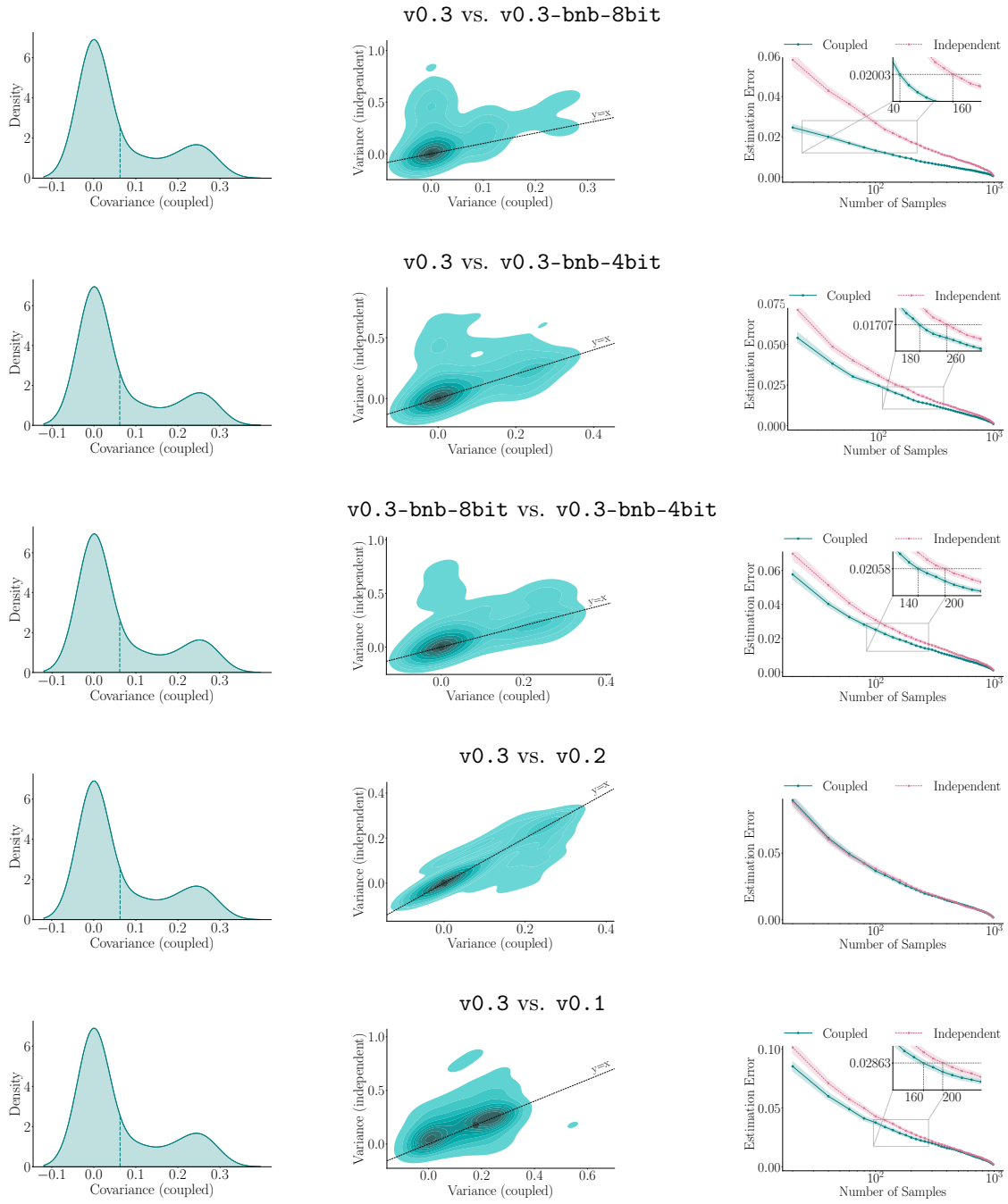
In this section, we experiment with LLMs in from the `Mistral` family. For brevity, we shorten the names of the LLMs, as listed in Table 8.

Full name	Shortened name
`Mistral-7B-Instruct-v0.3`	`v0.3`
`Mistral-7B-Instruct-v0.3-bnb-4bit`	`v0.3-bnb-4bit`
`Mistral-7B-Instruct-v0.3-bnb-8bit`	`v0.3-bnb-8bit`
`Mistral-7B-Instruct-v0.2`	`v0.2`
`Mistral-7B-Instruct-v0.1`	`v0.1`

Table 8: Full and shortened names of LLMs in the `Mistral` family.

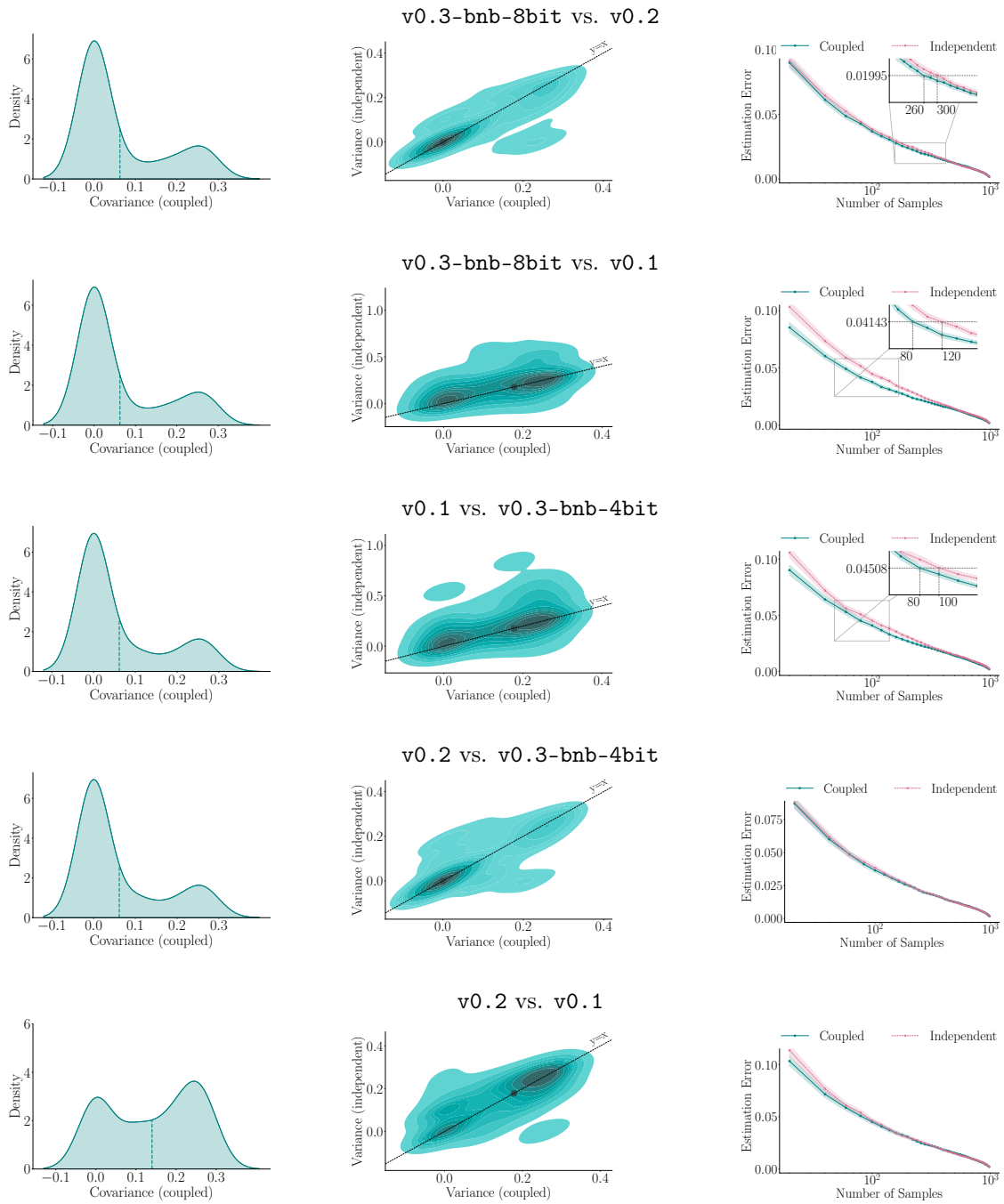
H.1 MMLU Dataset

Here, we experiment with models from the `Mistral` family on the MMLU dataset following the setup described in Section 5.1. Figures 28 and 29 show the results for all pairs of models described in Table 8 on the knowledge area `college computer science`, and Figure 30 shows the results for `v0.3` against `v0.3-bnb-8bit` on different knowledge areas. Overall, the results are qualitatively similar to those in Section 5.1.



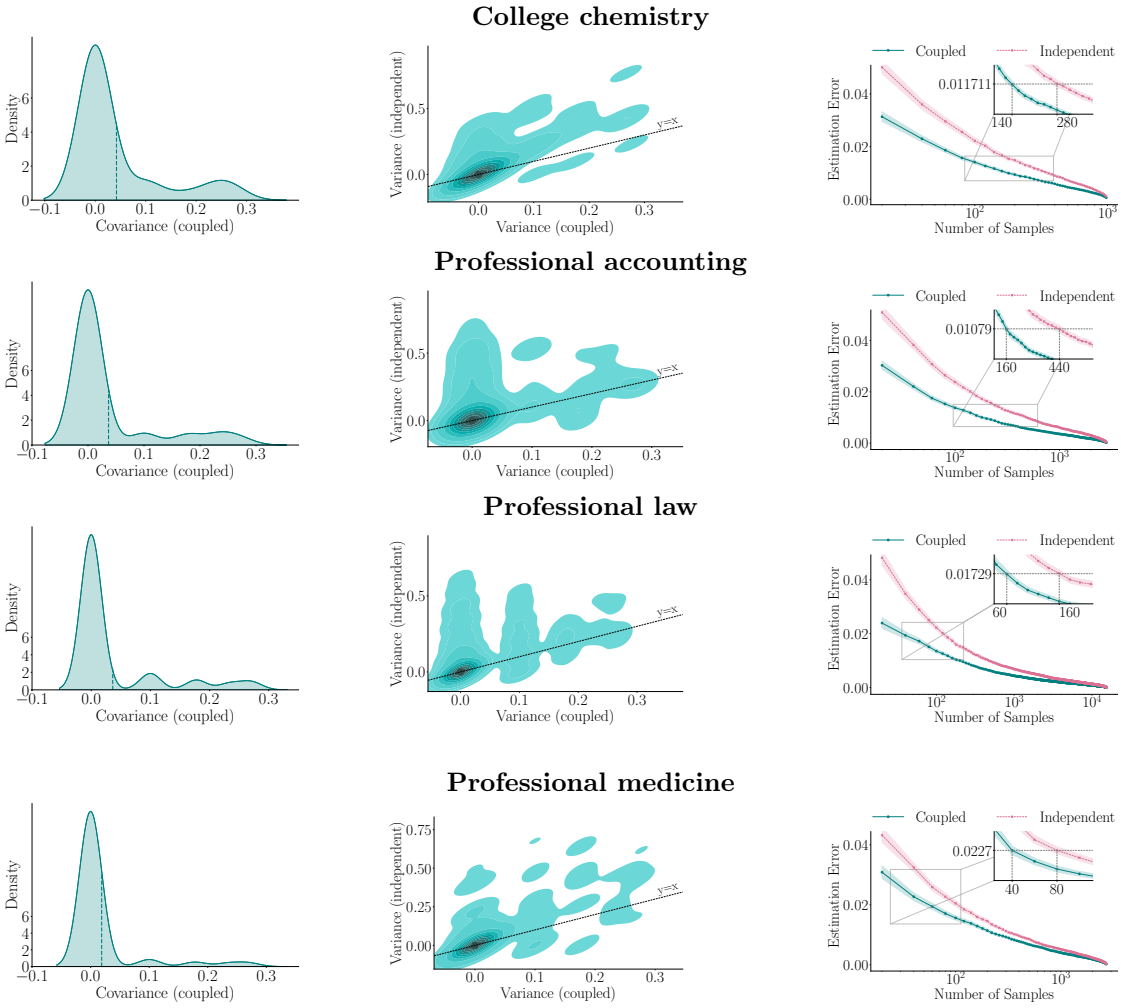
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 28: **Comparison between five pairs of LLMs in the `Mistral` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 29: **Comparison between five pairs of LLMs in the `Mistral` family on multiple-choice questions from the "college computer science" knowledge area of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.

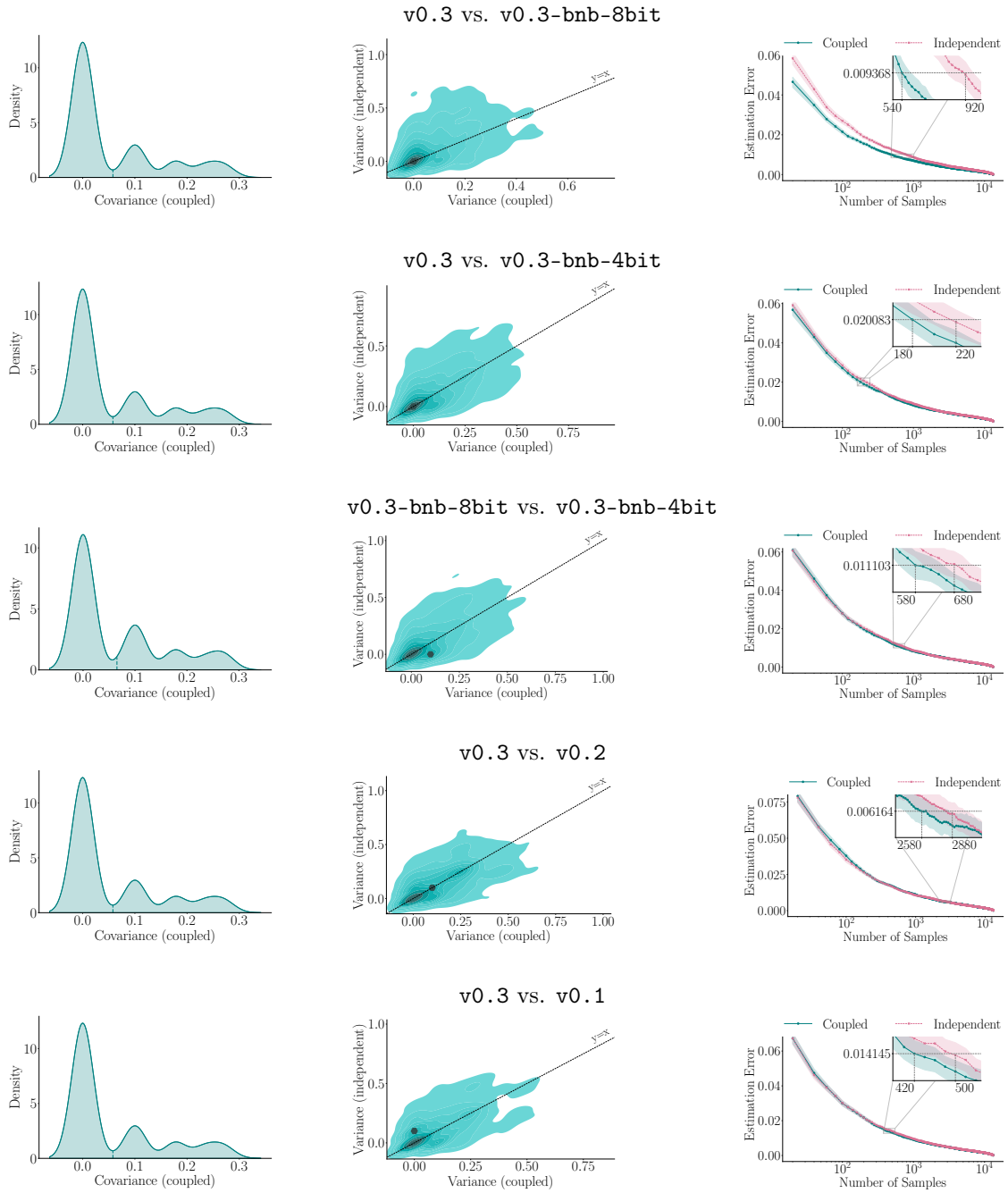


(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 30: **Comparison between `v0.3` and `v0.3-bnb-8bit` from the `Mistral` family on multiple-choice questions from four knowledge areas of the MMLU dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each question under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals. We observe qualitatively similar results for other knowledge areas.

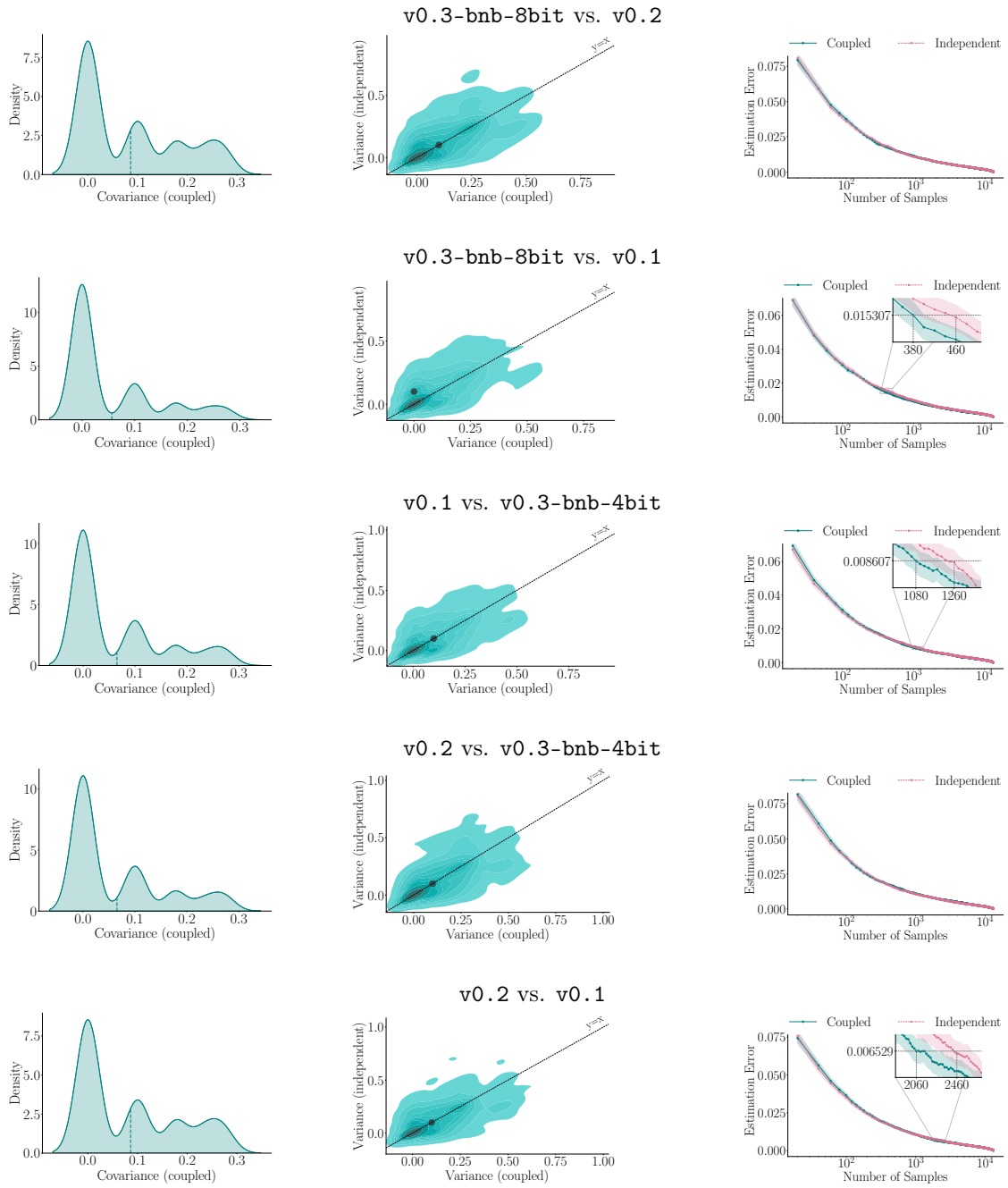
H.2 GSM8K and HumanEval Datasets

Here, we experiment with models from the `Mistral` family on the GSM8K and HumanEval datasets following the setup described in Appendix F.2. Figures 31–34 show the results for all pairs of models in Table 8, which are qualitatively similar to those in Appendices F.2 and G.2.



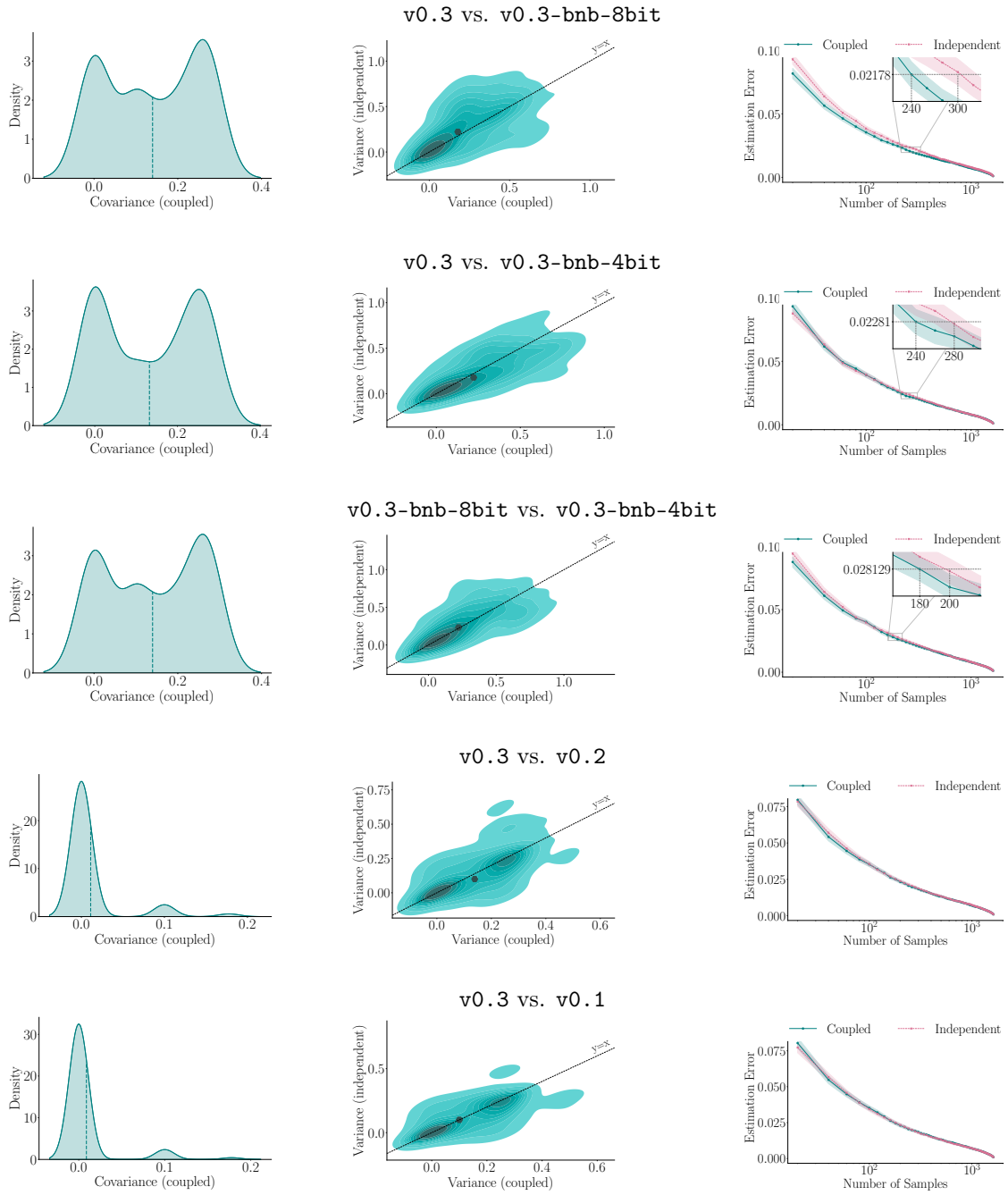
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 31: **Comparison between several pairs of LLMs in the `Mistral` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



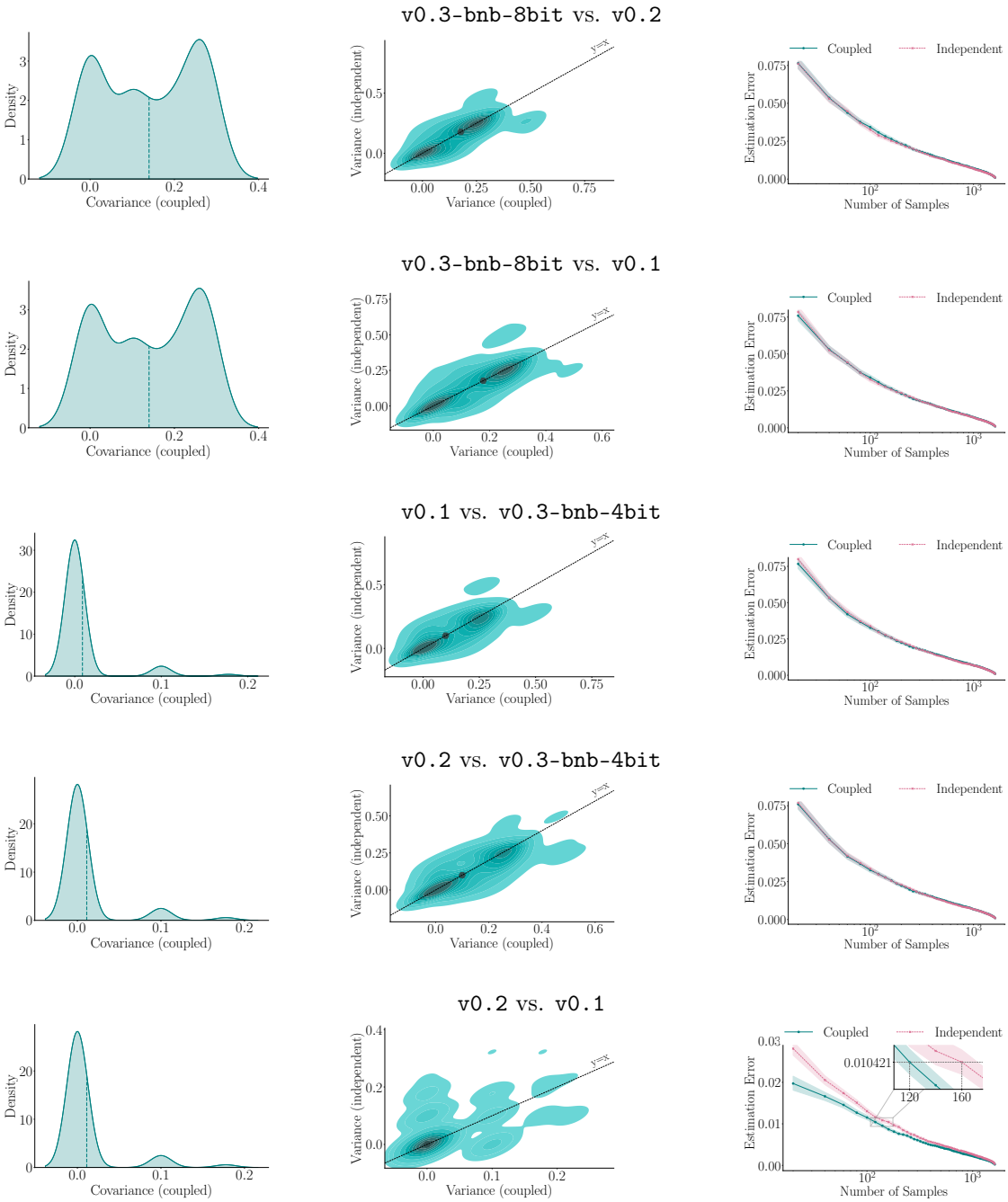
(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 32: **Comparison between several pairs of LLMs in the `Mistral` family on math questions from the GSM8K dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 33: **Comparison between several pairs of LLMs in the `Mistral` family on programming problems from the HumanEval dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.



(a) Score covariance (b) Variance of the score difference (c) Estimation error vs. # samples

Figure 34: **Comparison between several pairs of LLMs in the `Mistral` family on programming problems from the HumanEval dataset.** Panels in column (a) show the kernel density estimate (KDE) of the covariance between the scores of the two LLMs on each problem under coupled generation; the dashed lines correspond to average values. Panels in column (b) show the KDE of the variance of the difference between the scores of the LLMs on each question under coupled and independent generation; the highlighted points correspond to median values. Panels in column (c) show the absolute error in the estimation of the expected difference between the scores of the LLMs against the number of samples; for each point on the x-axis, we perform 1,000 sub-samplings and shaded areas correspond to 95% confidence intervals.

H.3 Pairwise Comparisons

Here, we experiment with models from the `Mistral` family using pairwise comparisons between their outputs by a strong LLM, when prompted with open-ended questions from the LMSYS Chatbot Arena platform, following the setup described in Section 4. Table 9 and Figure 35 summarizes the results, which are qualitatively similar to those in Section 4.

	Coupled		Independent	
LLM	Rank	Avg. win-rate	Rank	Avg. win-rate
`v0.2`	1	0.3310 ± 0.0026	1	0.3355 ± 0.0026
`v0.3-bnb-4bit`	2	0.2861 ± 0.0025	2	0.2953 ± 0.0025
`v0.3-bnb-8bit`	2	0.2844 ± 0.0025	2	0.2953 ± 0.0025
`v0.3`	4	0.2757 ± 0.0025	2	0.2918 ± 0.0025
`v0.1`	5	0.1598 ± 0.0020	5	0.1617 ± 0.0020

Table 9: Average win-rate of each LLM across all other LLMs in the `Mistral` family (± 95% confidence intervals). To derive the rankings, for each LLM, we choose the lowest ranking provided by the method of Chatzi et al. (2024).

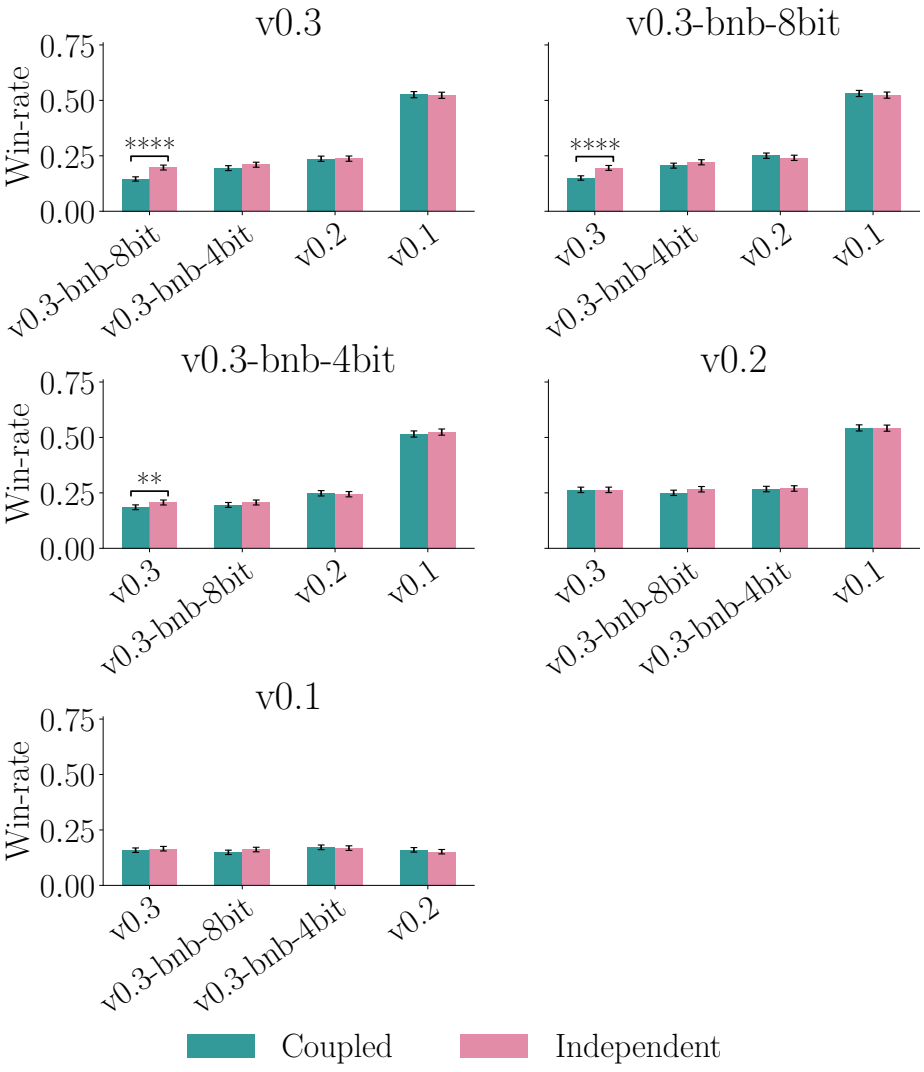


Figure 35: **Empirical win-rate of each LLM against other LLMs in the `Mistral` family on questions from the LMSYS-Chat-1M dataset.** Empirical estimate of the win-rate under coupled autoregressive generation as given by Eq. 6 and under independent generation generation as given by Eq. 5. Each empirical win-rate is computed using pairwise comparisons between the outputs of each LLM and any other LLM over 500 questions with 10 (different) random seeds. The error bars correspond to 95% confidence intervals. For each pair of empirical win-rates, we conduct a two-tailed test, to test the hypothesis that the empirical win-rates are the same; (****, ***, **, *) indicate p-values (< 0.0001, < 0.001, < 0.01, < 0.05), respectively.