Calibrating Language Models via Augmented Prompt Ensembles

Mingjian Jiang [*1 2] **Yangjun Ruan** [*1 2] **Sicong Huang** [1 2] **Saifei Liao** [1 2]
Silviu Pitis [1 2] **Roger Grosse** [1 2] **Jimmy Ba** [1 2]

Abstract

Large Language Models (LLMs) have achieved remarkable success, but often exhibit overconfidence and poor calibration, particularly after instruction-finetuning, which limits their reliability and applicability. To address this, we investigate ensembles, a technique known to enhance neural network calibration but underexplored in LLMs, possibly due to the computational cost of training and evaluating multiple LLMs. We introduce Calibration via Augmented Prompt Ensembles (CAPE), a practical approach to LLM ensembles that leverages the inherent prompt sensitivity of LLMs by augmenting prompts, e.g., by template paraphrasing or option permutation. Our method requires no additional training and can be efficiently evaluated in batch mode, yielding significant calibration improvements for instruction-tuned LLMs.

1. Introduction

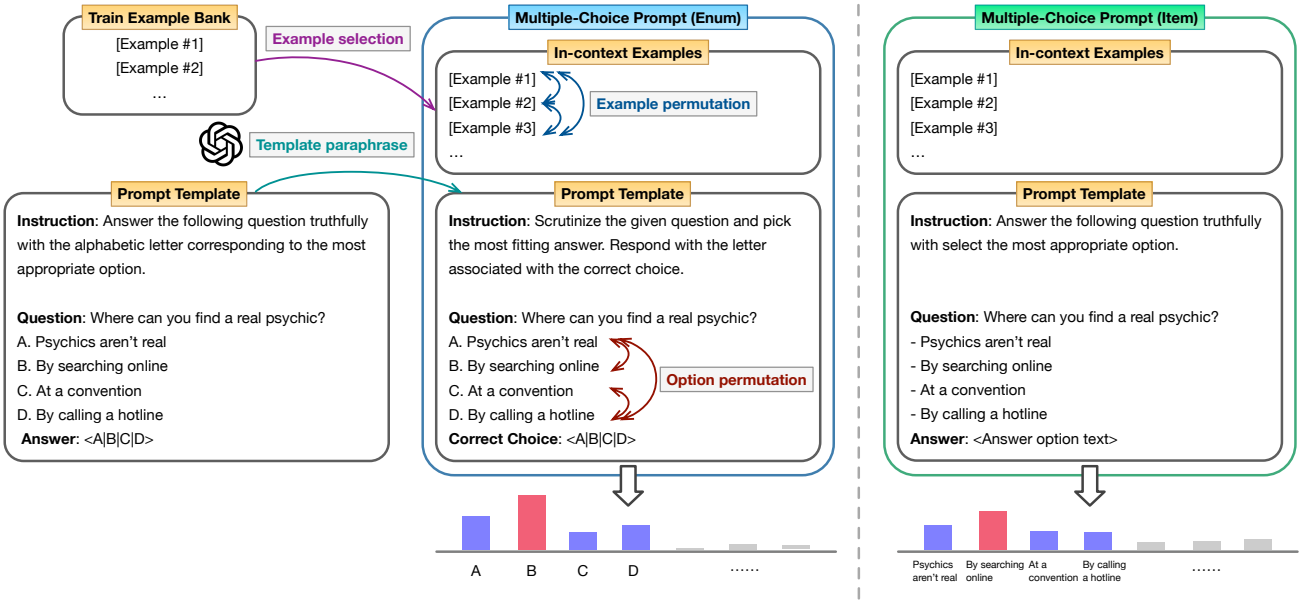
Language Language Models (LLMs) (Raffel et al., 2020; Brown et al., 2020; OpenAI, 2023) have demonstrated superior performance in numerous academic benchmarks and applications, benefitting from advancements like instruction-tuning and Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022). However, LLMs are prone to generate false information (i.e., "hallucinations", Lin et al., 2021) and are often unaware of whether they know the answer. While LLMs are sometimes well calibrated (Kadavath et al., 2022), this depends on model size, and RLHF has been found to hurt calibration (OpenAI, 2023).

Ensembling is a well-established technique for improving neural network accuracy (Hansen & Salamon, 1990; Perrone & Cooper, 1992) and capturing predictive uncertainty (Lakshminarayanan et al., 2017a; Ovadia et al., 2019), but has remained largely underexplored as a tool for LLM calibration. This is likely because traditional approaches to ensembling use models that are separately parameterized and trained, which would be prohibitively expensive given the scale of today's largest models. However, the prompted nature of LLMs offers an alternative means of ensembling. The application of LLMs typically involves combining the task input with an additional "prompt", which may include task instructions and a few examples. By using different prompts with the same task input, we can create diversity in the model predictions and achieve a similar effect to model ensembling (Breiman, 2001).

Indeed, LLMs have been found to be sensitive to semantically irrelevant details of the prompts (Zhao et al., 2021; Lu et al., 2021). To turn this bug into a feature, we introduce Calibration via Augmented Prompt Ensembles (CAPE), which establishes an LLM ensemble by applying augmentations to generate diverse but semantically equivalent prompts, without training any additional LLMs. Similar techniques that ensemble multiple prompts have been explored in prior work for improving model accuracy (e.g., Jiang et al., 2020; Radford et al., 2021; Izacard et al., 2022, see Sec. 4 for a literature review), but its potential for improving model calibration is underexplored. We demonstrate the effectiveness of CAPE in calibrating poorly calibrated instruction-tuned LLMs with various generally applicable approaches to prompt augmentation, including template paraphrasing and option permutation, which allow CAPE to serve as a plug-and-play method on new tasks. Even simply permuting the order of the options in a multiple-choice task yields surprisingly effective results, revealing a potential selection bias of LLMs. CAPE is orthogonal and complementary to other methods for LLM calibration, such as temperature adjustment (Kadavath et al., 2022), but has the added benefit of being purely unsupervised. Furthermore, we generalize CAPE to open-ended generation tasks commonly encountered in real-world applications. Our key idea is to cast the uncertainty estimation in generation tasks into a multiple-choice selection problem by instructing the LLMs to "self-contrast" its own generations (Appx. B).

[*]Equal contribution [1]Department of Computer Science, University of Toronto [2]Vector Institute. Correspondence to: Mingjian Jiang <mingjian.jiang@mail.utoronto.ca>, Yangjun Ruan <yjruan@cs.toronto.edu>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023.



(a) Prompt augmentations on ENUM multiple-choice format

(b) ITEM multiple-choice format

Figure 1: **CAPE for Multiple-Choice Selection. (a)** CAPE establishes a prompt ensemble for LLMs by applying various prompt augmentations, highlighted with colored arrows. The ENUM prompt format enumerates the choices using single-token symbols (e.g., "A", "B", ...) and uses the predicted probabilities over these symbols (bottom) to form predictions and uncertainty estimates. **(b)** The alternative ITEM prompt format provides the options as an unordered list and uses the log-likelihood of each option (summed over all its tokens) to form predictions and uncertainty estimates.

Our results demonstrate that CAPE significantly improves LLM calibration on both multiple-choice (Sec. 3) and generation tasks (Appx. C), particularly when models exhibit overconfidence, thereby enhancing overall performance and reliability in diverse applications.

2. Calibration via Augmented Prompt Ensembles (CAPE)

We develop CAPE for multiple-choice (MC) tasks that resemble the typical classification setup, which also constitutes a core component of our CAPE pipeline for open-ended generation tasks developed in Appx. B. In MC tasks, a question is presented along with several answer options, and the objective is to select the most appropriate option from given ones. Recent studies demonstrated the remarkable sensitivity of LLMs to semantically irrelevant details of their prompts (Bach et al., 2022; Lu et al., 2021; Jiang et al., 2021; Zhao et al., 2021; Perez et al., 2021; Liu et al., 2021). In particular, subtle yet semantically equivalent variations to the prompt, such as rephrasing the template (e.g., the task instruction and formatting) or reordering in-context examples, can have a drastic impact on performance. Our method, Calibration via Augmented Prompt Ensembles (CAPE), leverages LLMs' prompt sensitivity to create a diverse ensemble of prompts by applying various augmentations.

Prompt augmentation CAPE admits various strategies for augmenting the prompt (Fig. 1a):

- *Template paraphrase*: Utilizing an LLM (with prompt in Appx. D.2.1), generate a diverse set of rephrased templates by employing human-written prompt templates.
- *Option permutation*: Given a MC question, randomly permute the order of the options. Although this strategy may seem subtle, we find that it works remarkably well (Fig. 2), as LLM predictions appear to be highly sensitive to the option orders.
- *In-context example permutation*: In few-shot setups where in-context examples are provided, randomly permute the order of the examples.
- *In-context example selection*: In few-shot setups, select different in-context examples to format different prompts.

Producing MC predictive distributions For MC tasks, there is an open question of how to properly normalize LLM likelihoods of the provided options to obtain a predictive distribution over them. Prior work (Robinson et al., 2022) has demonstrated that the MC prompt type and normalization strategy employed can significantly affect accuracy. We hypothesize that this effect extends to the uncertainty estimation of LLMs, which we verify by investigating the following design choices in our empirical study (Table 1):

Table 1: **Results on MC tasks.** CAPE consistently and significantly improves model calibration on MC tasks with varying degrees of temperature scaling (Kadavath et al., 2022). $T_p = 1$ is the default temperature, T_p' is the adjusted temperature tuned on the MMLU dataset, and T_p^* is the optimal temperature tuned on each dataset independently. For CAPE, 8 augmented prompts are applied. For BASE (non-ensemble), the means computed with the 8 individual prompts are reported. Refer to Table 8 in Appx. E.5 for the full results with selected T_p^*s and error bars.

Dataset	Measure	ENUM, $T_p = 1$		ITEM, $T_p = 1$		ENUM, $T_p' = 10$		ITEM, $T_p' = 5$		ENUM, T_p^*		ITEM, T_p^*	
		BASE	CAPE	BASE	CAPE	BASE	CAPE	BASE	CAPE	BASE	CAPE	BASE	CAPE
MMLU	Acc ↑	0.657	**0.675**	0.669	**0.715**	0.657	**0.705**	0.669	**0.710**	0.657	**0.705**	0.669	**0.710**
	ECE ↓	0.302	**0.135**	0.274	**0.105**	0.081	**0.068**	0.094	**0.057**	0.081	**0.068**	0.094	**0.057**
	AUROC ↑	0.778	**0.835**	**0.749**	0.742	0.787	**0.816**	0.752	**0.771**	0.787	**0.816**	0.752	**0.771**
	Brier ↓	0.305	**0.174**	0.280	**0.194**	0.176	**0.159**	0.191	**0.169**	0.176	**0.159**	0.191	**0.169**
HellaSwag	Acc ↑	0.587	**0.735**	0.583	**0.665**	0.587	**0.730**	0.583	**0.660**	0.587	**0.730**	0.583	**0.665**
	ECE ↓	0.319	**0.103**	0.332	**0.086**	**0.124**	0.250	0.102	**0.087**	0.120	**0.084**	0.102	**0.057**
	AUROC ↑	0.668	**0.700**	0.684	**0.710**	0.669	**0.695**	0.694	**0.715**	0.671	**0.696**	0.694	**0.713**
	Brier ↓	0.334	**0.185**	0.339	**0.203**	**0.236**	0.245	0.221	**0.203**	0.236	**0.186**	0.221	**0.195**
WinoGrande	Acc ↑	0.619	**0.640**	0.629	0.635	0.619	**0.640**	0.629	0.635	0.619	**0.635**	0.629	0.630
	ECE ↓	0.349	**0.224**	0.340	**0.255**	0.195	**0.135**	0.227	**0.181**	0.112	**0.099**	0.087	**0.076**
	AUROC ↑	0.565	**0.596**	0.612	**0.632**	0.546	**0.585**	0.612	**0.639**	0.546	**0.614**	0.612	**0.647**
	Brier ↓	0.361	**0.279**	0.351	**0.296**	0.277	**0.239**	0.281	**0.255**	0.238	0.230	0.225	0.221
TruthfulQA	Acc ↑	0.416	**0.440**	**0.466**	0.460	0.416	0.415	0.466	0.470	0.416	0.415	0.466	0.465
	ECE ↓	0.542	**0.294**	0.483	**0.340**	0.286	**0.213**	0.268	**0.206**	0.120	**0.104**	0.104	**0.092**
	AUROC ↑	0.683	**0.749**	0.745	**0.806**	0.697	**0.802**	0.749	**0.780**	0.651	**0.790**	0.745	**0.775**
	Brier ↓	0.533	**0.290**	0.470	**0.308**	0.298	**0.224**	0.281	**0.236**	0.235	**0.199**	0.209	**0.197**

- ENUM: The prompt is structured to bind the options to a set of enumerated single-token symbols (such as "A", "B", etc), as illustrated in Fig. 1a. The predicted log-likelihoods $\ell \in \mathbb{R}^m$ over the m options are determined using the log-likelihoods of the associated symbols.

- ITEM: The prompt presents the option as an unordered list, as illustrated in Fig. 1b. The predicted log-likelihoods ℓ are computed as the sum of log-likelihoods of all individual tokens corresponding to each option. Note that ITEM is essentially different from cloze-based prompts used in GPT-3 (Brown et al., 2020), in that ITEM provides all options in the prompt, enabling direct comparison between options similar to ENUM. Since each option is completely specified (LLMs can probably produce an exact copy of one of the answers), no additional normalization (e.g., normalizing by length or unconditional log-likelihood as in (Brown et al., 2020)) is applied.

For both cases, the predicted distribution p over the potential options is computed by normalizing the predicted log-likelihoods, i.e., $p = \mathrm{softmax}(\ell/T_p)$, where T_p is the temperature hyper-parameter. By default, $T_p = 1$ is applied. In (Kadavath et al., 2022), it is shown that for instruction-tuned LLMs that exhibit overconfidence, using a calibration-adjusting $T_p' > 1$ can significantly improve their poor calibration. However, the optimal T_p^* for calibration may vary across different tasks (as shown in Table 8). It is often impractical to tune T_p^* in a task-specific manner in real-world scenarios, as this would require access to labeled data for calculating calibration metrics. In our empirical study (Table 1), we demonstrate that the effectiveness of CAPE is orthogonal to adjusting T_p and CAPE can consistently improve LLM calibration with various temperatures without the need of any labeled data.

Ensemble aggregation Given a MC question and n augmented prompts, we can obtain n predictive distributions $\{p_1, p_2, \ldots, p_n\}$ from each prompt. We take the average as the ensemble predictive distribution $\bar{p} = \sum_{i=1}^{n} p_i/n$, for computing the ensemble accuracy and calibration.

Computational cost CAPE distinguishes itself from conventional ensembling methods by establishing the ensemble during downstream inference instead of model training, which eliminates the need to train multiple LLMs for the purpose of ensembling. As with the conventional ensemble method, CAPE incurs an $O(n)$ computational overhead at inference time, trading off for improved calibration. Nevertheless, as the prompt ensemble is applied to a single LLM during inference, CAPE does not require multiple models, reducing space complexity from $O(n)$ to $O(1)$. The prompt ensemble can be evaluated in a batch, potentially reducing the introduced inference cost.

3. Experiments

In this section, we demonstrate the effectiveness of CAPE for enhancing LLM calibration in the MC setup by conducting an extensive empirical study of our prompt ensemble on MC tasks. Results on open-ended generation tasks are in Appx. C. Experimental details and additional results are in Appx. D and Appx. E, respectively.

Models Our primary results are carried out with the `text-davinci-003` model via the OpenAI API. We chose to focus on an instruction-tuned model because these models are currently being deployed but exhibit poor calibration (Kadavath et al., 2022; OpenAI, 2023). We also show how our results generalize to other `davinci`-series models (`text-davinci-001/002`), smaller-scale models (`text-curie-001`), and base models (`davinci`) in Appx. E.1. We found that for instruction-tuned models like `text-davinci-003`, few-shot in-context examples do not improve the model performance by much. Therefore, we focus on the zero-shot setup in the main paper and include the results for few-shot setups in Appx. E.2.

Evaluation metrics As measures of calibration and performance in our experiments, we report expected calibration error (ECE), AUROC, Brier score, and accuracy. Detailed descriptions of each metric are given in Appx. D.1. In some cases, we observed certain tradeoffs between ECE (calibration) and AUROC (discrimination), similar to the observation in (Kadavath et al., 2022).

Setup We run our experiments on four diverse MC tasks: MMLU (Hendrycks et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), and TruthfulQA (Lin et al., 2021). For TruthfulQA, we used the multiple-choice split with a single correct answer. Due to budget constraints, we randomly sub-sampled 200 samples from each dataset for our evaluation, as we found the results to be similar to those using 5x more samples (Appx. E.3). For CAPE, we augmented the prompt using template paraphrase and option permutation in the zero-shot setup. We used 8 augmented prompts by default to achieve a balance between better calibration and increased cost. The single prompt baseline (no prompt ensemble) is denoted as BASE.

Temperature adjustment In (Kadavath et al., 2022), it was observed that increasing the normalization temperature T_p can largely calibrate the model predictions. Note that, unlike CAPE which is purely unsupervised, temperature adjustment requires a labeled dataset and is orthogonal to our method. We illustrate the efficacy of CAPE in various scenarios involving different degrees of temperature adjustment: (i) $T_p = 1$, the default temperature without any adjustment; (ii) $T'_p = 10$ for ENUM and 5 for ITEM, which were selected by optimizing the ECE of the standard, single prompt approach on MMLU; and (iii) T^*_p, which is the optimal temperature selected by optimizing ECE *independently* for each dataset. Although the latter choice is typically impractical, we use it to investigate the potential gain achievable with CAPE upon the optimal T^*_p adjustment. To optimize ECE for purposes of (ii) and (iii), we considered temperatures in $\{0.25, 0.5, 1, 2.5, 5, 10, 20, 50\}$.

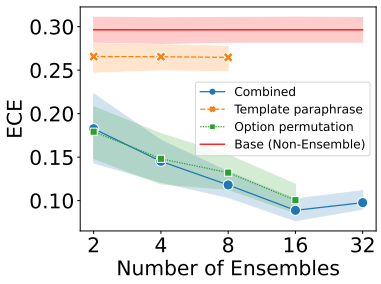


Figure 2: Comparison of prompt augmentations at various ensemble sizes CAPE (5 seeds)

How does CAPE improve uncertainty estimation over individual prompts (BASE)? As shown in Table 1, CAPE significantly improves the evaluation metrics over BASE with the default temperature $T_p = 1$, for which BASE exhibits very poor calibration. The gain remains nontrivial when a fixed (T'_p) or optimal (T^*_p) temperature adjustment is applied, illustrating that CAPE is complementary to it and versatile in its efficacy. In some rare cases, e.g., HellaSwag with ENUM prompt and $T'_p = 10$, BASE achieved better ECE but at the cost of lower accuracy and AUROC, demonstrating a certain *tradeoff between the two metrics*. Finally, we highlight that CAPE not only maintains the accuracy but, in most cases, also improves it.

Does the multiple-choice prompt type matter? We compare the ENUM and ITEM prompt types in Table 1. We find that the prompt type (along with the predictive distribution normalization) can sometimes have significant effects on model calibration. However, the discrepancy is setup-specific and neither prompt type clearly dominates.

Which prompt augmentation is more effective and can the gains be combined? In Fig. 2, We compare the ECE with different prompt augmentations applied on MMLU: template paraphrase, option permutation, and a combination of both. We observe that both augmentations improve the ECE over individual prompts and the option permutation gives larger gains and scales better with the number of ensembles. The gains of both can be combined, achieving the best calibration.

How does CAPE scale with ensemble size? Fig. 2 studies how the number of ensembled prompts impacts calibration on MMLU. We find that the calibration performance of CAPE improves until $n = 16$. We use $n = 8$ in our main

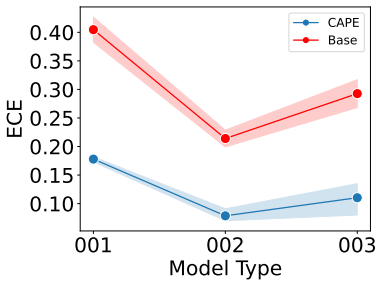


Figure 3: Calibration for different `text-davinci` models (3 seeds)

experiments (Tables 1 and 2), as it provides a nice balance between calibration and compute.

Does CAPE improve other instruction-tuned models? We investigate how CAPE can improve different instruction-tuned models in Fig. 3. Specifically, we evaluated the calibration of BASE and CAPE on MMLU with `text-davinci`-series models. We observe that CAPE leads to consistent improvement over BASE. Though the calibration of the base model degrades from 002 to 003 after RLHF, CAPE effectively recalibrates it.

4. Related Work

In this section, we discuss related work about language model calibaration and the utilization of prompt ensembles for MC tasks. Additional related work is included in Appx. A.

Calibration for Language Models Prior work on the calibration of language models has largely focused on classification or multiple-choice tasks (Desai & Durrett, 2020; Jiang et al., 2021; Desai & Durrett, 2020). Jiang et al. (2021) revealed that smaller models such as T5 (Raffel et al., 2020) and GPT-2 (Radford et al., 2019) display poor calibration on QA tasks by default, but can be improved through finetuning, post-hoc probability modification (including temperature adjustment), or adjustment of the predicted outputs or inputs. Kadavath et al. (2022) found that Anthropic LLMs exhibit reasonable calibration and self-knowledge (of what they know) by default. They found RLHF deteriorated model calibration and temperature adjustment could largely fix the miscalibration. Our method complements these techniques, offering the unique benefit of being entirely unsupervised.

Prompt Ensembles Ensembling multiple prompts for LLMs has been explored in previous work mainly for improving model accuracy in classification or multiple-choice tasks. It has been applied in various scenarios, including boosting predictive accuracy during inference (Jiang et al., 2020; Radford et al., 2021; Izacard et al., 2022), serving as data augmentations during training (Zhou et al., 2022), generating pseudo-labels for model distillation (Schick & Schütze, 2020) or prompt selection (Liao et al., 2022). Similar techniques have also been applied with "soft" prompts (Qin & Eisner, 2021; Lester et al., 2021). We refer interested readers to section 5.1 in Liu et al. (2023) for a more detailed survey of relevant prompt ensemble methods. Our work distinguishes itself from prior work in that: (i) We utilized prompt ensembles for improving model calibration instead of predictive accuracy; (ii) In Appx. B, we demonstrated how prompt ensembles can be applied in generation tasks beyond classification tasks.

5. Discussion

In this paper, we introduced CAPE, a novel prompt ensemble method that improves the calibration of LLMs. By capitalizing on the inherent prompt sensitivity of LLMs, CAPE leverages generally applicable prompt augmentations to build an ensemble that significantly improves calibration across a wide array of natural language tasks. Our empirical study reveals that even subtle prompt augmentations, such as option permutation, can yield surprisingly effective results, pointing to a potential selection bias in LLMs. CAPE is complementary to other LLM calibration methods like temperature adjustment, with the added benefit of being purely unsupervised.

Future work should tackle the limitation of our method concerning the increased inference cost, e.g., by exploring prompt augmentations with a better scaling to reduce the required ensemble size or by finetuning the model as (Lin et al., 2022; Kadavath et al., 2022).

References

Bach, S. H., Sanh, V., Yong, Z.-X., Webson, A., Raffel, C., Nayak, N. V., Sharma, A., Kim, T., Bari, M. S., Fevry, T., Alyafeai, Z., Dey, M., Santilli, A., Sun, Z., Ben-David, S., Xu, C., Chhablani, G., Wang, H., Fries, J. A., Al-shaibani, M. S., Sharma, S., Thakker, U., Almubarak, K., Tang, X., Tang, X., Jiang, M. T.-J., and Rush, A. M. Promptsource: An integrated development environment and repository for natural language prompts, 2022.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Bhatt, U., Antorán, J., Zhang, Y., Liao, Q. V., Sattigeri, P., Fogliato, R., Melançon, G., Krishnan, R., Stanley, J., Tickoo, O., et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 401–413, 2021.

Breiman, L. Random forests. *Machine learning*, 45:5–32, 2001.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Desai, S. and Durrett, G. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*, 2020.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Hafner, D., Tran, D., Irpan, A., Lillicrap, T., and Davidson, J. Reliable uncertainty estimates in deep neural networks using noise contrastive priors. *stat*, 1050:24, 2018.

Hansen, L. K. and Salamon, P. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12 (10):993–1001, 1990.

Havasi, M., Jenatton, R., Fort, S., Liu, J. Z., Snoek, J., Lakshminarayanan, B., Dai, A. M., and Tran, D. Training independent subnetworks for robust prediction. *arXiv preprint arXiv:2010.06610*, 2020.

He, P., Liu, X., Gao, J., and Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Holtzman, A., West, P., Shwartz, V., Choi, Y., and Zettlemoyer, L. Surface form competition: Why the highest probability answer isn't always right. *arXiv preprint arXiv:2104.08315*, 2021.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., and Grave, E. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 2022.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

Jiang, Z., Araki, J., Ding, H., and Neubig, G. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012.

Krogh, A. and Vedelsby, J. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 7, 1994.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation, February 2023. URL `http://arxiv.org/abs/2302.09664`. arXiv:2302.09664 [cs].

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017a.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017b.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., and Chen, W. On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*, 2022.

Liao, C., Zheng, Y., and Yang, Z. Zero-label prompt selection. *arXiv preprint arXiv:2211.04668*, 2022.

Lin, C.-Y. and Och, F. J. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Annual Meeting of the Association for Computational Linguistics*, 2004.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Lin, S., Hilton, J., and Evans, O. Teaching Models to Express Their Uncertainty in Words, June 2022. URL http://arxiv.org/abs/2205.14334. arXiv:2205.14334 [cs].

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

Malinin, A. and Gales, M. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020.

OpenAI. Gpt-4 technical report, 2023.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Perez, E., Kiela, D., and Cho, K. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070, 2021.

Perrone, M. P. and Cooper, L. N. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, Brown Univ Providence Ri Inst for Brain and Neural Systems, 1992.

Pitis, S., Zhang, M. R., Wang, A., and Ba, J. Boosted prompt ensembles for large language models. *arXiv preprint arXiv:2304.05970*, 2023.

Qin, G. and Eisner, J. Learning how to ask: Querying lms with mixtures of soft prompts. *arXiv preprint arXiv:2104.06599*, 2021.

Qin, Y., Wang, X., Beutel, A., and Chi, E. H. Improving uncertainty estimates through the relationship with adversarial robustness. *arXiv preprint arXiv:2006.16375*, 2020.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Robinson, J., Rytting, C. M., and Wingate, D. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*, 2022.

Ruan, Y., Singh, S., Morningstar, W. R., Alemi, A. A., Ioffe, S., Fischer, I., and Dillon, J. V. Weighted ensemble self-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CL-sVR9pvF.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Schick, T. and Schütze, H. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.

Sellam, T., Das, D., and Parikh, A. P. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Wen, Y., Tran, D., and Ba, J. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

Wen, Y., Jerfel, G., Muller, R., Dusenberry, M. W., Snoek, J., Lakshminarayanan, B., and Tran, D. Combining ensembles and data augmentation can harm your calibration. In *International Conference on Learning Representations*, 2021.

Xiao, Y., Liang, P. P., Bhatt, U., Neiswanger, W., Salakhutdinov, R., and Morency, L.-P. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*, 2022.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pp. 12697–12706. PMLR, 2021.

Zhou, C., He, J., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Prompt consistency for zero-shot task generalization. *arXiv preprint arXiv:2205.00049*, 2022.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. Additional Related Work

Calibration on Open-ended Generation Task Some recent work has studied calibration or uncertainty in natural language generation setups. Lin et al. (2022) fine-tuned GPT-3 to generate uncertainty estimates in natural language and demonstrated that the calibration gain is not solely attributable to output logits. Kadavath et al. (2022) proposed a "self-evaluation" method that instructs LLMs to self-evaluate the correctness of their own generated answers. They also found fine-tuning provides significant enhancements in calibration, though a generalization gap persists. Kuhn et al. (2023) measured the semantic entropy of the generated texts for a particular question from OPT models (Zhang et al., 2022) and used it to estimate the uncertainty for the particular question. Compared to existing methods, our method differs in that: (i) it provides multiple semantically distinct candidate generations with confidence scores assigned to each, in contrast to producing uncertainty only for the maximum-likelihood generations (Lin et al., 2022; Kadavath et al., 2022) or the entire question (Kuhn et al., 2023); and (ii) it requires no additional fine-tuning thus avoiding the potential generation gap.

Model Ensembles Ensembling (Hansen & Salamon, 1990; Perrone & Cooper, 1992) is a well-established method for improving both performance and calibration in image classification domains (Lakshminarayanan et al., 2017b; Ovadia et al., 2019). In language modeling, Jiang et al. (2020) demonstrated that prompt ensembles can enhance performance on classification tasks, while other studies have reported modest gains by ensembling different few-shot prompts (Li et al., 2022; Pitis et al., 2023). Regarding calibration, Xiao et al. (2022) conducted a large-scale empirical study, finding that ensembling with individually trained models improves calibration for LLMs, though not as significantly as temperature scaling (Bhatt et al., 2021). Our proposed CAPE method does not necessitate individually trained models for ensembling and is more akin to test-time augmentation, commonly employed in computer vision (Krizhevsky et al., 2012). Augmentation-based ensembling shares similarities with other single-model ensemble techniques that leverage multiple heads, shared weights, or stochasticity to create an ensemble without independently parameterizing each member (Lee et al., 2015; Gal & Ghahramani, 2016; Osband et al., 2016; Wen et al., 2020; Havasi et al., 2020; Ruan et al., 2023).

Integrating Data Augmentation and Ensembles While it might seem natural to assume that combining data augmentation with ensembles would enhance calibration, existing research presents conflicting evidence. Though based on train-time augmentation and not quite comparable to our work, certain studies (Hafner et al., 2018; Hendrycks et al., 2019) demonstrate improved calibration when combining data augmentation and ensembling, while others (Qin et al., 2020; Wen et al., 2021) reveal the opposite. Consequently, we should not trivially assume that calibration can be consistently enhanced by combining data augmentation and ensembling.

B. Generalizing CAPE to Open-Ended Generation

In this section, we extend CAPE to the more general and practical setting of open-ended generation (OG). In OG tasks, a context (e.g., a question) is provided and the objective is to generate a coherent and contextually relevant continuation (e.g., a semantically correct answer). We start by discussing the challenges of uncertainty estimation in OG setups, and then introduce a novel approach that casts OG tasks into MC selection problems, where CAPE can be naturally applied as in Sec. 2.

Addressing the Challenges of Uncertainty Estimation in Open-Ended Generation

We enumerate several key challenges of uncertainty estimation in OG setups (cf. the similar discussion in (Kuhn et al., 2023)) and discuss how we propose to tackle them:

Infinite sample space For OG tasks, the sample space includes sequences of tokens with indefinite lengths. This makes it intractable to marginalize over sequences with the same semantic meaning to address the surface-form competition described below. We instead consider obtaining a compact set of candidate generations sampled from the model predictions, similar to prior work such as (Kadavath et al., 2022; Kuhn et al., 2023).

Surface-form competition LLMs often generate samples that are semantically equivalent but have different syntactic or lexical forms, introducing the issue of surface-form competition (Holtzman et al., 2021). To address this, (Kuhn et al., 2023) pointed out that the estimated uncertainty should be computed over the "semantic" space defined by the semantic content of the sequences instead of their plain lexical forms, such that the uncertainty estimate will not be distributed across sequences that are semantically equivalent. We utilize two strategies to alleviate the surface-form competition: (i) we explicitly prompt LLMs to generate diverse and semantically distinct samples instead of naively sampling from model predictions multiple times with a sampling temperature $T_s > 0$ as in previous work (e.g., (Kadavath et al., 2022; Kuhn et al., 2023)), as shown in Step 1 of Fig. 4; and (ii) following (Kuhn et al., 2023), we cluster the generated samples into semantically equivalent groups using a bidirectional entailment classifier, as shown in Step 2 of Fig. 4.

Scoring candidate generations Another key challenge

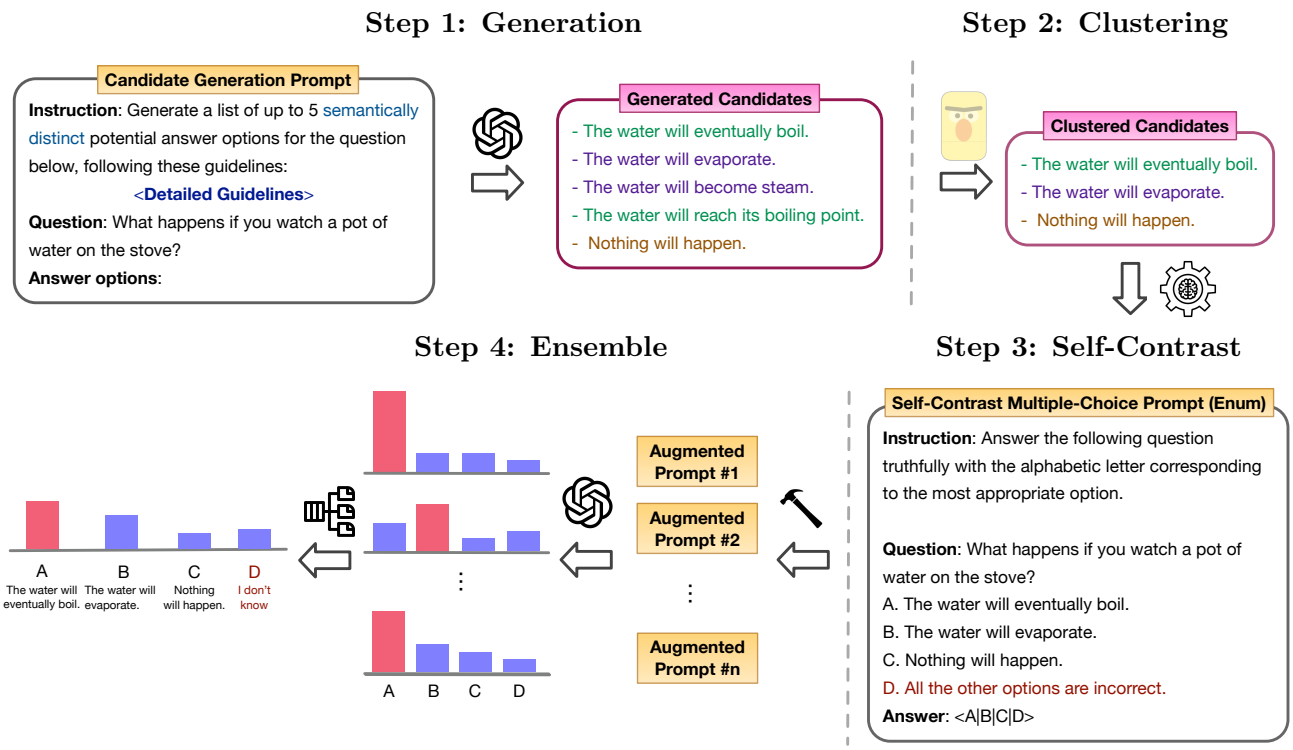


Figure 4: **CAPE for Open-Ended Generation(OG).** CAPE casts the uncertainty estimation for OG into a MC selection problem. It consists of four major steps: **(1)** The LLM is prompted to generate a diverse set of candidates based on the given context. **(2)** The candidates that are semantically equivalent are clustered to alleviate surface-form competition (Holtzman et al., 2021; Kuhn et al., 2023). This step is optional if Step 1 already generates semantically distinct candidates. **(3)** The LLMs are prompted to "self-contrast" the clustered candidates and select the one that best aligns with the context as a MC selection problem, producing a normalized predictive distribution to form predictions and uncertainty estimates. An additional IDK option is provided to accommodate the case that the generated options are low-quality. **(4)** CAPE for MC (Sec. 2) is applied to augment and ensemble multiple "self-contrast" MC prompts.

is to produce a predictive distribution or uncertainty estimate over candidate generations that genuinely captures the relative quality of the candidates. Prior work typically scores candidate LLM generations according to their log-likelihoods (conditioned on the provided context), optionally normalizing them by length or unconditional log-likelihood (Brown et al., 2020). However, recent studies have highlighted the limitations of using the conditional log-likelihood as a reliable proxy for making accurate predictions (Robinson et al., 2022) or estimating uncertainty (Jiang et al., 2021).

We note that the prior approach suffers from another crucial limitation: it scores each candidate generation independently, without taking into account the other candidates. However, the relative probability of generating answer A over B does not necessarily indicate that the LLM would prefer A or B if given the explicit choice. Motivated by this, we propose a novel approach that casts the problem of estimating the uncertainty of candidate generations into a MC selection problem as in Sec. 2. In particular, we provide the LLM with the original context and candidate generations, then prompt it to *"self-contrast"* them and pick the one that best aligns with the context, as illustrated in Step 3 of Fig. 4. This approach allows us to prompt the LLM for a diverse set of answers, rather than the best answer. To account for the possibility that all candidates generated are of poor quality, we provide an additional catch-all option,"All the other options are incorrect", which corresponds to a final answer of **IDK** ("I don't know"). While Kadavath et al. (2022) demonstrated degraded accuracy and calibration when a similar catch-all was introduced for MC tasks, we observe a decent improvement of calibration on OG tasks (Table 2).

Our method enjoys the following advantages: (i) it utilizes a direct comparison between candidate generations for scoring each candidate; and (ii) a normalized predictive distribution over candidate generations (and the catch-all IDK) can be naturally obtained for prediction and uncertainty estimation, which can be ensembled over multiple "self-contrast" MC prompts as described in Sec. 2. Our "self-contrast" method exhibits some similarities with the "self-evaluation"

method P(True) proposed by Kadavath et al. (2022). The key difference is that our method directly compares and evaluates all candidate generations, while "self-evaluation" only evaluates the maximum-likelihood generation with other candidates presented in the prompt; see Appx. D.2.3 for a comparison. Empirically, we find that this distinction significantly impacts the performance of uncertainty estimation (Table 2).

CAPE Pipeline for Open-Ended Generation

Motivated by the difficulties of OG uncertainty estimation described above, we now present our full pipeline with CAPE, as illustrated in Fig. 4. Our pipeline consists of four major steps:

1. **Generation**: Prompt the LLMs to generate a diverse set of k candidate generations $\mathcal{S} = \{s_1, s_2, \ldots, s_k\}$ based on a given context o.

2. **Clustering**: Identify the candidate generations with the same semantic content and cluster them into $h \leq k$ semantic clusters $\mathcal{C} = \{c_1, c_2, \ldots, c_h\}$. Each cluster c_i is a disjoint set of semantically equivalent candidate generations, i.e., $c_i = \{s_{i_1}, s_{i_2}, \ldots\}$, with $c_i \cap c_j = \emptyset, \forall i \neq j$. We apply the bidirectional entailment clustering algorithm proposed in (Kuhn et al., 2023). Specifically, for any pair of candidates (s_i, s_j), we use the Deberta-large model (He et al., 2020) fine-tuned for natural language inference to determine their semantic equivalence: s_i and s_j are semantically equivalent iff s_i entails s_j and s_j entails s_i. A more advanced model such as GPT-4 (OpenAI, 2023) could potentially be used to improve the accuracy and efficiency of the clustering algorithm, and we leave it for future work. If candidate generation in Step 1 is done in such a way as to generate candidates that are already semantically distinct, as we found was the case on certain tasks, this step becomes optional.

3. **Self-Contrast**: Prompt the LLMs to "self-contrast" the clustered candidate generations and select the one that best semantically aligns with the context as a MC problem. Specifically, we randomly select one candidate $s_{i_j} \in c_i$ as one potential option for each cluster $c_i \in \mathcal{C}$, and format the MC selection problem with the context as the question and all selected candidates and an additional IDK option as potential options. As discussed in Sec. 2, the "self-contrast" MC problem can be structured with ENUM or ITEM format to obtain a normalized predictive distribution over the provided options (i.e., clustered candidates) to form predictions and uncertainty estimates.

4. **Ensemble**: Ensemble the "self-contrast" predictions over n augmented MC prompts for improving uncertainty estimation, as described in detail in Sec. 2.

In contrast to prior methods, CAPE offers a normalized predictive distribution over multiple semantically distinct candidates, assigning confidence scores for each. This has the potential to be more informative and practically useful than uncertainty estimates that only assess overall uncertainty for the problem (Malinin & Gales, 2020; Kuhn et al., 2023) or measure the confidence for the maximum-likelihood candidate (Kadavath et al., 2022).

C. Open-Ended Generation Experiments

Setup To assess our CAPE pipeline on OG tasks, we chose to focus on free-form QA, as it is an important category of OG for which ground truth answers are relatively easy to determine. We used two tasks for the evaluation: TriviaQA (Joshi et al., 2017) and TruthfulQA (Lin et al., 2021). To evaluate the correctness of a generated answer, we used the Rouge-L score (Lin & Och, 2004) for TriviaQA following (Kuhn et al., 2023) and the BLEURT metric (Sellam et al., 2020) adopted in (Lin et al., 2021) for TruthfulQA. See Appx. D.1 for details.

Baselines The major baseline we compare to is *P(True)* (Kadavath et al., 2022), which prompts the LLM to "self-evaluate" its maximum-likelihood answer with other candidate answers (sampled with $T_s = 1$) incorporated in the context. We also included some uncertainty measures to offer interested readers a more comprehensive context and analysis, including *Predictive Entropy* (the entropy of predictive samples estimated with Monte Carlo integration), *Normalized Entropy* (Malinin & Gales, 2020) (a variant of Predictive Entropy which divides the log-likelihood of each sequence by its length), and *Semantic Entropy* (Kuhn et al., 2023) (the entropy computed over semantic space by clustering semantically equivalent sequences as the clustering step in Fig. 4). We followed (Kuhn et al., 2023) and sampled $k = 5$ candidates from model predictions with $T_p = 1$. We only compute AUROC for these methods because they only estimate the uncertainty for discrimination without providing a predictive distribution or confidence for generated answers. This limits their practical utility because determining the threshold for discriminating correct answers requires a labeled dataset which may not be available.

CAPE Recall that the key distinctions between our CAPE pipeline, as shown in Fig. 4, and the baseline approaches are primarily in the generation of candidates, the "self-contrast" method to produce a predictive distribution over these candidates, and the ensembling of these "self-contrast" predictions. Below, we conduct a careful qualitative analysis to better understand the effectiveness of our pipeline and its main components. We generated 5 candidates to match the baseline setup and used 8 augmented prompts for CAPE as before. For TruthfulQA, we observed semantically distinct generations from our candidate generation, and thus did not

Table 2: **Results on OG tasks.** CAPE leads to substantial gains for model calibration on free-form QA tasks. The gains come from "self-contrast" (BASE vs P(True)), prompt ensemble (CAPE vs BASE), the introduction of an "IDK" option (row "w/o IDK"), and further temperature adjustment (rows highlighted in cyan). We used 8 augmented prompts for CAPE. For BASE (non-ensemble), the means and standard deviations computed with the 8 individual prompts are reported. For temperature adjustment, we used T_p' selected on MMLU as Table 1 *without tuning* it to emulate practical scenarios.

Method	TriviaQA				TruthfulQA			
	Acc ↑	ECE ↓	AUROC ↑	Brier ↓	Acc ↑	ECE ↓	AUROC ↑	Brier ↓
Baselines								
Predictive Entropy	0.695	-	0.784	-	0.390	-	0.502	-
Normalized Entropy	0.695	-	0.721	-	0.390	-	0.515	-
Semantic Entropy	0.695	-	0.801	-	0.390	-	0.517	-
P(True)	0.695	0.285	0.573	0.294	0.390	0.591	0.527	0.586
Our methods								
BASE-ENUM	0.676 ± 0.017	0.272 ± 0.016	0.733 ± 0.024	0.289 ± 0.011	0.371 ± 0.029	0.527 ± 0.051	0.536 ± 0.038	0.533 ± 0.049
BASE-ITEM	0.695 ± 0.019	0.255 ± 0.027	0.769 ± 0.015	0.259 ± 0.023	0.394 ± 0.017	0.485 ± 0.028	0.549 ± 0.046	0.497 ± 0.028
CAPE-ENUM	0.705	**0.128**	0.789	**0.178**	0.365	0.312	0.533	0.350
- w/o IDK	0.690	0.161	0.792	0.194	0.375	0.362	0.501	0.408
- $T_p' = 10$	0.700	0.077	0.801	0.158	0.360	0.178	0.544	0.272
CAPE-ITEM	0.725	0.139	0.769	0.178	0.390	**0.282**	**0.569**	**0.346**
- w/o IDK	0.750	0.146	0.749	0.185	0.415	0.332	0.528	0.388
- $T_p' = 5$	0.710	0.069	0.797	0.150	0.375	0.146	0.600	0.264

apply semantic clustering (Step 2).

How does CAPE improve calibration over the baselines? We compare our methods (including both BASE without ensembling and CAPE) with baselines in Table 2. First, we find that our BASE method with ENUM or ITEM prompts, which "self-contrast" generated candidate answers, both perform much better than the P(True) baseline on all evaluation metrics for uncertainty estimation. Next, ensembling (CAPE-ENUM/ITEM) improves the uncertainty estimation by a large margin over BASE, consistent with the observations in MC tasks. Finally, we incorporated temperature adjustment into CAPE to get insight into the best possible calibration that CAPE can achieve. To emulate a practical and realistic scenario, we directly used temperature T_p' selected on MMLU as Table 1 *without tuning* it on current tasks. We observe that CAPE with temperature adjustment, highlighted in cyan, further boosts model calibration, achieving an ECE of **0.069** on TriviaQA. In terms of AUROC, our methods are comparable to entropy-based uncertainty measures on TriviaQA and outperform them on TruthfulQA. We hypothesize this is due to a property of these datasets: TriviaQA is short-form but TruthfulQA is relatively longer-form which makes Monte Carlo estimation of entropies challenging.

Does IDK affect the model calibration? In Table 2, we investigate the impact of integrating an additional IDK option on model calibration by ablating it from CAPE (denoted as "w/o IDK"). The metrics for uncertainty estimation notably deteriorate upon this exclusion, suggesting the benefits of introducing an IDK option. Kadavath et al. (2022) observed that introducing a similar option degraded accuracy and calibration on MC tasks; we conjecture that the difference stems from the possible lower-quality generations in OG tasks, hence necessitating a catch-all option to deal with such cases.

Table 3: **Candidate generation.** Our BASE-ITEM produces more diverse generations and achieves a better diversity-accuracy tradeoff than naive generation with different sampling temperatures T_s.

Generation	# of Clusters	Acc ↑	ECE ↓	AUROC ↑	Brier ↓
$T_s = 0$	1.00	0.694	0.304	0.649	0.303
$T_s = 0.5$	1.12	0.711	0.283	0.712	0.283
$T_s = 1$	1.31	0.673	0.304	0.730	0.300
$T_s = 2$	1.94	0.494	0.482	0.726	0.474
Ours	4.64	0.695	**0.255**	**0.769**	**0.259**

Does diverse candidate generation help? To understand how our candidate generation (Step 1 in Fig. 4) differs from the naive generation (sampling from models with $T_s > 0$) used in prior work, we compare the performance of BASE-ITEM with different generation methods in Table 3 (in each case, after generating, "self-contrast" is applied). For naive generation, we used different sampling temperatures T_s in $\{0, 0.5, 1, 2\}$. We find that with T_s increasing for naive generation, the model accuracy and calibration first improve and then degrade (peak at $T_s = 0.5$), exhibiting a tradeoff between generation diversity and accuracy. In contrast, our method generates more diverse candidates (implied by a larger number of clusters) while obtains comparable accu-

racy and better calibration. This suggests that our approach achieves a better diversity-accuracy tradeoff with no additional hyperparameters.

D. Experimental Details

D.1. Metrics

Evaluation metrics We measure the LLMs' uncertainty estimation with various evaluation metrics:

- **ECE**: We use expected calibration error (ECE) to measure model calibration. ECE is computed as the mean, across 10 equally sized bins, of the absolute difference between the model's confidence (i.e., the probability of the top prediction) and the accuracy (i.e., how often the model is correct).
- **AUROC**: We also report the AUROC with respect to prediction confidence or other uncertainty measures (e.g. predictive entropies). Higher AUROC is better, with a perfect score of 1 and a chance score of 0.5. Note that AUROC captures the discriminative power of scalar uncertainty measures rather than calibration. We observed certain tradeoffs between calibration (ECE) and discrimination (AUROC), which is consistent with the observation in (Kadavath et al., 2022).
- **Brier Score**: To alleviate the tradeoff described above, we also report the Brier Score, which measures the mean squared difference between the predicted distributions and the actual outcome.

Accuracy metrics on OG tasks For TriviaQA, we followed (Kuhn et al., 2023) and evaluated the correctness of a generated answer with the Rouge-L score (Lin & Och, 2004): an answer is considered correct if its Rouge-L score with respect to the reference answer is larger than 0.3. For TruthfulQA, we used the BLEURT metric (Sellam et al., 2020) adopted in (Lin et al., 2021): an answer is considered correct if its maximum BLEURT score with the correct reference answers is larger than its maximum score with the incorrect reference answers.

D.2. Prompts

D.2.1. Prompt for Template Paraphrase

We provide the prompt that we used to instruct GPT-4 for paraphrasing prompt templates. We include the prompt for generating the ENUM type of prompts on the Hellaswag dataset for illustration, which can be easily generalized to the ITEM prompt type and other datasets (we omit for brevity).

We begin by presenting the instructions and guidelines, followed by two human-written prompts that we adapted from (Bach et al., 2022) as the example prompts. The remaining content is left to be generated by GPT-4.

```
Create a set of 10 diverse
multiple-choice prompt templates by
following the guidelines provided
below.

Guidelines:
1. Each template must be formatted as a
JSON object containing the fields
`instruction' and `template'. The
`instruction' field is optional and
provides task instructions, while the
`template' field is mandatory for
framing specific instances of the task.
Remember that the text within braces
represents fields that can be
substituted into sentences. 'template'
must have three fields of question,
options and answer.
2. Ensure that the structure and
phrasing of templates are diverse,
avoiding repetition in style or format.
You can modify the format and phrasing
of the template field to enhance
diversity, as long as the semantics
remain the same.
3. Develop templates that accommodate a
wide range of topics to enhance the
versatility of the generated
multiple-choice questions.

List of 10 templates:
1. {"template": "From the list of
endings described below, what ending
makes the most sense for the sentence?
Answer it from A, B, C, D.\n\nSentence:
{question}\n{options}\nAnswer:
{answer}"}
2. {"instruction": "Read the following
sentences and choose the possible
endings from A, B, C, D.", "template":
"Question: {question}\n{options}\nThe
correct ending is: {answer}"}
3.
```

D.2.2. Prompt for Candidate Generation

We provide the complete prompt for candidate generation (Step 1 in Fig. 4) below.

```
Generate a list of up to 5 distinct
potential answer options for the
question below, following these
guidelines:
1. Ensure each generated answer option
is semantically distinct from all
previous options for the question.
Avoid simply rephrasing or reiterating
the previous options to generate a new
option.
2. Stop generating answer options if no
more options can satisfy the above
requirements, even if the list contains
fewer than 5 options.

Question: English economist and
physician Nicholas Barbon helped to
pioneer which type of insurance in
1666?
Answer options:
1.
```

D.2.3. PROMPT FOR SELF-CONTRAST

We provide an example prompt for self-contrast (Step 3 in Fig. 4) with the ENUM prompt on the TriviaQA dataset below. In this example, the model generated five answers: {"Greece", "Finland", "Sweden", "Greece", "The Kingdom of Denmark."}. These answers result in four distinct clusters: {"Finland", "Sweden", "Greece", "The Kingdom of Denmark."}. We present an exemplar of a self-contrast prompt that is augmented with a specific ENUM prompt paraphrase and option permutation as follows:

```
Pick the correct answer to the
following trivia question from the
provided options. Answer with the
capital letter of the correct option.

Question: What European country has 227
inhabited islands?
A: The Kingdom of Denmark.
B: Sweden
C: All the other options are incorrect.
D: Finland
E: Greece

Answer:
```

For comparison purposes, we include the prompt for P(true) (Kadavath et al., 2022) with the same example below:

```
Question: What European country has 227
inhabited islands?
Here are some ideas that were
brainstormed: Greece \n Finland \n
Sweden \n Greece \n The Kingdom of
Denmark
Possible answer: Finland
Is the possible answer:
A: True
B: False
The possible answer is:
```

E. Additional Results

E.1. CAPE with Other Models

`davinci`-series models We evaluated the performance of CAPE with different `davinci`-series models in Table 4. The experimental setup exactly followed Table 1 except that different models were used. We find that for instruction-tuned (`text-davinci-001/002/003`) models that exhibit overconfidence and poor calibration, CAPE significantly improves over BASE. However, for the pretrained model (`davinci`) that is already well-calibrated, CAPE may hurt the calibration performance. We conjecture that in this case, ensembling may "over-smooth" the model predictive distributions to be underconfident, leading to the degradation of calibration.

Table 4: **Results on `davinci`-series models.** CAPE significantly improves over BASE for instruction-tuned (`text-davinci-001/002/003`) models that exhibit overconfidence and poor calibration, but may hurt the calibration for pretrained model (`davinci`) that is already well-calibrated. We used the MMLU dataset and the ENUM format in a zero-shot setup. For CAPE, 8 augmented prompts are applied. For BASE, the means and standard deviations computed with the 8 individual prompts are reported.

Model	Method	Acc ↑	ECE ↓	AUROC ↑	Brier ↓
`davinci`	BASE	0.326±0.018	**0.093±0.029**	0.572±0.049	**0.217±0.008**
	CAPE	**0.429**	0.152	**0.643**	0.253
`text-davinci-001`	BASE	0.480±0.014	0.396±0.021	0.689±0.029	0.396±0.025
	CAPE	**0.506**	**0.182**	**0.733**	**0.240**
`text-davinci-002`	BASE	0.665±0.021	0.217±0.025	0.764±0.025	0.241±0.021
	CAPE	**0.703**	**0.070**	**0.806**	**0.161**
`text-davinci-003`	BASE	0.657±0.022	0.302± 0.026	0.778±0.027	0.305±0.021
	CAPE	**0.675**	**0.135**	**0.835**	**0.174**

`curie`-scale model We assessed the effectiveness of CAPE on the smaller-scale `text-curie-001` model. We used the MMLU dataset for illustration. The ITEM prompt type was applied because we found ENUM did not work well on small-scale models, possibly due to the lack of symbolic binding capability (Robinson et al., 2022). The other experimental setups exactly followed Table 4. We find that `text-curie-001` with BASE also exhibits poor calibration and CAPE substantially improves its calibration.

Table 5: **Results on `text-curie-001` model.** `text-curie-001` with BASE also exhibits poor calibration and CAPE substantially improves its calibration. We used the MMLU dataset and the ITEM format. Other experimental setups exactly followed Table 4.

Model	Method	Acc ↑	ECE ↓	AUROC ↑	Brier ↓
`text-curie-001`	BASE	**0.296±0.016**	0.440±0.033	0.542±0.024	0.427±0.031
	CAPE	0.293	**0.261**	**0.560**	**0.287**

E.2. CAPE in Few-Shot Setups

For our experiments in the main paper, we mainly focused on zero-shot setups because instruction-tuned `text-davinci` models already work well. Here, we demonstrated how our results can be generalized to few-shot setups.

Setup We used `text-davinci-003` model with the default temperature $T_p = 1$ on the MMLU dataset. We included 4 in-context examples in the prompt which were randomly sampled from the dataset. The sampled in-context examples were fixed and shared across all test samples. We applied the ENUM prompt type. For CAPE, we augmented the prompt using all prompt augmentations described in Sec. 2. We used 8 augmented prompts by default as before.

How does CAPE improve uncertainty estimation in few-shot setups? As shown in Table 6, CAPE significantly improves the calibration metrics over BASE in the few-shot setups. Interestingly, compared to the zero-shot results in Table 1, both the calibration metrics of BASE, CAPE improve, implying that adding few-shot examples could improve model calibration which is consistent with the observation in (Kadavath et al., 2022).

Table 6: **Results in few-shot setups.** CAPE significantly improves calibration with few-shot examples. Compared to the corresponding zero-shot results in Table 1, the calibration metrics for both BASE and CAPE improve. We used the ENUM prompt type on the MMLU dataset, 4 in-context examples, and the default temperature $T_p = 1$. For CAPE, we used 8 augmented prompts. For BASE, the means and standard deviations computed with the 8 individual prompts are reported.

Method	Acc ↑	ECE ↓	AUROC ↑	Brier ↓
BASE	0.692±0.021	0.251±0.015	0.799±0.030	0.253±0.012
CAPE	**0.710**	**0.102**	**0.850**	**0.153**

Which prompt augmentation is more effective and can the gains be combined? We compare the ECE with different prompt augmentations for few-shot setups in Fig. 5: template paraphrase, option permutation, in-context example ordering, in-context example selection, and a combination of all. Similar to the observation in zero-shot setups (Fig. 2), we observe that all augmentations improve the ECE over individual prompts and their gains can be combined, achieving the best calibration. Option permutation and in-context example selection provide the largest gains and scale better with the number of ensembles, compared to the other individual augmentations.

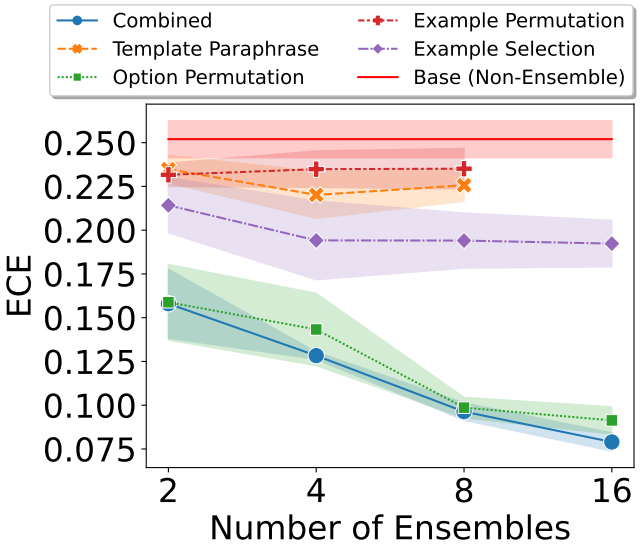


Figure 5: **Comparison of prompt augmentations at various ensemble sizes in few-shot setups.** All prompt augmentations provide calibration improvements and their gains can be combined. Option permutation and in-context example selection provide the most significant gains and scale better with the number of ensembles. 3 seeds are run and the error bars are the standard deviations.

E.3. Additional Analysis

How does the number of data samples affect the evaluation? We compared the evaluation results with 200 and 1000 data points on the HellaSwag dataset in Table 7. We find that using 200 data points for evaluation yields similar results (within the error bar) as using 1000 data points, and does not affect the relative comparison between BASE and CAPE. Therefore, due to budget constraints, we mainly used 200 data points for evaluation in our experiments to reduce cost.

E.4. Calibration Plots

We include the calibration plots in Figs. 6 to 9 corresponding to the results in Table 1 for visualizing the prediction patterns (accuracy vs confidence) of BASE and CAPE. For each plot, we use 10 bins with an equal number of predictions in each. We plot the calibration with the default

Table 7: **Evaluation results with a different number of data points**. Using 200 data points yields similar evaluation results and relative comparisons as using 1000 data points and the deviation across different data splits is small. We used the ENUM prompt type on the HellaSwag dataset. For CAPE, we used 8 augmented prompts. We reported the means and standard deviations across 3 data split seeds for the results with 200 data points.

# data	Method	Acc ↑	ECE ↓	AUROC ↑	Brier ↓
200	BASE	0.597±0.015	0.312±0.011	0.673±0.008	0.327±0.009
	CAPE	0.737±0.013	0.096±0.007	0.732±0.015	0.175±0.008
1000	BASE	0.602	0.302	0.683	0.318
	CAPE	0.743	0.084	0.742	0.172

temperature $T_p = 1$ and the fixed adjusting temperature $T'_p = 10$ for the ENUM prompt type.

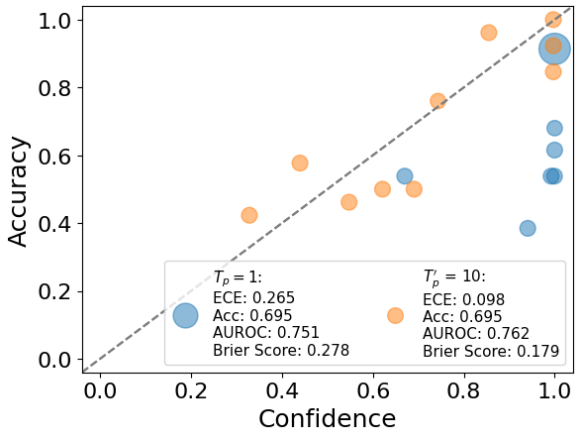
E.5. Full Results for Reference

We include the full results of Table 1 in Table 8.

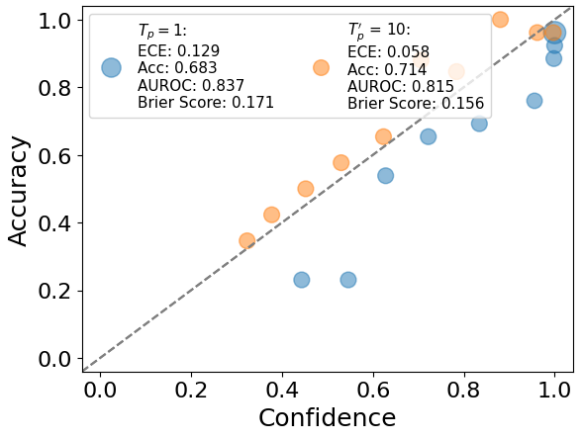
F. Analysis

F.1. Ensemble Ambiguity as a Selection Criteria

Given the large space of possible prompt augmentations, how we pick a subset to serve as our prompt ensemble? In their pioneering work, Krogh & Vedelsby (1994) propose an "ambiguity decomposition" for ensembles, which decomposes the MSE of the ensemble (Brier score) as a the sum of the mean individual classifier error and the ensemble "ambiguity" or variance. In Fig. 10, we see that we can use ambiguity to guide our prompt selection in an unsupervised manner. These figures were formed by subsampling ensembles of 8 from a stratified sample of various ensemble compositions. We considered the following ensemble compositions: 8 different option permutations, 4 different option permutations and 2 template paraphrases, 2 different option permutations and 4 template paraphrases, and finally, 8 template paraphrases. From each, we sampled various ensembles of size 8 to apply to the entire dataset, and plotted the ambiguity against the ECE and the accuracy. We see that ensembles with larger ambiguity tend to be better calibrated and more accurate. This suggests using ambiguity as a selection criteria for ensemble composition. Note that, for purposes of this experiment/ablation, the stratified sampling approach constrains the ensemble composition, thereby reducing ambiguity—in our main experiments, we use 8 template paraphrases and 8 option permutations within each ensemble in order to maximize ambiguity.



(a) BASE

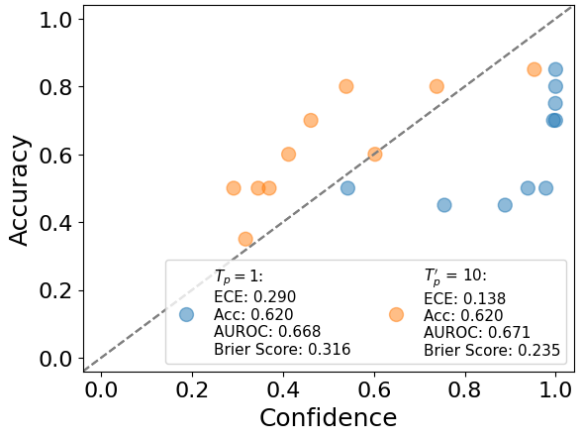


(b) CAPE

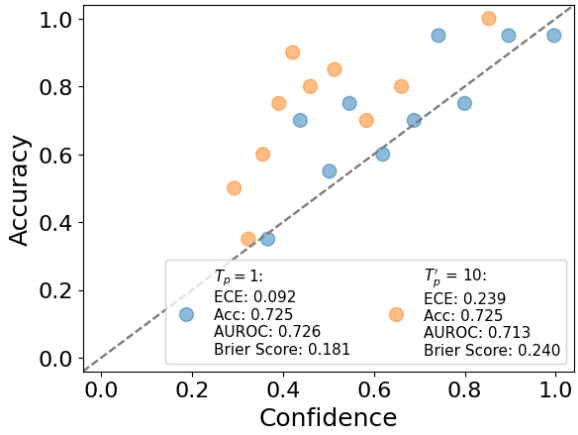
Figure 6: **Calibration plots of BASE and CAPE on MMLU.** For both the default temperature ($T_p = 1$) and the fixed adjusting temperature ($T'_p = 10$), applying CAPE consistently enhances calibration. The ENUM prompt was used and 8 augmented prompts were applied for CAPE.

Table 8: **Full Results of Table 1 on MC tasks.** CAPE consistently and significantly improves model calibration on MC tasks with varying degrees of temperature scaling (Kadavath et al., 2022). $T_p = 1$ is the default temperature, T_p' is the adjusted temperature tuned on the MMLU dataset, and T_p^* is the optimal temperature tuned on each dataset independently. The selected T_p^*s are recorded as two tuples for each dataset with the first tuple for ENUM and the second for ITEM, the first element in each tuple is for BASE and the second for CAPE. For CAPE, 8 augmented prompts are applied. For BASE, the means and standard deviations computed with the 8 individual prompts are reported.

Dataset	Measure	ENUM, $T_p = 1$		ITEM, $T_p = 1$		ENUM, $T_p' = 10$		ITEM, $T_p' = 5$		ENUM, T_p^*		ITEM, T_p^*	
		BASE	CAPE	BASE	CAPE	BASE	CAPE	BASE	CAPE	BASE	CAPE	BASE	CAPE
MMLU	Acc ↑	0.657 ± 0.022	0.675	0.669 ± 0.024	**0.715**	0.657 ± 0.022	**0.705**	0.669 ± 0.024	**0.710**	0.657 ±0.022	**0.705**	0.669 ± 0.024	**0.710**
	ECE ↓	0.302 ±0.026	**0.135**	0.274 ±0.028	**0.105**	0.081 ±0.017	**0.068**	0.094 ±0.021	**0.057**	0.081 ±0.017	**0.068**	0.094 ±0.021	**0.057**
$T_p^* = (10, 10), (5, 5)$	AUROC ↑	0.778 ±0.027	**0.835**	**0.749** ±0.026	0.742	0.787 ±0.028	**0.816**	0.752 ±0.030	**0.771**	0.787 ±0.028	**0.816**	0.752 ±0.030	**0.771**
	Brier ↓	0.305 ±0.021	**0.174**	0.280 ±0.021	**0.194**	0.176 ±0.009	**0.159**	0.191 ±0.010	**0.169**	0.176 ±0.009	**0.159**	0.191 ± 0.010	**0.169**
HellaSwag	Acc ↑	0.587 ± 0.046	**0.735**	0.583 ± 0.023	**0.665**	0.587 ± 0.046	**0.730**	0.583 ±0.023	**0.660**	0.587 ± 0.046	**0.730**	0.583 ±0.023	**0.665**
	ECE ↓	0.319 ± 0.037	**0.103**	0.332 ±0.022	**0.086**	**0.124** ± 0.034	0.250	0.102 ±0.014	**0.087**	0.120 ± 0.045	**0.084**	0.102 ±0.014	**0.057**
$T_p^* = (5, 0.25), (5, 2.5)$	AUROC ↑	0.668 ± 0.039	**0.700**	0.684 ± 0.032	**0.710**	0.669 ± 0.032	**0.695**	0.694 ± 0.031	**0.715**	0.671 ± 0.032	**0.696**	0.694 ± 0.031	**0.713**
	Brier ↓	0.334 ± 0.037	**0.185**	0.339 ± 0.021	**0.203**	**0.236** ± 0.024	0.245	0.221 ± 0.010	**0.203**	0.236 ±0.032	**0.186**	0.221 ± 0.010	**0.195**
WinoGrande	Acc ↑	0.619±0.034	**0.640**	0.629±0.024	0.635	0.619±0.034	**0.640**	0.629±0.024	0.635	0.619±0.034	**0.635**	0.629±0.024	0.630
	ECE ↓	0.349±0.029	**0.224**	0.340±0.023	**0.255**	0.195±0.046	**0.135**	0.227±0.024	**0.181**	0.112±0.031	**0.099**	0.087±0.016	0.076
$T_p^* = (50, 20), (20, 20)$	AUROC ↑	0.565±0.040	**0.596**	0.612±0.029	**0.632**	0.546±0.047	**0.585**	0.612± 0.030	**0.639**	0.546±0.047	**0.614**	0.612±0.030	**0.647**
	Brier ↓	0.361±0.028	**0.279**	0.351±0.022	**0.296**	0.277±0.018	**0.239**	0.281±0.014	**0.255**	0.238±0.005	0.230	0.225± 0.005	0.221
TruthfulQA	Acc ↑	0.416±0.048	**0.440**	0.466±0.035	0.460	0.416±0.048	0.415	0.466±0.035	0.470	0.416±0.048	0.415	0.466±0.035	0.465
	ECE ↓	0.542±0.048	**0.294**	0.483±0.035	**0.340**	0.286±0.026	**0.213**	0.268±0.035	**0.206**	0.120±0.021	**0.104**	0.104±0.016	**0.092**
$T_p^* = (50, 20), (10, 10)$	AUROC ↑	0.683±0.034	**0.749**	0.745±0.028	**0.806**	0.697±0.0326	**0.802**	0.749±0.026	**0.780**	0.651±0.018	**0.790**	0.745±0.022	**0.775**
	Brier ↓	0.533±0.045	**0.290**	0.470±0.032	**0.308**	0.298±0.018	**0.224**	0.281±0.019	**0.236**	0.235±0.012	**0.199**	0.209±0.007	**0.197**

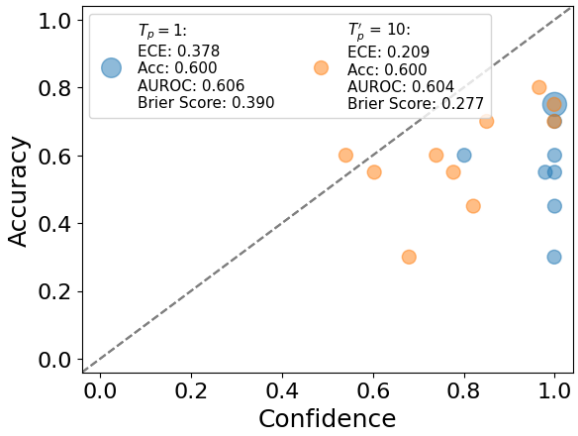


(a) BASE

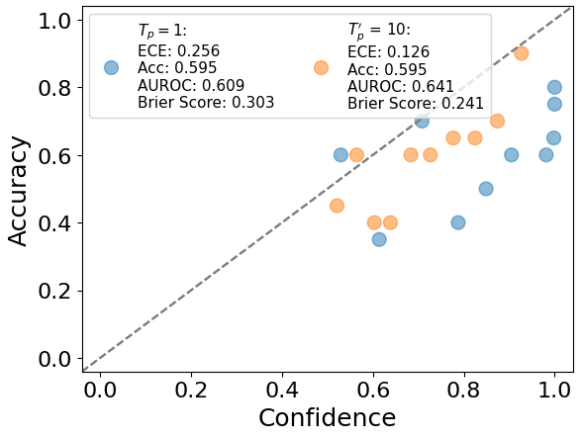


(b) CAPE

Figure 7: **Calibration plots of BASE and CAPE on Hellaswag.** The effectiveness of CAPE in calibration varies depending on the temperature settings. When $T_p = 1$, CAPE demonstrates improved calibration. However, when $T'_p = 10$, the calibration performance could be negatively affected, possibly because ensembling could "over-smooth" the predictive distributions to be underconfident. The ENUM prompt was used and 8 augmented prompts were applied for CAPE.

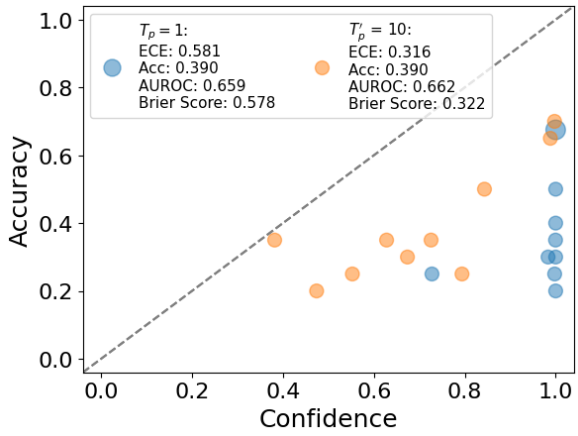


(a) BASE

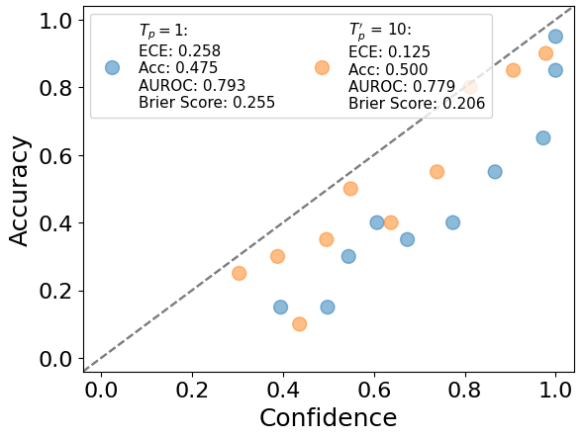


(b) CAPE

Figure 8: **Calibration plots of BASE and CAPE on Winogrande.** For both the default temperature ($T_p = 1$) and the fixed adjusting temperature ($T'_p = 10$), applying CAPE consistently enhances calibration. The ENUM prompt was used and 8 augmented prompts were applied for CAPE.



(a) BASE



(b) CAPE

Figure 9: **Calibration plots of BASE and CAPE on TruthfulQA.** For both the default temperature ($T_p = 1$) and the fixed adjusting temperature ($T'_p = 10$), applying CAPE consistently enhances calibration. The ENUM prompt was used and 8 augmented prompts were applied for CAPE.

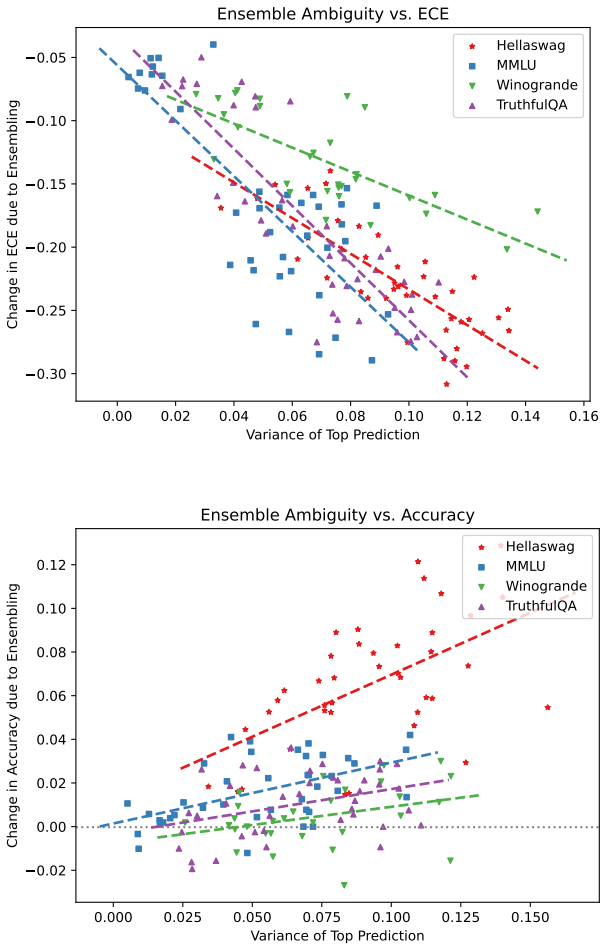


Figure 10: **Ensemble Ambiguity as a Selection Criteria.** Plotting the change in calibration error due to ensembling **(left)**, and the change in accuracy due to ensembling **(right)**, against the variance of the ensemble's top prediction (ensemble ambiguity), we see that ensembles with larger ambiguity tend to be better calibrated and more accurate. This suggests using ambiguity as a selection criteria for ensemble composition.