Inconsistencies in Masked Language Models

Anonymous ACL submission

Abstract

Learning to predict masked tokens in a sequence has been shown to be a helpful pretraining objective for powerful language models such as PaLM2. After training, such masked language models (MLMs) can provide distributions of tokens in the masked positions in a sequence. However, this paper shows that distributions corresponding to different masking patterns can demonstrate considerable inconsistencies, i.e., they cannot be derived from a coherent joint distribution when considered together.

This fundamental flaw in MLMs can lead to self-contradictory behaviors during inference. On various benchmark datasets including MMLU, MLMs can give different predictions to the same input question. From BERT-base to UL2-20B, we show that such inconsistencies exist ubiquitously in MLMs of diverse sizes and configurations. In light of our observations, we further propose an inference-time strategy for MLMs called Ensemble of Conditionals. It jointly considers a selected range of inconsistent conditionals directly produced by the MLM for the final prediction, which often leads to considerable accuracy improvement.

1 Introduction

Pretraining objectives of large language models can be roughly divided into two categories. First, vanilla next token prediction (also known as casual language modeling) aims to learn the distribution of the next token in a sequence given the context to the left (Brown et al., 2020). Second, the masked language modeling (MLM) objective, which masks out a portion of the tokens in a sequence and asks the model to predict them, aims to learn the distribution of one or more tokens given surrounding context (Devlin et al., 2018; Raffel et al., 2020).

While GPT-3 (Brown et al., 2020) used vanilla next token prediction, following work such as PaLM-2 (Anil et al., 2023), U-PaLM (Tay et al., 2022b), GPT-FIM (Bavarian et al., 2022), UL2 (Tay et al., 2022a), and GLM (Zeng et al., 2022) have hinted that incorporating the MLM objective could be highly beneficial to performance. In addition, Tay et al. (2022b) has demonstrated that such bidirectional conditionals provide strong infilling capabilities. Empirically speaking, predicting masked tokens in the middle of the sentence can be seen as a natural data augmentation technique to vanilla next token prediction, which might be helpful to alleviating the data scarcity problem (Xue et al., 2023) in the current large model era.

One may notice that, unlike the unidirectional conditional distributions that vanilla next token prediction learns, the bidirectional conditionals that MLMs learn are overly abundant in terms of representing a coherent joint distribution. Therefore, they are not guaranteed to be self-consistent. This paper explains our effort on exposing and quantifying this issue and corresponding strategies during inference.

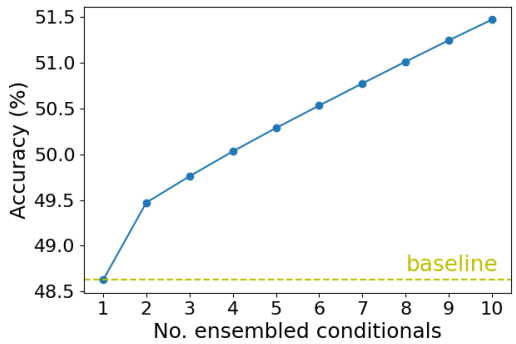

Figure 1: Self-ensembling improves MLMs' accuracies on standard benchmarks including MMLU, Lambada and BigBench. Aggregated results based on Figure 5.

To begin with, a simple example for such inconsistencies is shown in Figure 2. In this example, we obtain the bidirectional conditional distributions

that the T5 model learned using two input masked sequences. The two similar sequences are designed with a small difference, in order to examine if the resulting conditionals satisfy a basic law of probabilities (hold consistency). Results clearly show otherwise. We design experiments to quantify such inconsistencies on benchmark datasets in Section 4.2. We further show an inference-time ensemble algorithm in Section 4.3 which utilizes many inconsistent conditionals for a more accurate prediction. We demonstrate that ensembling the numerous inconsistent conditionals directly provided by the MLM can improve its performance (Figure 1).

In summary, our contributions are (1) We expose the commonly overlooked flaw in MLMs that they can represent inconsistent distributions depending on the mask patterns. (2) We quantify such inconsistencies in benchmark datasets including Lambada (Paperno et al., 2016), MMLU(Hendrycks et al., 2021) and BigBench (Srivastava et al., 2023). For example, on multiple choice questions in MMLU, 2 different distributions given by UL2-20B disagree on the answer 14% of the time on average. (3) We show that the numerous inconsistent conditionals can be ensembled together to considerably improve accuracy on said benchmarks.

2 Why inconsistencies can occur in MLMs

For a set of conditional distributions to be self-consistent, they need to be able to be derived from a single coherent joint distribution.

One essential reason for the inconsistencies to occur among the conditionals provided by a trained MLM is that the number of conditionals it can provide far *exceeds* the degrees of freedom of a joint distribution.

Consider a sequence of length L with vocabulary V. The joint distribution of the tokens in such a sequence is defined by $|V|^L$ probabilities that sum to 1. Therefore, the degrees of freedom (D) of such a joint distribution is:

$$D_{joint} = |V|^L - 1, \qquad (1)$$

Both vanilla next token prediction models and MLMs essentially learn conditionals that predict some tokens in the sequence given others. Such conditional probabilities and probabilities from the joint distribution can be linearly derived from each other. Therefore, each free conditional that the language model is capable of specifying places a constraint on the joint distribution. One can easily verify (by counting the conditionals left to right for a geometric sequence) that a vanilla next token prediction based language model provides just $|V|^L - 1$ free conditionals[1] to exactly determine the joint distribution. Therefore, a vanilla next token prediction model (no matter how it is trained, or even untrained) would never suffer from inconsistencies among its conditionals.

MLMs, which can provide distributions of masked tokens given bidirectional context, could specify far more free conditionals. For the simplest case, where the MLM predicts the distribution of only 1 (masked) token given $L - 1$ other (unmasked) tokens in the sequence, the total number of free conditionals (N) is

$$N_{mlm}(1) = L \times (|V|^L - |V|^{L-1}), \qquad (2)$$

Just $N_{mlm}(1)$ is already far larger than D_{joint}. Not to mention $N_{mlm}(k)$ for $k \in [2, N-1]$. See Appendix B for $N_{mlm}(k)$ and both of their derivations. The fact that the number of conditionals an MLM provides far exceeds what is needed for defining a joint distribution sets up room for there to be inconsistencies among them.

The first portion of our experiments (Sections 4.2 & 5) focus on exposing and quantifying the inconsistencies that exist among the conditionals provided by common MLMs. The second portion of our experiments (Section 4.3) demonstrates our new inference-time algorithm "Ensemble of Conditionals" that unites them for more accurate predictions.

To begin with, the next section explains the backbone models that this paper works with.

3 Backbone MLMs

We work with 3 different MLMs in this paper that belong to two different styles, which can be called the T5-style and the BERT-style.

3.1 T5-style

For T5-style MLMs, the definition here is that each mask token in the input functions as a placeholder for the prediction of an entire span of tokens of variable length. Below we introduce 2 different

[1] A single softmax operation over V essentially gives $|V| - 1$ free conditionals. Here we call conditionals free when they can be assigned any values decided by an underlying neural network.

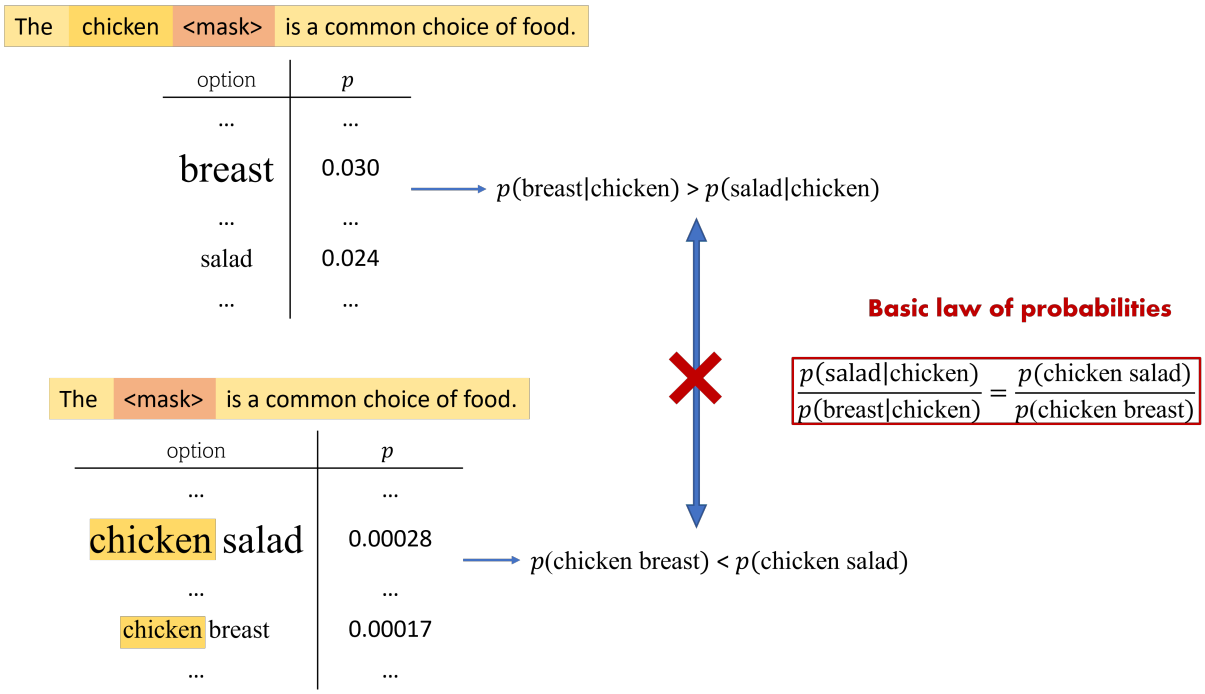

Figure 2: A simple bigram comparison example that exposes the inconsistencies in the T5 model. The conditional probabilities that the model learned (quoted from T5-11B fed with the shown masked sequences) contradict each other greatly. Not only are the ratios unbalanced, the model confuses its own preference of the two bigrams.

T5-style MLMs that we will work with in the experiments. They differ in their architecture design, masking strategies and sizes.

1. T5

   The T5 model (Raffel et al., 2020) uses an Encoder-Decoder architecture. It uses a corruption rate of 15% and an average span length of 3 tokens. The masked spans can be anywhere in the sequence. We use the largest model T5-11B in the experiments.

2. UL2-20B

   The UL2 model (Tay et al., 2022a) follows T5's architecture design and aims to mix up 3 masking strategies to more comprehensively utilize the pretraining corpus. The MLM objective is also known as the auto-denoising objective, since the masks can be considered as adding noise to the sequence. UL2 calls masking strategies denoisers.

   • The R(Regular)-Denoiser mimics T5's masking scheme.

   • The S(Sequential)-Denoiser simply partitions the input sequence into two consecutive sub-sequences and predict the second subsequence as the masked sequence.

   • The X(Extreme)-Denoiser is an extreme version of denoising marked by long corrupted spans or high corruption rates. The X-Denoiser is aimed as an interpolation between R- and S-Denoiser.

   Tay et al. (2022a) showed that such a mixture of masking strategies achieved a superior performance than T5 on many tasks. The 3 different denoisers were differentiated by 3 respective sentinel tokens ([R], [S], [X]) prepended to the sequence. These sentinel tokens are also used during inference to invoke the corresponding behavior from the model. Without losing generality, we restrict ourselves to the X-Denoiser in our experiments due to its superior performance in our pilot trials.

3.2 BERT-style

Our definition for BERT-style MLMs, named after BERT (Devlin et al., 2018), is that the model uses each mask token as the placeholder for the prediction of exactly one real token. We use the better-trained RoBERTa (Liu et al., 2019) for our experiments as our example for BERT-style MLMs, which shares the same architecture as BERT. While considered somewhat deprecated (Tay et al., 2022a) compared to later MLMs like T5, UL2 and PaLM2,

BERTs are unique in terms of their architecture design because they use a single transformer with bi-directional attention (or, an Encoder-only architecture), as opposed to GPTs (Radford et al., 2018; Brown et al., 2020), which use a transformer with uni-directional attention (Decoder-only) or the T5 model (Encoder-Decoder).

Our paper mainly focuses on the inconsistencies in T5-style MLMs since they are most useful in practice (Section 4). But we also touch on BERT-style MLMs due to its unique architecture and historical impact (Section 5).

4 Inconsistencies in T5-style MLMs

4.1 Conditionals for various mask patterns

This section lists a few different types of conditional distributions that a trained T5-style MLM can give depending on the mask pattern. This sets up for the next two Sections (4.2 and 4.3)which discusses their inconsistencies and how to ensemble them on various benchmark datasets.

First, we discuss the baseline conditional distribution (first row in Figure 3). Since most NLP tasks can be formulated as predicting continuing tokens given an input sequence, we consider the use case of MLMs where we append a single [MASK] token behind the input sequence (Tay et al., 2022a). The MLM takes as input this modified sequence to generate a distribution of tokens for the [MASK] position, which is essentially our distribution of interest for the target tokens.

Tweaking the mask pattern can make the MLM generate different values for our target distributions of tokens. We consider two types of mask patterns: the K-offset pattern and the Multimask pattern.

1. The K-offset mask pattern additionally masks the last K tokens from the input sequence ($K = 3$ in the second row in Figure 3), and feed them to the MLM as ***given*** output. For example, for Encoder-Decoder models like UL2, we feed K starting tokens to the decoder instead of the usual 0[2]. The model then generates a different version of our distribution of interest. Because of the inconsistency issue, this new distribution is often remarkably different than the one from the baseline.

2. The Multimask pattern additionally masks N random spans in the input sequence ($N = 1$ in the third row in Figure 3). This pattern is also parameterized by span length S and gap length G between spans. Recall that masking multiple spans is a common practice during the pretraining of MLMs. When our input contains multiple [MASK]'s, we feed additional tokens to the decoder, which correspond to the masked tokens in the N spans. The Multimask pattern will prompt the MLM to generate another different version of our conditional of interest.

While our K-offset and Multimask conditionals may seem contrived at first glance, they potentially represent different knowledge learned by the model during pretraining. And as we will show in the sections next, they often contradict the baseline conditional and can be complementary to it.

To begin with, we select a number of specific mask patterns through parameterization of K-offset and Multimask patterns. This is done on the validation set of the evaluation dataset based on their individual accuracies. For any dataset in Lambada, MMLU and BigBench and any model in UL2 and T5, we always consider 10 types of conditional distributions as our set of interest. For example, for the combination of UL2 and Lambada we consider the baseline conditional, 6 K-offset conditionals ($K \in [1, 6]$), and 3 Multimask conditionals ($(N, S, G) \in \{(3, 5, 1), (3, 5, 2), (3, 10, 1)\}$). See the full list of patterns in Appendix C.

4.2 Exposing inconsistencies

To quantitatively expose the severity of the inconsistencies among the numerous conditionals we use 3 benchmark datasets.

1. Lambada (LAnguage Modeling Broadened to Account for Discourse Aspects): Lambada is a dataset crafted to test the capabilities of computational models in language understanding, particularly in predicting the final word of a text passage when it requires understanding the broader context. This dataset focuses on the challenge of word prediction requiring a broad discourse context, aiming to evaluate if models can effectively utilize long-range dependencies in text. To evaluate inconsistency and ensembling on Lambada, we use the baseline conditional to generate on average 5 last words as candidates.

[2]For decoder-only MLMs like PaLM2, the input tokens and the K tokens are simply concatenated.

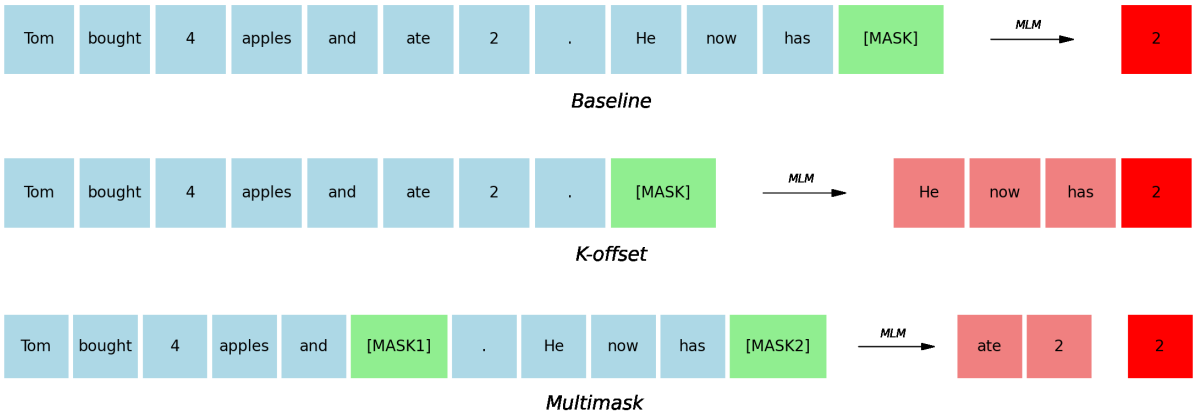

Figure 3: K-offset and Multimask patterns. The goal here is to prompt the MLM for different versions of the target token distribution. The red token is our target token. The coral tokens are taken from the original input sequence and fed as starting tokens to the decoder of the MLM.

2. MMLU (Massive Multitask Language Understanding): The MMLU benchmark represents a leap towards evaluating the comprehensive knowledge acquired by models during pre-training. It encompasses a wide array of subjects, spanning elementary to advanced professional levels across STEM, humanities, and social sciences. MMLU aims to understand the depth and breadth of models' knowledge and reasoning abilities. MMLU is a multiple choice dataset from which the model chooses the best answer given the input question.

3. BigBench (Beyond the Imitation Game Benchmark): BigBench focuses on challenging tasks and aim to evaluate and understand models' performance across a spectrum of complexities and subject areas. This benchmark is designed not just to test models but also to highlight potential areas for future research and development. Similar to MMLU, BigBench is also a multiple choice dataset.

We show that these incoherent conditionals often disagree on which answer is the best for a multi-choice question in MMLU and BigBench or which word is the best for last word prediction in Lambada. We demonstrate such incoherence on the three datasets by measuring how often the distributions cannot agree on the prediction.

For example, consider a toy last word prediction task mimicking Lambada: `The cutest cat breed in the world is the [MASK]`. While the baseline conditional might rank `Munchkin` the highest, another conditional under our consideration might rank `Persian` the highest. We choose between 2 to 10 conditionals from our set of interest. When we choose less than 10 conditionals, all possible combinations are run and the results are aggregated. We count how often the conditionals cannot agree on the prediction. Figure 4 shows that there exists considerable disagreement among the different conditionals we consider. And as expected, the more conditionals are considered, the more likely there is disagreement. But even with 2 conditionals, the disagreement can be as high as 20%. However, disagreement converges to an upper bound. This means that there exists some questions in every benchmark on which the model is "confident" on its answer.

4.3 Ensemble of Conditionals

Since we have shown that there are considerable inconsistencies among the conditionals corresponding to different masking patterns, it is worth investigating the potential benefit of ensembling them at inference time.

The gist of Ensemble of Conditionals (EOC) is to put the numerous raw conditionals provided by a trained MLM through an ensemble heuristic. EOC can be seen as a self-ensemble approach where the different outputs provided by one model are ensembled together, similar to ensembling outputs from multiple models in traditional ensemble learning.

To ensemble different conditionals for a final prediction, we use the max-pooling approach[3]. Con-

[3]Outperforms average-pooling in our pilot experiments.

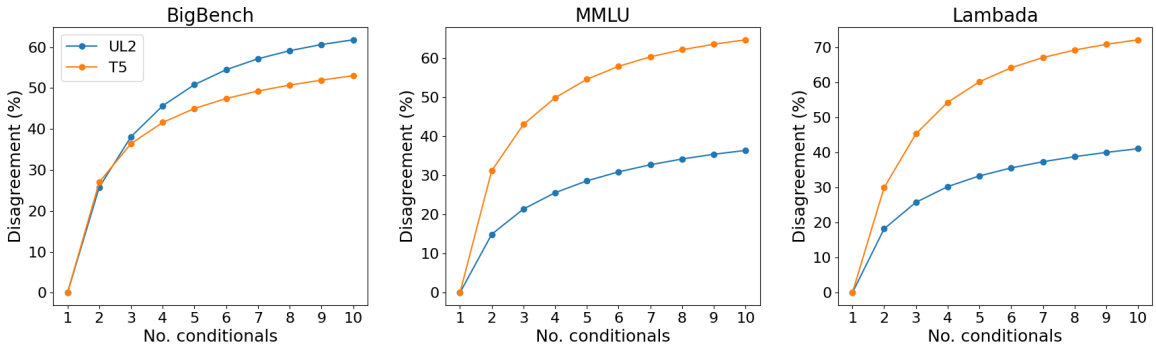

Figure 4: Different conditionals disagree on the prediction to make.

sider that the ith competing conditional assigns probability p_{ij} to the jth candidate completion (either a last word candidate in Lambada or an answer from a multiple choice question in MMLU). The winning conditional and final completion prediction is

$$\hat{i}, \hat{j} = \arg\max p_{ij}, \tag{3}$$

In our experiments, we progressively ensemble more conditionals to observe accuracy changes. Similar to in the experiment on disagreement, when the number of ensembled conditionals is less that the total 10, all combinations are run and the results are aggregated. Results in Figure 5 show that EOC can improve the accuracy of the model's final prediction. In additional, more ensembled conditionals can lead to higher accuracy.

5 Inconsistencies in BERT-style MLMs

T5-style MLMs have the flexibility of generating sequences of variable length and are very useful in practice. Although researchers mainly focus on T5-style MLMs in the current era, we touch on inconsistencies in BERT-style MLMs in this section because of the historical impactfulness of the BERT model and their unique architecture with only bidirectional attention.

While BERT-style models can only model the distributions of individual tokens by their default design, there has been research effort (Goyal et al., 2021; Wang et al., 2019; Yamakoshi et al., 2022) on sampling sequences from it by modeling its implicitly specified joint distribution one way or another. For example, Goyal et al. (2021) views it as an energy-based model defined using the bidirectional conditionals of the masked tokens. Such research effort is based on the intuition that bidirectional conditionals could be more robust than unidirectional conditionals (Goyal, 2021). This line of research has operated based on the assumption that the overly abundant bidirectional conditionals that the BERT-style MLMs provide are self-consistent.

We demonstrate in this section that this is not the case at all. There are considerable inconsistencies that exist among the bidirectional conditionals that a trained BERT-style model provides. Figure 6 demonstrates such an example. Since BERT-style models do not easily offer token distributions for completions, here we use bigrams in raw unstructured text to expose the inconsistencies instead of on standard benchmarks.

We consider 4 bigrams in a surrounding context: $x_{11}x_{21}$, $x_{11}x_{22}$, $x_{12}x_{21}$ and $x_{12}x_{22}$. x_{11} and x_{12} are two possible tokens that the first position can take; x_{21} and x_{22} the second. One can easily verify[4] that the 8 conditional distributions concerning such four bigrams should theoretically satisfy

$$\frac{p(x_{21}|x_{11})}{p(x_{22}|x_{11})} \times \frac{p(x_{11}|x_{22})}{p(x_{12}|x_{22})} = \frac{p(x_{11}|x_{21})}{p(x_{12}|x_{21})} \times \frac{p(x_{21}|x_{12})}{p(x_{22}|x_{12})} \tag{4}$$

One way to test the inconsistencies among the 8 conditionals is to try to solve one using the other 7 and compare the solved conditional with the original (inferred by model) one. We show the solved conditionals in the example in Figure 6. It clearly demonstrates that the probabilities given by a BERT-style MLM can be in serious inconsistencies with each other.

We use the first segment of the validation partition of the C4 (Raffel et al., 2020) dataset as the

[4]Clue: converting each fraction term using the basic law in Figure 2. Equation 4 was discussed in (Arnold and Press, 1989).

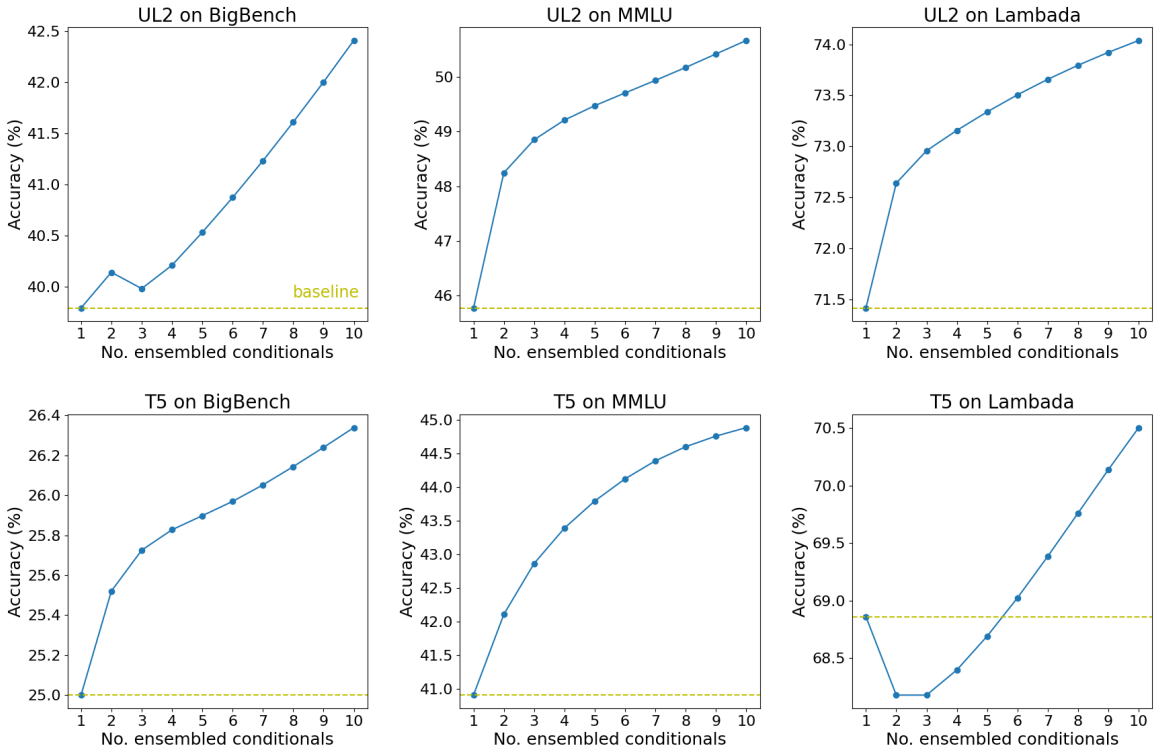

Figure 5: EOC improves MLM accuracy

Table 1: Difference of log-probabilities between inferred and solved conditionals. The difference would be 0 for self-consistent MLMs. Roughly a 0.8 difference means that one is 120% larger than the other.

Metric	RoBERTa-base	RoBERTa-large
log-probability difference ($d_{\log p}$)	0.836	0.792

unstructured text corpus for quantification. Our goal here is to come up with quadruples of bigrams in the form of ($x_{11}x_{21}$, $x_{11}x_{22}$, $x_{12}x_{21}$, $x_{12}x_{22}$) in a certain context. We perform a full bigram sweep for the sequence. We always include the original bigram into the quadruple. To find the 3 alternative bigrams, we mask the whole original bigram, and generate alternatives using BART (Lewis et al., 2019). In practice, we use beam search in `bart.generate()` with beam size 50. Note that BART by default is a T5-style MLM therefore it can generates multiple tokens for one mask. We keep all resulting generations that are 2 tokens. We verify if there is a quadruple of bigrams in the generations in the said fashion and add them to our diagnostics dataset if so. We end up with 7431 quadruples. The following is an example in our diagnostics dataset.

Original sequence: `Brown cats are the most common type of pets in America.`

Original bigram: `Brown cats` ($x_{11}x_{21}$).

Alternative bigrams: `Brown dogs` ($x_{11}x_{22}$), `White cats` ($x_{12}x_{21}$), `White dogs` ($x_{12}x_{22}$).

To obtain conditionals in Figure 6, we mask the bigram with two `[MASK]`'s and feed the sequence to Roberta.

Note that there are many variables that go into building the diagnostics dataset. Our approaches were automatic but also somewhat unprincipled. We are not surprised if variations in the diagnostics dataset could result in some differences in the evaluation results.

We quantify inconsistencies using difference of log probabilities[5].

$$d_{\log p} = |\log p_{\texttt{solved}} - \log p_{\texttt{inferred}}| \quad (5)$$

Table 1 shows the results, which clearly indicate strong inconsistencies among the bidirectional conditionals provided by the RoBERTa model.

[5]Here using logarithm makes it robust against changes in scale. One may also use other metrics for quantification.

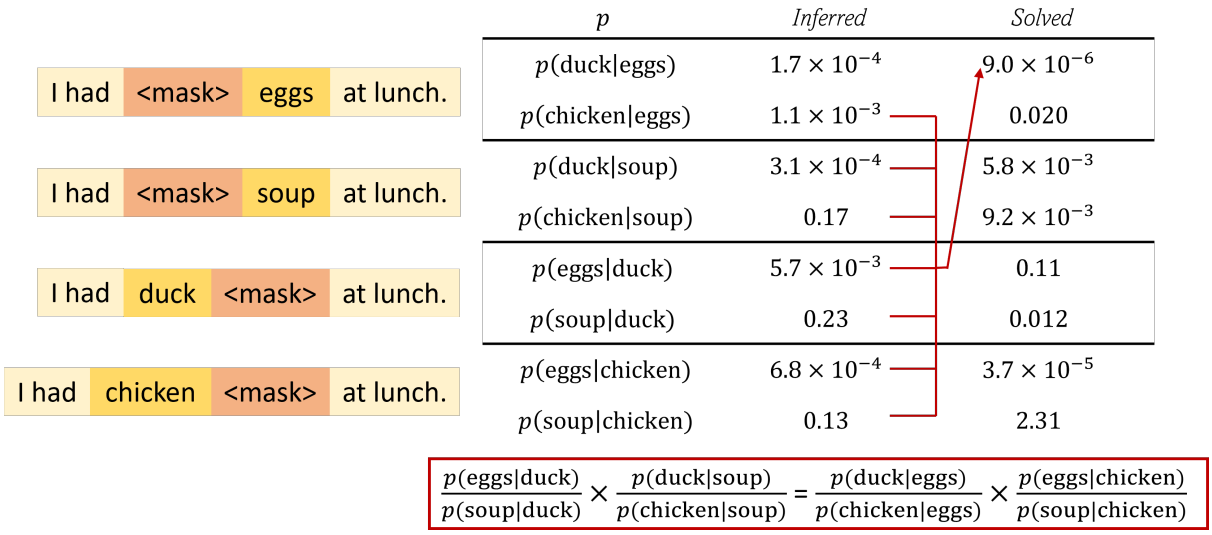

Figure 6: Inconsistencies in the BERT-style MLM. Each "inferred" value refers to the probability given by the MLM (RoBERTa-large in this figure). Each "solved" value is obtained by passing the other 7 "inferred" values to the equation in the red square. We see that the difference between each inferred and solved value is significant (the solved value may even be larger than 1).

6 Summary & Discussions

This paper focused on the inconsistency problem concerning the conditionals provided by MLMs. We demonstrated and quantified the inconsistencies that exist in large MLMs. Based on our observations, we propose an inference-time approach that ensembles multiple inconsistent conditionals to improve the models' performance. The inconsistencies originate from the fact that the number of bidirectional conditionals MLMs can learn far exceeds what is needed for constructing the joint distribution. Given the recent evidence that MLM-based pretraining is a useful paradigm, we think that resolving its inconsistency issue could be a necessary step for future work. While our inference-time ensembling approach improves accuracy, it can only be seen as a limited patch-up method that only unite a certain number of selected conditionals. We believe that for long-term research, this problem should be ideally addressed as part of the expensive pretraining stage, for which our experiment techniques and results can be seen as a reference.

Such inconsistencies may remind readers of GPT's sensitivity to prompts (OpenAI, 2023). It's crucial to understand that those sensitivities refer to inconsistencies in the space of semantics, which are distinct from the focus of our discussion. The inconsistencies highlighted in this paper address the peculiarities of MLMs in the fundamental space of token distributions.

Limitations

1. The discussion in Section 2 only specified a prerequisite for inconsistencies. As for why such inconsistencies mechanistically form during training and how they might be mitigated or avoided during training, we leave the research to future work.

2. Although we tested mid-sized MLMs such as UL2-20B, it is no secret that some powerful masked language models like U-PaLM and PaLM2 are kept out of open access and they might behave somewhat differently. We leave diagnostics on those models for researchers with access. We don't expect the issue to completely disappear for those models.

3. Apart from pretraining, it has been shown that paradigms like instruction tuning (Wei et al., 2021) and reinforcing (Ouyang et al., 2022) can improve the performance of language models. How those techniques interplay with the inconsistency phenomenon is worth looking into.

Impact Statements

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Barry C Arnold and S James Press. 1989. Compatible conditional distributions. *Journal of the American Statistical Association*, pages 152–156.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

K. Goyal. 2021. Characterizing and overcoming the limitations of neural autoregressive models. *PhD thesis*.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2021. Exposing the implicit energy networks behind masked language models via metropolis–hastings. *arXiv preprint arXiv:2106.02736*.

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

P. Langley. 2000. Crafting papers on machine learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1207–1216, Stanford, CA. Morgan Kaufmann.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *TMLR*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022a. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q Tran, David R So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, et al. 2022b. Transcending scaling laws with 0.1% extra compute. *arXiv preprint arXiv:2210.11399*.

Alex Wang, Kyunghyun Cho, and CIFAR Azrieli Global Scholar. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. *NAACL HLT 2019*, page 30.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To repeat or not to repeat: Insights from scaling llm under token-crisis. *arXiv preprint arXiv:2305.13230*.

Takateru Yamakoshi, Thomas L Griffiths, and Robert Hawkins. 2022. Probing bert's priors with serial reproduction chains. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3977–3992.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

A Why not use Llama in the experiments?

Llama and Llama2 are causal autoregressive LLMs that did not utilize the MLM training objective (not mentioned in paper). We expect the MLM pretraining objective to be a useful supplementary to them.

B No. bidirectional conditionals specified by MLMs

$N_{mlm}(1)$ is given by:

$$\begin{aligned} N_{mlm}(1) &= L \times |V|^{L-1} \times (|V| - 1) \\ &= L \times (|V|^{L} - |V|^{L-1}) \end{aligned} \tag{6}$$

L represents how many positions the predicted one token could be in. The number of variations of the surrounding context of length $L-1$ is $|V|^{L-1}$. Given the surrounding context and the position of the predicted token, the number of free conditionals is $|V|-1$ (we assume a BERT-style MLM here; a T5-style MLM naturally provides distributions of tokens of a variable amount). Multiplying the 3 numbers together gives Equation 6.

One may also consider $N_{mlm}(k)$ for BERT-style MLMs, where the k masked tokens can be anywhere in a sequence of L tokens. Note that BERT-style MLMs by default do not model the joint distribution of the k tokens. Instead it models their individual marginal distributions conditioned on the context, which we let $N_{mlm}(k)$ denote the number of here.

$N_{mlm}(k)$ is given by:

$$N_{mlm}(k) = \binom{L}{k} \times |V|^{L-k} \times (|V| - 1)^{k} \tag{7}$$

In Equation 7 (same as Equation 2), $\binom{L}{k}$ represents how many combinations of positions the predicted k tokens could be in. The number of variations of the surrounding context of length $L-k$ is $|V|^{L-k}$. Given the surrounding context and the positions of the predicted tokens, the number of free conditionals is $(|V|-1)^k$.

One can easily see that the number of conditionals an MLM provides far exceeds what is needed for defining a joint distribution, which sets up room for there to be inconsistencies among them. We omit detailed discussions for the number of conditionals provided by T5-style MLMs here.

C Mask patterns

1. UL2 on MMLU. $K \in [1, 6]$; $(N, S, G) \in \{(3, 5, 1), (3, 5, 2), (3, 10, 1)\}$
2. UL2 on Lambada. $K \in [1, 6]$; $(N, S, G) \in \{(3, 5, 1), (3, 5, 2), (3, 10, 1)\}$
3. UL2 on BigBench. $K \in [1, 3]$; $(N, S, G) \in \{(3, 5, 1), (3, 5, 2), (3, 3, 1), (3, 3, 2), (3, 4, 1), (3, 4, 2)\}$
4. T5 on MMLU. $K \in [1, 3]$; $(N, S, G) \in \{(3, 5, 1), (3, 5, 2), (3, 3, 1), (3, 3, 2), (3, 4, 1), (3, 4, 2)\}$
5. T5 on Lambada. $K \in [1, 6]$; $(N, S, G) \in \{(3, 5, 1), (3, 5, 2), (3, 10, 1)\}$
6. T5 on BigBench. $K \in [1, 3]$; $(N, S, G) \in \{(3, 5, 1), (3, 5, 2), (3, 3, 1), (3, 3, 2), (3, 4, 1), (3, 4, 2)\}$

   Some subjects (subsets) in MMLU and BigBench are very challenging for mid-sized models like UL2-20B and T5-13B. We report on subjects that the baseline has a decent performance on (accuracy > 0.4).