MODIFYING MEMORIES IN TRANSFORMER MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Transformer models have achieved impressive performance in many natural language tasks. In particular, Transformer based language models have been shown to have great capabilities in encoding factual knowledge in their vast amount of parameters. While the tasks of improving the memorization and generalization of Transformers have been widely studied, it is not well known how to make transformers forget old facts and memorize new ones. In this paper, we propose a new task of *explicitly modifying specific factual knowledge in Transformer models while ensuring the model performance does not degrade on the unmodified facts*. This task is useful in many scenarios, such as updating stale knowledge, protecting privacy and eliminating unintended biases stored in the models. We benchmarked several approaches that provide natural baseline performances on this task. This leads to the discovery of key components of a Transformer model that are especially effective for knowledge modifications. The work also provides insights on the role of training phases such as pretraining and finetuning in the model's ability in memorization and knowledge modification.

1 INTRODUCTION

Large-scale Transformer based language models (Vaswani et al., 2017; Devlin et al., 2018; Radford et al., 2019; Raffel et al., 2019; Brown et al., 2020) have pushed state-of-the-art results on standard natural language processing (NLP) benchmarks such as GLUE and SQuAD. They have also been proven crucial in improving various real-world systems (see, e.g., Nayak, 2019; Rao et al., 2019).

Given that these models are pretrained on a large corpora of text such as Wikipedia and BookCorpus (Zhu et al., 2015), it's intuitive that they are able to *implicitly memorize* the factual knowledge in their large number of parameters. This hypothesis has been verified recently (Petroni et al., 2019; Roberts et al., 2020), where language models are evaluated on factual knowledge based tasks without supporting evidence. It was shown that large pretrained Transformer based language models achieve non-trivial performance on various open-domain question answering (QA) tasks that probe the factual knowledge stored in their parameters.

The memorization capability of Transformers opens up many exciting opportunities. In addition to the great generalization capability, Transformers may also replace or assist traditional knowledge bases (KBs) that are either manually curated or require significant amount of supervision (Roth & Yih, 2002; Kambhatla, 2004; Surdeanu & Ji, 2014). Different from conventional knowledge bases that memorize factual knowledge explicitly, Transformer memorizes knowledge implicitly in its model parameters. But they lack one key advantage of the conventional databases: *efficiently removing or modifying* the factual knowledge stored in the model. On the other hand, in conventional databases like SQL or NoSQL that explicitly store knowledge in the forms of structured tables, key–value pairs, wide columns, graphs, or documents, the knowledge can be easily updated.

Can Transformers cope with the ever changing world where knowledge is constantly being added, updated and deprecated? To answer this question, we propose a new task of *explicitly modifying specific factual knowledge in Transformer models while ensuring that model performance does not degrade on the unaltered facts*. This task is useful in many scenarios. For example, the factual knowledge stored by the model can become stale over time, which needs to be updated periodically, e.g. a sports player may play with different teams over time. Users may ask a Transformer-based assistant model to update certain knowledge (factual or otherwise) that they asked model to memorized in the past, e.g., their favorite tourist destination. In the context of privacy one may need to overwrite unintendedly memorized sensitive information without retraining the model(Carlini et al., 2019).

Furthermore, language models are susceptible to various biases present in the large corpora of text used for their training, and such biases may need to be eliminated to ensure a fair application of such models in real-world (Bolukbasi et al., 2016; Bordia & Bowman, 2019; Blodgett et al., 2020).

To the best of our knowledge, this is the first work studying how to effectively modify the factual knowledge memorized by Transformers. The paper makes the following contributions.

- We create a new benchmark to evaluate the ability of Transformer models in modifying learned factual knowledge as desired while preserving the model's performance on the unmodified factual knowledge (§ 3.1).
- We formulate this objective as a constrained optimization problem, and identify the best way to enforce the constraint in Transformer models. (§ 3.3)
- We show that constrained layer-wise finetuning is a simple yet effective way to modify the knowledge memorized by Transformers (§ 4).
- We find that it is not easier to modify the memorized facts of memory networks like FaE (Verga et al., 2020) than Transformer-based models without symbolic memory (§ 4).

2 Related Works

Traditionally, KBs are commonly utilized to store and access the relational knowledge in NLP domain (Ji et al., 2020; Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005, inter alia). However, given the recent success of Transformer-based language models on multitude of NLP tasks have fueled an increasing number of efforts on exploring the ability of these language models to serve as unstructured/non-symbolic KBs.

Language models as a source of factual knowledge. To assess the performance of off-the-self modern language models as KBs, Petroni et al. (2019) introduced LAMA (LAnguage Model Analysis) probing method that convert various facts and fact-seeking question-answer pairs into cloze sentences. Petroni et al. (2019) concluded that pretrained BERT (Devlin et al., 2018) shows factual knowledge that is competitive with KBs generated using some of the traditional off-the-self techniques. Further, Roberts et al. (2020) probed the knowledge within T5 models (Raffel et al., 2019) and found very promising results. Another line of work (Sun et al., 2019; Zhang et al., 2019; Peters et al., 2019) focuses on leveraging the readily available structured KBs to further complement the knowledge possessed by language models. Also, there have been attempts of developing novel Transformer models and/or training procedures that aim to leverage both available high-quality KBs and large corpora of (unstructured) text (Dhingra et al., 2019; Guu et al., 2020; Lewis et al., 2020), further broadening the scope of factual knowledge. However, unlike structured KB, which are accompanied by infrastructure for querying, inferring, or updating facts, neural language models do not poses such capabilities directly. Jiang et al. (2020) explores designs for better prompts to query the knowledge implicitly stored in the model parameters of a neural language model. To best of our knowledge, however, there has been no work on designing efficient ways to modifying knowledge in neural language model, which is the focus of our present work.

Memory augmented Transformer models. Multiple recent research efforts augment the Transformer models with explicit long-term memory modules to increase their factual knowledge. Use of memory augmented neural networks had been explored in pre-Transformer era as well (Weston et al., 2014; Sukhbaatar et al., 2015). More recently, in the context of Transformer models, Févry et al. (2020) utilized an explicit key-value memory to store entity representations, which are trained along with the rest of model in an end-to-end manner. Verga et al. (2020) build on Févry et al. (2020), and introduced Facts as Expert (FaE) model with explicit symbolic memory of $(\text{subject}, \text{relation}, \text{object})$ triples based on end-to-end trained entity representations. Notably, one of the motivations behind FaE is the ease of knowledge update by directly modifying the content of the explicit symbolic memory. However, even though FaE has successfully demonstrated knowledge modification when previously unseen facts are introduced, it exhibits poor performance when one tries to modify the facts that model encountered during the training due to contradictions between the implicit knowledge of underlying Transformer model and explicit content of the symbolic memory (Verga et al., 2020, §5.3). Thus, our work on modifying the implicit memory of Transformer models also has utility for the task of knowledge update in memory augmented Transformer models.

Generalization often requires memorization. In general, without specifically focusing on language models, Feldman (2020); Feldman & Zhang (2020) have demonstrated both theoretical results and empirical evidences to imply that close-to-optimal generalization requires memorization of labels for samples from the low-frequency sub-populations. This line of work is further supported by the recent efforts on adding k-NN component to language model to improve their generalization via memorization (Kassner & Schütze, 2020; Khandelwal et al., 2020). We believe that our work on modifying the implicit memories in Transformer models can be used to improve their generalization by boosting their factual knowledge in specific domains.

3 MODIFYING IMPLICIT FACTUAL KNOWLEDGE OF TRANSFORMER MODELS

In this section, we define a new knowledge modification task. We then present several approaches to solve this task with different computational costs. We focus on one approach based on constrained optimization that is highly effective and efficient.

3.1 MODIFICATION OF IMPLICIT KNOWLEDGE

We propose a new task of modifying specific pieces of knowledge in a Transformer that are stored implicitly in its weights. Specifically, we would like to change the model's weights in a way that a pre-selected subset of its knowledge is updated, while the rest of its knowledge is preserved. Such modifications can be challenging as each fact in implicit knowledge is stored non-locally across a large number of weights and each weight can affect a large number of implicitly memorized facts.

More formally, a pretrained Transformer based language model is defined by its parameters $\theta_0 \in \Theta$, which encodes a collection of factual knowledge $\mathcal{F}$ that the model has implicitly memorized. We would like to update a desired subset of facts $\mathcal{S} \subset \mathcal{F}$ to a new set of facts $\mathcal{M}$. At the end of the modification process, we should arrive at a model θ^{new} that implicitly stores the collection $\mathcal{F}' = \left\{\mathcal{F} \backslash \mathcal{S}\right\} \cup \mathcal{M}$. In particular, the new model θ^{new} not only stores the desired modified knowledge, but also retains the performance of θ_0 on the unmodified knowledge $\mathcal{F} \backslash \mathcal{S}$. For example, a Transformer model may have memorized 'Eliud Kipchoge' given the context 'The marathon world record is held by [MASK]'. When another athlete breaks this record, we will need to update this specific piece of knowledge while keeping most others intact.

3.2 BASELINE APPROACHES

In this subsection we discuss several obvious baseline approaches and setup our notation.

Retraining the model. A natural and reliable approach to solve the aforementioned knowledge modification task is to update all the training data to be consistent with the new fact, and retrain the model from scratch. This includes both the pretraining corpora and the finetuning dataset. This approach, however, is not practical for modifying a small amount of knowledge: updating the whole pretraining corpora is highly non-trivial and retraining the model from scratch is too expensive.

Finetuning on modified facts Another obvious approach is to fine tune the model on supporting evidences for the modified facts $\mathcal{D}_{\mathcal{M}}$. With θ_0 as the initialization, we solve:

$$\text{minimize}_{\theta \in \Theta} \quad \frac{1}{m} \sum_{x \in \mathcal{D}_{\mathcal{M}}} L(x; \theta), \tag{1}$$

where $m = |\mathcal{D}_{\mathcal{M}}|$ denotes the number of supporting evidences corresponding to the facts to be modified; and $L(x; \theta)$ denotes per-instance loss employed during the finetuning process. This approach indeed achieves almost perfect accuracy on the modified facts. But due to overfitting and catastrophic forgetting, the model's knowledge about the unmodified facts $\mathcal{F} \backslash \mathcal{S}$ can be destroyed, as we demonstrate in our experimental studies (cf. § 4.5.1).

Finetuning on a mixture of modified and unmodified facts This option appears when the model is first pretrained on a large corpus and then finetuned on a much smaller, knowledge intensive datasets, as is done frequently for question answering tasks. In this case the set of factual knowledge $\mathcal{F}$ we care about is contained in the finetuning dataset $\mathcal{D}_{\mathcal{F}}$. So we could finetune the model again on $\mathcal{D}_{\mathcal{F}'}$ – the updated finetuning dataset with supporting evidences for $\mathcal{F}' = \left\{\mathcal{F} \backslash \mathcal{S}\right\} \cup \mathcal{M}$. However this

is still too slow for many applications: its computational cost scales with the size of the finetuning dataset, even when only a handful of facts are modified.

3.3 CONSTRAINED FINETUNING ON SUPPORTING EVIDENCES FOR MODIFIED FACTS

In this paper, we explore a simpler yet more effective approach for knowledge modification, where we finetune the original model only on the modified facts $\mathcal{D}_M$ while using explicit constraints on the weights θ to achieve minimum interference with unmodified facts. With complexity that scales only with the number of modifications, this approach works surprisingly well in memorizing the new knowledge while preserving existing ones.

In the ideal scenario, instead of (1), the model should learn the new facts while keeping the loss small on unmodified facts:

$$\text{minimize}_{\theta\in\Theta} \quad \frac{1}{m}\sum_{x\in\mathcal{D}_{\mathcal{M}}} L(x;\theta) \quad \text{subject to} \quad \frac{1}{n}\sum_{x'\in\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}} \big(L(x';\theta) - L(x';\theta_0)\big) \leq \delta. \tag{2}$$

With a small positive δ we added a constraint on the model's performance on all $n = |\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}|$ training samples that provide supporting evidences for the unmodified facts $\mathcal{F} \setminus \mathcal{S}$.

However, it is expensive to enforce this constraint. So, we approximate the constraint by using local continuity of the loss around θ_0, to obtain the following program:

$$\text{minimize}_{\theta\in\Theta} \quad \frac{1}{m}\sum_{x\in\mathcal{D}_{\mathcal{M}}} L(x;\theta) \quad \text{subject to} \quad \|\theta - \theta_0\| \leq \delta, \tag{3}$$

where $\|\cdot\|$ denotes any suitable norm in the parameter space. We use ℓ_2 and ℓ_∞ norms in our experiments, where ℓ_∞ consistently leads to more stable results for knowledge modification.

Note that, if we use a very small δ, the model will not change much and the accuracy on the modified facts will be low while the accuracy on the unmodified facts will remain high. If δ is too large, we are essentially solving (2) which results in almost zero accuracy on unmodified facts. Therefore, δ is an important design parameter that needs to be chosen carefully.

Finetuning specific Transformer blocks. When finetuning large models on a small amount of data, a commonly used approach is to finetune only one layer while keeping others frozen. Note that with appropriately chosen δ to avoid overfitting, full-model finetuning and 1-layer finetuning will explore very different functional spaces and the later is not contained in the former.

We found modifying the initial and final layers of Transformers results in better adaptation to modified facts and better preservation of performance on unmodified facts (cf. § 4). This approach, interestingly, outperforms the case when the whole network is modified. This is partially consistent with Houlsby et al. (2019), who demonstrated that finetuning top layers of the BERT-base model are the best for certain tasks, except that we are also interested in retaining the memorization of the unmodified facts. Here, we found that sometimes initial layers give better results. For more work related to the roles of different layers on QA tasks see e.g. van Aken et al. (2019); Cao et al. (2020).

4 EXPERIMENTS

We now conduct a systematic and detailed experimental evaluation of different approaches to modify the knowledge implicitly stored in the Transformer model parameters. Similar to the prior work on assessing the amount and quality of memorization in language models (Petroni et al., 2019; Roberts et al., 2020), we rely on factual knowledge-based datasets. We create two new benchmarks for the knowledge modification tasks based on two such datasets (cf. § 4.1). The memorization is assessed by using cloze(-style) statements. Subsequently, we explore the performance of the baselines (cf. § 3.2) and the proposed constrained finetuning approach on the knowledge modification tasks for BERT (Devlin et al., 2018) and ALBERT (Lan et al., 2019) models. We conclude our experimental study by exploring the knowledge modification in the recently proposed FaE model (Verga et al., 2020) with explicit symbolic fact memory module.

4.1 DATASETS AND BENCHMARKS

We want to construct datasets of a collection of facts $\mathcal{F}$ along with modifications $\mathcal{M}$ for a subset of facts $\mathcal{S}$. We take two fact-based datasets, namely T-REx (Elsahar et al., 2018) and *Zero-shot Relation Extraction* (zsRE) (Levy et al., 2017), as the source of the original facts $\mathcal{F}$ (see Appendix A for details). These datasets contain a large number of facts (cf. Table 1), with each fact being supported by potentially multiple evidences (in the form of sentences or QA pairs). To modify m facts $\mathcal{S}$ from $\mathcal{F}$, we update the objects/answers in the supporting evidences for those facts. For replacing the object in an evidence, we sample a new object from the collection of objects that are connected to the same relation as the initial object, according to its frequency in the training set. For example, if the original supporting evidence appears in the form of a QA pair, with the question being *Which country was Charles Darwin born? [ANS]*, we modify the label for the [ANS] token into any object that appears as someone's birth place in the training set, other than *United Kingdom*. Note that we consistently modify all the training and test samples for this fact.

Dataset	**# question**	**# facts**
T-REx (training)	1,282,567	34,039
T-REx (test)	34,039	34,039
zsRE (training)	197,829	147,905
zsRE (test)	59,527	47,156

Table 1: Statistics of T-REx and zsRE datasets.

We develop two benchmarks for the knowledge modification task based on T-REx and zsRE. These benchmarks allow a model to be finetuned on a given set of facts by providing the supporting evidences, in which the object of the fact are masked out. The model is then asked to predict the masked out object and learns the fact via supervision. During the test-time, the memorization of a fact by the model is assess by presenting a cloze-type statement to the model. For example, to understand if the model knows the relationship between Hamlet and William Shakespeare, one can probe the model with either of the two cloze-type statements: *Hamlet is written by [MASK]* or *Who is the author of Hamlet? [MASK]*. If the model knows the corresponding fact, it should lead to *William Shakespeare* as the prediction for the [MASK] token. Here, it is important to note that the model may respond with the correct answer just by memorizing some spurious correlations in the pretraining or finetuning dataset. Thus, to assess if the model actually "knows" a fact, it is important to probe the model for the given fact in different formats. Both T-REx and zsRE datasets indeed provide different forms of questions about the same fact. To enable better comparisons with existing works on probing the implicit memorization in language models, we use the versions of T-REx and zsRE datasets from LAMA (Petroni et al., 2019) and KILT (Petroni et al., 2020) benchmarks, respectively.

4.2 Model architectures

We work with three Transformer based language models for our experimental study:

BERT (Devlin et al., 2018) We evaluate both the uncased BERT-Base and BERT-Large models without whole word mask training, as released by the official repository[1]. The two models have 12/24 Transformer blocks with 768/1024 hidden dimension and 110M/340M parameters respectively.

ALBERT (Lan et al., 2019) We only evaluate ALBERT-XXLarge model, which is the largest ALBERT model from Lan et al. (2019). It has a total of 235M parameters. The weights are shared in each transformer block, so the only option here is to finetune all its parameters on the modified facts.

FaE (Verga et al., 2020) FaE adds symbolic memories to the BERT-Base model. It inherits the entity memory module from EaE (Févry et al., 2020) and adopts an additional fact memory to enhance the representation of the facts. EaE alone has 367M parameters, comparable to the BERT-Large model.

4.3 Notations

Let's first introduce the candidates for the original model that we begin with. We use `PT` to refer to a pre-trained model, `FT` to refer to a model that have been finetuned on $\mathcal{D}_{\mathcal{F}}$, which denotes the supporting evidences for the facts $\mathcal{F}$ - that we expect the original model to implicitly memorize. Furthermore, `RI` refers to a randomly initialized model.

For knowledge modification, we explore the baselines discussed in § 3.2. Let `FTM` denote the models obtained by finetuning the original model on only $\mathcal{D}_{\mathcal{M}}$, i.e., the supporting evidences for the modified facts $\mathcal{M}$. Similarly, `FTA` denotes the models that result from finetuning the original model on $\mathcal{D}_{\mathcal{F}'}$, i.e., the supporting evidences for both modified and unmodified facts.

[1] https://github.com/google-research/bert.git

Finetuned layer	0		5		11	
	$\mathfrak{A}_{\mathcal{M}}$	$\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$	$\mathfrak{A}_{\mathcal{M}}$	$\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$	$\mathfrak{A}_{\mathcal{M}}$	$\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$
`RI + FTM`	19.38 (2.40)	0.63 (0.12)	21.25 (1.05)	0.33 (0.06)	20.00 (0.68)	0.53 (0.09)
`PT + FTM`	75.00 (3.19)	0.30 (0.03)	66.25 (2.40)	0.83 (0.05)	67.50 (1.12)	0.49 (0.03)
`PT+FT + FTM`	77.50 (2.40)	0.37 (0.02)	77.50 (1.37)	15.09 (1.94)	82.50 (2.27)	1.12 (0.25)

Table 2: Finetuning BERT-Base on the modified supporting evidences $\mathcal{D}_{\mathcal{M}}$ without constraints. We report the accuracy by averaging over 5 independent runs with standard deviation in parentheses. $\mathfrak{A}_{\mathcal{M}}$ is the accuracy on modified facts, and $\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$ is the accuracy on the unmodified facts.

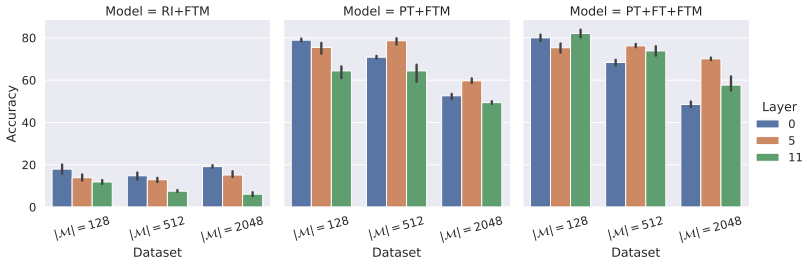

Figure 1: Mean and standard deviation of test accuracies after finetuning `RI`, `PT`, `PT+FT` models on different number of modified facts.

4.4 Performance measure

As discussed in § 4.1, we evaluate the implicit memorization of a model by probing its factual knowledge with cloze-type sentences, where the object/answer is masked out. Given a benchmark and a collection of facts $\mathcal{F}$, let $\mathcal{P}_{\mathcal{F}}$ denote the set of probes for $\mathcal{F}$ in the test set of the benchmark. We use $\mathfrak{A}_{\mathcal{F}}$ to denote the accuracy of the model on $\mathcal{F}$, which corresponds to the fraction of probes in $\mathcal{P}_{\mathcal{F}}$ for which the model predicts the correct object/answer.

As discussed in § 3.3, for constrained finetuning, we vary the parameter δ across a broad range and use a weighted sum of the accuracies on modified and unmodified facts as a metric to select the best δ. The sum is weighted by the inverse of the numbers of corresponding facts, which suitably focus on the performance on the relatively small number of modified facts.

4.5 Results

We now present the results for different baselines on the two benchmarks that we have introduced for the knowledge modification task. In particular, we show that in order to ensure that modifying the certain facts does not lead to degraded performance on the unmodified facts, it is important to perform fine-tuning on the supporting evidences for the modified fact in a constrained manner (cf. (3)). Furthermore, for constrained fine-tuning, training only specific layers leads to better performance on knowledge modification tasks as compared to training all the model parameters.

4.5.1 Fine-tuning on modified facts without constraints

For T-REx benchmark and BERT-Base, Table 2 presents the results for finetuning on only modified facts without any constraints, i.e., we employ (1) which is also equivalent to (3) with $\delta = \infty$. Note that these results are for a setting where we modify $|\mathcal{M}| = 32$ facts from T-REx benchmark. We present result for the settings with `RI`, `PT`, and `PT + FT` as the original model (cf. § 4.3). Note that the randomly initialized model `RT` has no language understanding to begin with. Thus, with limited finetuning data, `RT + FTM` model, i.e., the model obtained by finetuning `RT` model on the modified facts, exhibits poor accuracy on both the modified and the unmodified facts.

In contrast, both `PT + FTM` and `PT + FT + FTM` models results in non-trivial accuracy on the modified facts. However, as a result of performing unconstrained finetuning on only modified facts, the performance of these model significantly degrades on the unmodified facts. Note that this degradation is indeed a result of the knowledge modification process as `PT` and `PT + FT` models have accuracy of 28.85 and 50.50 on all the facts in T-REx to begin with.

Before we move to address the model degradation on the unmodified facts during the knowledge modification, we would like to notice that it is already clear from Table 2 that training different layers in a Transformer models lead to different outcome for the knowledge modification task, which depend on the state of the original model. We present additional results on the role of different layers for knowledge modification for different number of modified facts in Figure 1.

4.5.2 FINETUNING ON MODIFIED FACT WITH CONSTRAINTS

As observed from the results in § 4.5.1, unconstrained finetuning on the modified facts leads to significant overfitting on the modified facts. Note that this is the case even when we train a single layer of BERT-Base model. To prevent the overfitting during the modification process, we employ the finetuning on modified facts with ℓ_∞ constraints (cf. (3)). In what follows, we restrict ourselves to `PT + FT` models as original model as it leads to the best results for our knowledge modification task.

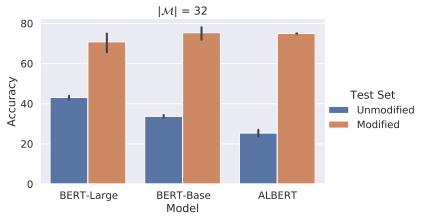

Figure 2: Performance of constrained finetuning of all layers for BERT-Large, BERT-Base, and ALBERT on T-REx.

Figure 2 describes the results for `PT + FT + FTM` models for three different model architectures and $|\mathcal{M}| = 32$ modified facts on T-REx benchmark. Note that in the constrained finetuning setting, we report the results for the best model according to the sum accuracy on modified and unmodified facts (cf. § 4.4). As evident from Figure 2, even when we train all model parameters, the constrained finetuning is able to prevent the overfitting observed in § 4.5.1 and ensure good performance on both modified and unmodified facts.

Next, we explore the performance on the knowledge modification task when we finetune individual layers of BERT-Base and BERT-Large. Note that a similar exploration is not possible for ALBERT as it employs parameter sharing across layers. Figure 3 shows the result on T-REx benchmark when we modify $|\mathcal{M}| = 32$ and 512 facts. Note that both initial and final layers play important role in knowledge modification. Also, the constrained finetuning of single layer ensures good accuracy for both modified and unmodified facts as long as we modify a small number of facts. However, as the number of modified facts increases, we see performance degradation, with the accuracy on unmodified fact experiencing a larger hit. In Figure 4, we observe the similar results with BERT-Base on the zsRE-based benchmark.

4.5.3 FINETUNING ON BOTH MODIFIED AND UNMODIFIED FACT WITH CONSTRAINTS

As we observe in § 4.5.1, with increasing number of modified facts, the accuracy on unmodified facts degrades quite sharply. One plausible reason for such degradation might be that, besides constraining the magnitude of the change in the model parameters, the modification process in § 4.5.1 does not take incorporate unmodified facts during the training. Thus, we explore another natural baseline from § 3.2 where we perform a constrained finetuning based on a mixture of modified and unmodified facts to attain the knowledge modification. In this approach during the knowledge modification, we finetune the relevant model parameters based on mini-batches that include supporting evidences for both modified and unmodified facts. We present the results for this approach on T-REx benchmark in the supplementary material (cf. Appendix B). Surprisingly, when we train the parameters for a single layer, this approach does help the accuracy of the unmodified facts. These results also allude

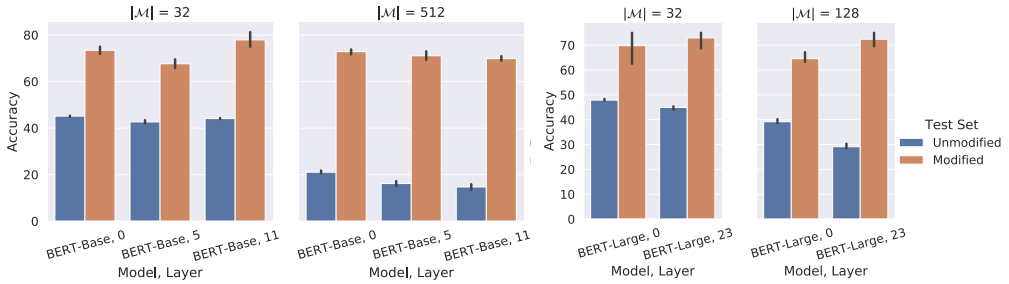

Figure 3: Performance of fact modification for BERT-Base and BERT-Large on the T-REx benchmark. We report the result for the best models obtained by varying δ. The results are averaged over 5 independent runs.

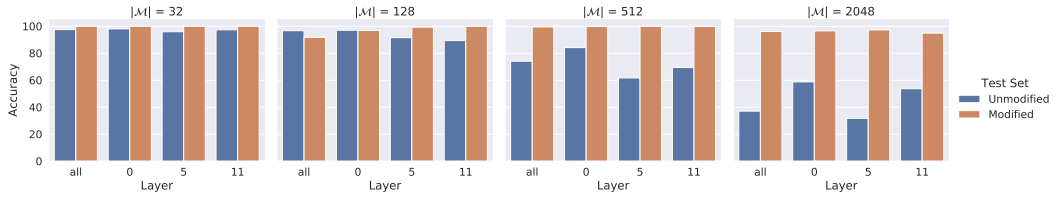

Figure 4: Performance for fact modification on the zsRE benchmark. From left to right, the columns show the test accuracy for modifying 32, 128, 512, and 2048 facts, respectively. In each column, we show the best accuracies of constrained finetuning the 0th, 5th and 11th, and all the transformer blocks of BERT-Base, which we achieve under different ℓ_∞ constraints. The results are averaged over 5 independent runs.

that, when $|\mathcal{M}|$ is large, the performance degradation on unmodified facts might be a byproduct of the limited capacity that finetuning a single layer approach provides.

4.6 Modifying symbolic memories in FaE model

Finetuned parameters	**None**	**Cls.**	**3 + Cls.**	**7 + Cls.**	**All**
$\mathfrak{A}_{\mathcal{M}}$	46.88	75.00	78.12	81.25	75.00
$\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$	60.38	57.38	45.22	41.06	53.37
$\Delta\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$	0.00	-3.00	-15.16	-19.32	-7.01

Table 3: Results on fine-tuning different layers of FaE on the $|\mathcal{M}| = 32$ modified facts of T-REx under a range of constraints. $\Delta\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$ is the drop in accuracy on unmodified facts. We report the results with $\mathfrak{A}_{\mathcal{M}}$ closest to 77.50%, the accuracy on the modified facts achieved by a BERT-Large model. By comparison, FaE does not have a significant advantage in terms of tradeoff between $\mathfrak{A}_{\mathcal{M}}$ and $\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$ when we require $\mathfrak{A}_{\mathcal{M}}$ to be high. Cls. refers to all the weights of FaE that are outside the Transformer model, 3 and 7 are the middle/last Transformer block of its second-stage Transformer encoder. None means no parameter is fine-tuned but the symbolic knowledge are modified.

An important advantage of the models with symbolic memory modules such as FaE (Verga et al., 2020) is that they could be easily updated by modifying the symbolic links. However, since these models rely on both the contextual representation and the symbolic links, inconsistency between the contextual representation and the fact memory can result in wrong prediction by the models. Our experimental results show that finetuning the parameters of FaE is necessary for it to obtain high accuracy on the modified facts, but a similar drop on the accuracy for unmodified facts $\mathcal{F} \setminus \mathcal{S}$ like BERT models is inevitable (cf. Table 3). For this set of experiments, we modify $|\mathcal{M}| = 32$ facts from T-REx dataset. After modifying the symbolic links stored in the database, FaE achieves 46.88% accuracy on the modified facts, which is higher than the 30% reported by Verga et al. (2020), and its accuracy on unmodified facts stays almost unchanged at 60.38%. We find finetuning only the layers that directly map symbolic memory to the predictions result in the best trade-off (denoted as Cls. in Table 3). After the finetuning, FaE reaches 75.00% accuracy on the modified facts with a drop of 3.00% on the unmodified facts; and reaches 85.00% accuracy on the modified facts with a drop of 6.5% on the unmodified facts. By comparison, BERT-Large is able to achieve an accuracy of 77.50% on modified facts with a drop of less than 4.00% on unmodified facts.

5 Conclusion

We proposed a novel task of modifying the factual information implicitly stored in the parameters of a Transformer model. For the knowledge modification task, we introduced two benchmarks based on T-REx and zsRE datasets. We further established the effectiveness of constrained finetuing approach on the knowledge modification task without adversely affecting the model's performance on unmodified factual knowledge.

While we briefly explored the issue of knowledge modification in the context of the models with symbolic fact memory, a comprehensive exploration of mechanisms to achieve reliable and consistent modification of both implicit and explicit knowledge of such models is an interesting research direction. Another natural future work would be to understand implications of modifying facts on multi-hop logical inference, i.e. whether generalization aspect can interact well with modified facts.

REFERENCES

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of "bias" in NLP. *arXiv preprint arXiv:2005.14050*, 2020.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4349–4357. Curran Associates, Inc., 2016.

Shikha Bordia and Samuel Bowman. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 7–15, 2019.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Nicola De Cao, Michael Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. *arXiv preprint arXiv:2004.14992*, 2020.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*, pp. 267–284, 2019.

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

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W Cohen. Differentiable reasoning over a virtual knowledge base. In *International Conference on Learning Representations*, 2019.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, 2018.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959, 2020.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *arXiv preprint arXiv:2008.03703*, 2020.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. Entities as experts: Sparse memory access with entity supervision. *arXiv preprint arXiv:2004.07202*, 2020.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799, 2019.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *arXiv preprint arXiv:1911.12543*, 2020.

Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. pp. 22–es, USA, 2004. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. Bert-knn: Adding a knn search component to pretrained language models for better qa. *arXiv preprint arXiv:2005.00766*, 2020.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*, 2017.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

Pandu Nayak. Understanding searches better than ever before, 2019. URL `https://blog.google/products/search/search-language-understanding-bert/`.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, et al. KILT: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pp. 9689–9701. Curran Associates, Inc., 2019.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*, 2020.

Dan Roth and Wen-tau Yih. Probabilistic reasoning for entity & relation recognition. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2440–2448. Curran Associates, Inc., 2015.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. ERNIE: enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

Mihai Surdeanu and Heng Ji. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. 2014.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. How does BERT answer questions? A layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1823–1832, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008. Curran Associates, Inc., 2017.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *arXiv preprint arXiv:2007.00849*, 2020.

Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

J. Zelle and R. Mooney. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI, Vol. 2*, 1996.

Luke S Zettlemoyer and Michael Collins. Learning to map sentences to logical form: structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 658–666, 2005.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 19–27, 2015.

Supplementary material for "Modifying Memories in Transformer Models"

A Dataset details

T-REx dataset. We consider the 41 Wikipedia relations with a total of 34039 facts from Petroni et al. (2019). All the object labels in the dataset can be represented by a single token. In this version of dataset, each fact has at least one supporting sentence from Wikipedia with the object annotated, plus a template for each relation to construct an additional cloze-type question. For example, the annotated supporting sentence in the training set may look like *Charles Darwin, born February 12, 1809, The Mount House, [MASK], United Kingdom, was an English naturalist, geologist and biologist*, with the template question in the test set being *Charles Darwin was born in [MASK]* and the answer/object being *Shrewsbury*. When doing modification, we may modify *Shrewsbury* into *Middlebury*. We use all of the supporting sentences with sequence length no larger than 512 to construct the training set.

Zero-shot Relation Extraction (zsRE) dataset. zsRE is a relation extraction dataset originally formulated as a reading comprehension problem to match each question with a sentence from Wikipedia (Levy et al., 2017). We take the reformulated version of zsRE from KILT (Petroni et al., 2020), which includes multiple template questions for most of the subject-relation pairs. For example, for the relation triple (Charles Darwin, birth palce, Shrewsbury), it may contain (at least) the following two templates: *In which city was Charles Darwin born? [MASK]* and *Where is the birth place of Charles Darwin? [MASK]*, where answer/object should be *Shrewsbury*. When doing modification, we may modify *Shrewsbury* into *Middlebury*.

Since the relations in different splits from KILT are disjoint, and we need to validate the memorization of a certain fact with a different question, we create the modification benchmark from only its training set. For each relation, we randomly put two of its questions into the test set if it has more than 3 questions, preserve the question in the training set if it has only one question, and randomly put one question into the test set otherwise. When applying the uncased BERT tokenizer, we limit the length of the input sequence to be no longer than 512 and the answer's length to be no longer than 20. We see the prediction as correct only when all the predicticted tokens match the label.

B Finetuning on a mixture of modified and unmodified facts

Here, we explore the constrained finetuning approach for the knowledge modification task on T-REx benchmark. Recall that $\mathcal{D}_{\mathcal{M}}$ and $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$ denote the supporting evidences for the modified facts $\mathcal{M}$ and the unmodified facts $\mathcal{F}\setminus\mathcal{S}$, respectively. Table 4 presents the result for the setting where $|\mathcal{M}| = 512$.

Finetuned layer	0		5		11	
	w/ $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$	w/o $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$	w/ $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$	w/o $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$	w/ $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$	w/o $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$
$\mathfrak{A}_{\mathcal{M}}$	73.31 (0.74)	72.85 (0.51)	76.04 (0.65)	71.09 (0.88)	70.64 (0.68)	69.86 (0.46)
$\mathfrak{A}_{\mathcal{F}\setminus\mathcal{S}}$	18.51 (0.94)	21.06 (0.31)	8.73 (0.41)	16.19 (0.50)	15.30 (0.50)	14.71 (0.60)

Table 4: Comparing the results of fine-tuning with constraints on the supporting evidence of $|\mathcal{M}| = 512$ modified facts with and without the supporting evidences for the unmodified facts in every mini-batch (T-REx benchmark). We report the results after averaging over 5 independent runs with standard deviation in parentheses.

We train the model for 10 epochs with a batch size of 128, which results in a total of 112 iterations per step on $\mathcal{D}_{\mathcal{M}}$. In each iteration, if using the unmodified training samples, we additionally sample 128 samples from $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$, and compute the gradient of the averaged loss on the 256 samples. This effectively uses around 10% of the samples of $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$. Such a mixture of modified and unmodified supporting evidence in every iteration is supposed to make the accuracy high on $\mathcal{M}$, while also preserving the accuracy on $\mathcal{F} \setminus \mathcal{S}$. However, as we observe in this table, there is no significant improvement by using such mixed batch. Though we are using 50% of unmodified training sample in each iteration, the optimizer repeatedly loops over $\mathcal{D}_{\mathcal{M}}$, which effectively makes the model 10

times as more biased towards the expected loss $\mathcal{D}_{\mathcal{M}}$ (as we train 10 epochs) than $\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}$. Such a bias can be alleviated by increasing the ratio of unmodified data in each batch, but would probably require more iterations and computation to achieve the same level of accuracy on $\mathcal{D}_{\mathcal{M}}$, even if it is able to improve the accuracy on unmodified data.

C THE SMALL MODIFICATION LIMIT

In this section we theoretically discuss the small modification limit of the loss constraint in (2), reproduced here:

$$\text{minimize}_{\theta\in\Theta} \quad \frac{1}{m}\sum_{x\in\mathcal{D}_{\mathcal{M}}} L(x;\theta) \quad \text{subject to} \quad \frac{1}{n}\sum_{x'\in\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}} \big(L(x';\theta) - L(x';\theta_0)\big) \leq \delta. \tag{4}$$

It is expensive to evaluate the constraint in over the entire $\mathcal{D}_{\mathcal{F}'}$. But in the limit where only a small number of facts are modified and the changes to the weights are small, the constraint simplifies to:

$$\sum_{ij} \Delta\theta_i \Delta\theta_j \frac{1}{2n}\Big(\frac{\partial}{\partial\theta_i}\frac{\partial}{\partial\theta_j}\sum_{x'\in\mathcal{D}_{\mathcal{F}\setminus\mathcal{S}}} L(x';\theta_0)\Big) + \mathcal{O}(\Delta\theta^3) \leq \delta, \tag{5}$$

where $\Delta\theta \equiv \theta - \theta_0$. Here because the number of modified facts are small, we can assume that we are still at the minimum of the loss function with respect to the unmodified facts. Thus the linear term in $\Delta\theta$ vanishes and the second order term should dominate.

If we use cross-entropy loss then the quantity in the bracket is the Fisher metric. Even though the Fisher metric only needs to be computed once, it is still difficult to parallelize across samples and thus time-consuming. We experimented with an approximate fisher information computed with batchsize 128 and found that it did not outperform the ℓ_∞ norm when using (3). We leave the detailed exploration of Fisher metric to future work.