MUSCLE: A Model Update Strategy for Compatible LLM Evolution

Jessica Echterhoff[*] **Fartash Faghri** **Raviteja Vemulapalli**
UC San Diego   Apple   Apple

Ting-Yao Hu **Chun-Liang Li** **Oncel Tuzel** **Hadi Pouransari**[*]
Apple   Apple   Apple   Apple

Abstract

Large Language Models (LLMs) are regularly updated to enhance performance, typically through changes in data or architecture. Within the update process, developers often prioritize improving overall performance metrics, paying less attention to maintaining compatibility with earlier model versions. Instance-level degradation (*instance regression*) of performance from one model version to the next can interfere with a user's mental model (Bansal et al., 2019) of the capabilities of a particular language model. Users having to adapt their mental model with every update can lead to dissatisfaction, especially when the new model has degraded compared to a prior version for a known use case (*model update regression*). We find that when pretrained LLM base models are updated, fine-tuned user-facing downstream task adapters experience negative flips – previously correct instances are now predicted incorrectly. We observe model update regression between different model versions on a diverse set of tasks and models, even when the downstream task training procedures remain identical. We argue for the importance of maintaining *model update compatibility* during updates, and present evaluation metrics designed specifically for generative tasks, while also being applicable to discriminative tasks. We propose a training strategy to minimize the extent of instance regression in model updates, involving training of a compatibility adapter that can enhance task fine-tuned language models. We show negative flips reduce by up to 40% e.g. when updating Llama 1 to Llama 2 with our proposed method.

1 Introduction

Large Language Models (LLMs) are often pretrained on large-scale corpora to obtain a base model with general world knowledge. These base

---
[*] Research conducted during an internship at Apple. Corresponding authors: jechterh@ucsd.edu, mpouransari@apple.com
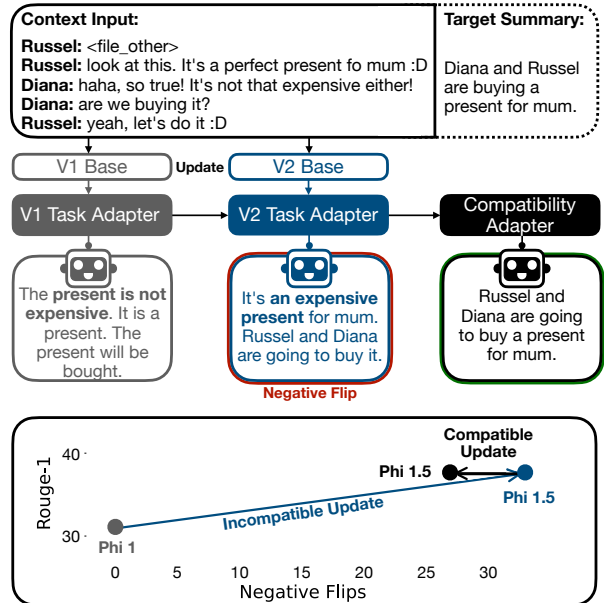
Figure 1: A real example of a model update that introduces instance regression (negative flip, where a previously correct prediction becomes incorrect) (top). With our model update strategy using a compatibility adapter approach, we enhance model update compatibility to the previous model while maintaining the overall performance gain (e.g. measured by the ROUGE-1 score for the summarization task) of the model update (bottom).

models are typically evaluated using a suite of benchmarks that mostly focus on zero/few-shot performance and in-context learning capabilities. Training these models is expensive, and only a few organizations have access to the resources needed. Hence, to enable various user-facing applications such as summarization, chatbots, code assistants, and question-answering, practitioners often adapt pre-trained base models by training task-specific parameter-efficient adapters using downstream task datasets.

Several scenarios drive updates to the base model, e.g. improved training strategy, advances in LLM architectures or increasing model context

length (Touvron et al., 2023a), the availability of additional or higher quality datasets (Gunasekar et al., 2023), the expansion of model vocabulary (e.g., to support multilingual or multimodal models), or simply training for a longer period (Biderman et al., 2023).

When a base model is updated, all associated task adapters need to be retrained/updated to have meaningful downstream models. Hence, in the rest of the paper, we use the term *model update* to refer to an update to a downstream task model, which includes updating the base model and retraining the task adapter. When a model is updated, we evaluate *model update compatibility* with different metrics by four quadrants shown in Fig. 2. The new model could produce a worse prediction (negative flip (quadrant 4)) for many samples (Yan et al., 2021) even when it has a better overall performance when compared to the previous model. A real example of instance regression measured as a negative flip is shown in Fig. 1 for a dialogue summarization task. This kind of regression can confuse the user and impair their satisfaction (Sakai, 2022). We denote the aggregated overall regression for all individual instances as *model update regression* (Table 1). Regression testing has become increasingly important for the evolving use of LLMs accessed via APIs (Ma et al., 2023). When updating models, practitioners typically focus on increasing positive flips (quadrant 2) while avoiding negative flips (quadrant 4) (Cai et al., 2022; Sakai, 2022; Yan et al., 2021; Li et al., 2023b; Schumann et al., 2023). However, they neglect prediction inconsistencies in the scenarios where both model versions are incorrect (quadrant 3) or already correct, but slightly different (quadrant 1). For example, Träuble et al. (2021) assumes the cost of flips from one incorrect class to another incorrect class to be zero. We argue that there is value in evaluating consistency when both models are wrong. A user may have developed coping strategies on how to interact with a model when it is incorrect; therefore, inconsistencies in mistakes can lead to user dissatisfaction.

Previous works have mainly addressed the model update regression challenge for classification tasks (Cai et al., 2022; Sakai, 2022; Yan et al., 2021; Li et al., 2023b; Schumann et al., 2023). In this work, we systematically study the problem of model update compatibility using discriminative and generative downstream tasks and different base models.

Figure 2: Four possibilities arise for each sample when a model is updated. Quadrants 2 and 4 show positive and negative flips, respectively. Quadrant 3 corresponds to instances where both models are incorrect. Encouraging similarity between the old and new models in this case (i.e., making the same mistakes) results in a more seamless model update from the user's perspective.

Following is a summary of our contributions:

- We formulate the compatibility problem when updating LLMs. We focus on common setups where base LLMs undergo updates and evaluate model compatibility and performance via parameter-efficient, task-specific fine-tuning, as these models are deployed for user interaction.

- We extend the notion of model compatibility from discriminative to generative tasks, and propose compatibility metrics that consider similarities in model behavior after an update, going beyond the negative flip rate metric.

- We investigate model compatibility for different update scenarios using open-weight models and find significant model update regression across various tasks.

- We propose learning a compatibility adapter to align model versions and minimize model update regression. We demonstrate up to 40% reduction in negative flip rate (e.g. for Llama 1 to Llama 2 update in language understanding) and reduced model inconsistency for downstream tasks such as summarization, math reasoning, and commonsense question-answering.

2 Related Work

2.1 Measuring Model Update Regression

Classification Yan et al. (2021) introduce negative flip rate (NFR) to evaluate model compatibility for classification tasks. NFR calculates the fraction of instances that were previously correct but

Instance Regression	An instance i is predicted correctly by an old model $\mathcal{M}_{v1}$, but incorrectly by a new model $\mathcal{M}_{v2}$.
Model Update Regression	Aggregation of all instances correctly predicted by an old model $\mathcal{M}_{v1}$, but incorrectly by an updated model version $\mathcal{M}_{v2}$.
Model Update Compatibility	Measures the degree to which predictions made by an updated model $\mathcal{M}_{v2}$ remain consistent with the predictions of the previous model version $\mathcal{M}_{v1}$, specifically on instances where $\mathcal{M}_{v1}$ was correct. Quantifies the alignment or compatibility between model versions across updates.

Table 1: Definitions of Instance Regression, Model Update Regression, and Model Update Compatibility.

are now incorrect with the new model. A similar statistic (Sakai, 2022), backward trust compatibility (BTC) (Srivastava et al., 2020), measures the ratio of instances that the new model predicts correctly among all instances the old model predicts correctly. Matsuno and Sakuma (2023) propose backward compatibility with a conditional distribution, which computes the ratio at which the accuracy of the conditional distribution of the new model is equal to or higher than that of the old model. Cai et al. (2022) introduce the negative flip impact for graph NLP tasks, taking into account the negative flips and the overall error rate of the model. These aforementioned metrics are limited to the evaluation of discriminative classification tasks.

2.2 Reducing Model Update Regression

Model Ensembles Prior work found that model ensembles reduce model update regression. This can be attributed to the reduction of variance by ensembling, as every single model may capture the training data from a distinct aspect (Yan et al., 2021; Xie et al., 2021). Extensions on this line of work include choosing the most centric model from an ensemble (Xie et al., 2021), aligning two models' uncertainties (Li et al., 2023b), or using gating mechanisms (Lai et al., 2023). Previous work has also used model parts from an old model to infuse into the new one (Ran et al., 2023), with the limitation of both models being required at inference time. For limited use cases, Qin et al. (2023) have shown to re-use previously learned adapters when purely a data update was performed. All these methods either introduce a larger memory footprint by re-using old model parts or are limited to (same-domain) data-updates and same models.

Knowledge Distillation Originally proposed for model compression (Buciluǎ et al., 2006), a (smaller) student model is trained to mimic a (larger) teacher model. By treating the old model as the teacher and the new model as the student, knowledge distillation has been shown to reduce

model update regression in vision and language discriminative tasks (Yan et al., 2021; Xie et al., 2021; Schumann et al., 2023; Jaeckle et al., 2023; Zhang et al., 2021; Shen et al., 2020; Ramanujan et al., 2022). Shen et al. (2020) propose a distillation-based influence loss to align new model representations with those of the old model. Similarly, Ramanujan et al. (2022); Jaeckle et al. (2023) apply distillation after a learned transformation module. Schumann et al. (2023) propose weight interpolation between the old and the new model, Zhao et al. (2022) suggest matching old and new model distributions, Yan et al. (2021) distill from an ensemble, and Caciolai et al. (2023) recommend using focal distillation. However, none of these approaches evaluate their approaches on generative tasks or they require both the old and new models to be in memory at inference time.

3 Problem Formulation

While existing methods provide valuable approaches to mitigating model update regression, they predominantly focus on discriminative classification tasks and often require additional memory at inference time. We propose a flexible approach to updating models without sacrificing performance or compatibility across downstream tasks.

Setup We follow the common setup of finetuning a pre-trained base LLM to multiple downstream tasks with task-specific LoRA adapters (Hu et al., 2021). Let $\mathcal{M}_i^{\text{base}}$ denote the ith version of a base LLM with parameters θ_i. We adapt $\mathcal{M}_i^{\text{base}}$ to a downstream task $\mathcal{T}$ using an adapter $\mathcal{A}_i^{\mathcal{T}}$ to obtain a downstream model $\mathcal{M}_i^{\mathcal{T}}$ with weights $\theta_i^{\mathcal{T}} = \theta_i + \Delta_i^{\mathcal{T}}$, where $\Delta_i^{\mathcal{T}}$ denotes the weights of the task-specific adapter $\mathcal{A}_i^{\mathcal{T}}$ learned using the training data corresponding to task $\mathcal{T}$. When the base model is updated from $\mathcal{M}_{v1}^{\text{base}}$ to $\mathcal{M}_{v2}^{\text{base}}$, the task-specific adapters are re-trained for each downstream task. Hereafter, for simplicity of notation, we use $\mathcal{M}_{v1}$ and $\mathcal{M}_{v2}$ to refer to the task-adapted models $\mathcal{M}_{v1}^{\mathcal{T}}$ and $\mathcal{M}_{v2}^{\mathcal{T}}$, respectively, and explicitly mention the task $\mathcal{T}$ when needed.

3.1 Backward Compatibility Metrics

A backward compatibility metric outputs a compatibility score based on two models, $\mathcal{M}_{v1}$ to $\mathcal{M}_{v2}$. Yan et al. (2021) propose negative flip rate (NFR) to measure regression in classification models over a dataset $\{x_i, y_i\}_{i=1}^{N}$, where y_i is the ground truth class for input x_i for a particular task $\mathcal{T}$:

$$\text{NF}(x_i) \triangleq [\mathcal{M}_{v1}(x_i) = y_i] \wedge [\mathcal{M}_{v2}(x_i) \neq y_i]$$

$$\text{NFR} \triangleq \frac{1}{N} \sum_i^N \mathbf{1}[\text{NF}(x_i)]$$

Here $\mathbf{1}$ denotes the indicator function. This notion of regression is partly applicable to autoregressively trained tasks. LLM benchmarks (Gao et al., 2023) calculate the likelihood of every possible choice in a multiple-choice question and choose the response with the highest likelihood to calculate a *likelihood based accuracy*. This evaluation can indicate model update regression similar to classification for tasks when multiple choices are available (Zellers et al., 2019; Wang et al., 2019; Welbl et al., 2017).

Unobserved Inconsistencies Other inconsistencies arise when the old model predicts class A, the new model class B, and the ground truth is class C. Similarly, if there are multiple ground truth options, a flip could occur within the ground truth options, as in quadrant 1 in Fig. 2. These inconsistencies are not captured by positive or negative flips. We argue that if we cannot produce a better prediction, we should at least stay consistent with the user's expectations and propose extended negative flips metrics for this use case.

Continuous Metrics In generative tasks for language models, we do not necessarily have multiple choices for which we can predict if an instance regressed or not. Hence a metric incorporating multiple choices and then calculating negative flips is not applicable for continuous evaluation metrics. Typical continuous metrics in generative language tasks are ROUGE (Lin, 2004) or BERT (Zhang et al., 2019) scores. As regular negative flips are incapable of capturing these nuances, we require a new metric to evaluate compatibility for generative tasks like summarization.

3.2 Extended Evaluation Metrics

We propose a suite of metrics that evaluate model update compatibility on a fine-grained basis specifically for generative tasks (e.g summarization).

Accounting for Flips when Both Models are Incorrect To capture inconsistencies for instances where both old and new models are incorrect, we adapt the negative flip rate as follows when multiple choice options are available:

$$\text{NF}_{mc}(x_i) \triangleq [\mathcal{M}_{v2}(x_i) \neq y_i] \wedge$$
$$[\mathcal{M}_{v1}(x_i) \neq \mathcal{M}_{v2}(x_i)]$$

$$\text{NFR}_{mc} \triangleq \frac{1}{N} \sum_i^N \mathbf{1}[\text{NF}_{mc}(x_i)]$$

This includes the possibility that neither of the models gives the correct answer, but a change in behavior occurs that can confuse a user. Similarly, for multi-label tasks, this notion can account for ground-truths that may have flipped during a model update when multiple ground truth options exist.

Smooth Compatibility Metrics To add continuous metrics for generative tasks, we evaluate the *expected* model update regression. Our general framework aims to be independent of the actual similarity metric, such that it can be chosen in accordance with any respective task of interest. For example, translation tasks might require BLEU (Papineni et al., 2002), but summarization ROUGE (Lin, 2004) evaluation. Additionally, we want to measure a notion of *performance gain*, when both models are correct, but one might still be better than the other.

Given a similarity metric S, and model outputs $\mathcal{M}_{v1}(x_i)$ and $\mathcal{M}_{v2}(x_i)$ for an input x_i, the difference for a model update for a particular test instance i is

$$D(x_i) \triangleq S(\mathcal{M}_{v2}(x_i), y_i) - S(\mathcal{M}_{v1}(x_i), y_i)$$

acting as an indication of the distance between the two model outputs with respect to the grand truth. This per instance indication of distance enables us to classify which model update compatibility quadrant an instance falls into. In practice, similarity metrics could be BERT Score (Zhang et al., 2019), ROUGE Score (Lin, 2004), BLEU Score (Papineni et al., 2002), or model-as-a-judge metrics (Huang et al., 2024) depending on the use case and task.

To get a metric for generative tasks that is similar to the positive and negative flip rate in classification tasks, we observe the distribution of instances with positive gain or negative regression:

$$\widetilde{\text{PFR}} \triangleq \frac{1}{N} \sum_i^N \mathbf{1}[D(x_i) > 0]$$
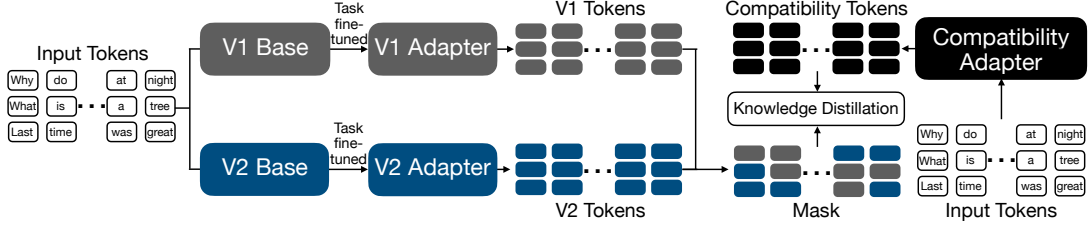
Figure 3: When updating a model, regression on individual tokens and instances can arise. We use a masked approach to select tokens to be aligned with knowledge distillation either with the old version to remain consistent or with the new task model to increase performance.

$$\widetilde{\text{NFR}} \triangleq \frac{1}{N} \sum_i^N \mathbf{1}[D(x_i) < 0]$$

To observe the magnitude of model update regression and gain, we can observe the expectation:

$$m_g \triangleq \frac{1}{N \cdot \widetilde{\text{PFR}}} \sum_i^N D(x_i) \, \mathbf{1}[D(x_i) > 0]$$

$$m_r \triangleq \frac{1}{N \cdot \widetilde{\text{NFR}}} \sum_i^N |D(x_i)| \, \mathbf{1}[D(x_i) < 0]$$

These give an indication of the average magnitude of change in a similarity metric when gain or regression occurs.

4 Knowledge Transfer

Now that we have metrics to indicate model update regression, we propose a knowledge distillation approach to minimize this regression for the task-specific models $\mathcal{M}_{v1}$ and $\mathcal{M}_{v2}$. Typically, knowledge distillation minimizes the KL divergence between the soft targets $\sigma(z_t)$ and $\sigma(z_s)$, where z_s and z_t are logits predicted by student and teacher models, respectively.

$$\mathcal{L}_{KL} = \frac{1}{n} \sum_{i=1}^n KL(\sigma(z_{t,i}/T) \| \sigma(z_{s,i}/T))$$

, i denotes the i'th token, n is the total number of tokens available for training, T is temperature parameter, and σ denotes softmax. Most knowledge distillation works consider the distillation from a trained teacher to an untrained student (Tian et al., 2022; Rajasegaran et al., 2020). Recent work (Roth et al., 2024) tackles the goal of knowledge transfer between pre-trained student and teacher models while retaining student knowledge gained a priori, and shows that standard knowledge distillation between pre-trained models struggles to transfer

knowledge without performance drops. Complementary to this work focusing on performance and maintaining prior knowledge, we tackle compatibility with prior models through knowledge transfer.

4.1 Model Update Strategy for Compatible LLM Evolution (MUSCLE)

When the base model is updated, we train a task-specific fine-tuned model, $\mathcal{M}_{v2}^C$, that has the accuracy benefits of $\mathcal{M}_{v2}$, but with most compatibility with $\mathcal{M}_{v1}$. We obtain $\mathcal{M}_{v2}^C$ by training a *compatibility adapter* applied to the base model $\mathcal{M}_{v2}^{\text{base}}$. We use knowledge from task-specific fine-tuned models $\mathcal{M}_{v1}$ and $\mathcal{M}_{v2}$ when training $\mathcal{M}_{v2}^C$. $\mathcal{M}_{v2}$ typically has increased prediction capabilities over $\mathcal{M}_{v1}$ (due to improvements in the base model), but $\mathcal{M}_{v1}$ has information on already correctly predicted tokens or instances that we want to minimize degradation towards.

We initialize the compatibility adapter with the task-specific adapter of $\mathcal{M}_{v2}$, and further fine-tune it (using the task training dataset) by aligning the next token prediction to either $\mathcal{M}_{v1}$ or $\mathcal{M}_{v2}$. We define masking (for individual tokens of a training sequence) following a simple heuristic depending on whether $\mathcal{M}_{v2}^C$ (the adapter being trained) predicts the correct tokens or not. If it does, we align to $\mathcal{M}_{v2}$ logits, otherwise, we align to $\mathcal{M}_{v1}$. The fine-tuning process of $\mathcal{M}_{v2}^C$ is depicted in Fig. 3. The fine-tuning loss to train the compatibility adapter, $\mathcal{L}_{comp}^m$, is defined below:

$$m_i = \mathbf{1}[\arg\max \sigma(z_{\mathcal{M}_{v2}^C, i}) \neq y_i]$$

$$a_{\mathcal{M}_{v1}} = KL(\sigma(z_{\mathcal{M}_{v1},i}/T) \| \sigma(z_{\mathcal{M}_{v2}^C,i}/T))$$

$$a_{\mathcal{M}_{v2}} = KL(\sigma(z_{\mathcal{M}_{v2},i}/T) \| \sigma(z_{\mathcal{M}_{v2}^C,i}/T))$$

$$\mathcal{L}_{comp}^m = \frac{1}{n} \sum_{i=1}^n m_i \cdot a_{\mathcal{M}_{v1}} + (1 - m_i) \cdot a_{\mathcal{M}_{v2}}$$

(1)

$\mathcal{M}_{v1}^{\text{base}}$	$\mathcal{M}_{v2}^{\text{base}}$	Update
Phi 1	Phi 1.5	Synthetic data, data selection (Gunasekar et al., 2023; Li et al., 2023a)
Phi 1.5	Phi 2	Synthetic data, data selection, $\uparrow$ parameters (Javaheripi et al., 2023)
Llama 1	Llama 2	More data, $\uparrow$ context size (Touvron et al., 2023a,b)
Vicuna 1.3	Vicuna 1.5	Llama 1 $\rightarrow$ Llama 2, Instruction fine-tuning (Zheng et al., 2024)

Table 2: We select a suite of models with varying update scenarios and parameter sizes and evaluate them on different LLM benchmark tasks. We select Llama and Vicuna models with 7B parameters.

Dataset	Task	Metric
HellaSwag	Language Understanding (Multiple-Choice)	Log-Likelihood Accuracy
PIQA	Commonsense Reasoning (Binary-Choice)	Log-Likelihood Accuracy
GSM8k	Grade School Math (Exact Match)	Exact Match Accuracy
SAMsum	Dialogue Summarization	ROUGE-1

Table 3: We tackle compatible model updates for different downstream tasks and datasets, including multiple-choice and generative tasks (Zellers et al., 2019; Bisk et al., 2019; Cobbe et al., 2021; Gliwa et al., 2019).

When evaluating, we denote NFR as negative flip rate between $\mathcal{M}_{v1}$ $\mathcal{M}_{v2}$, NFR$_c$ as the observed negative flip rate between $\mathcal{M}_{v1}$ and our compatibility model $\mathcal{M}_{v2}^C$, and ΔNFR$_c$ = NFR$_c$ − NFR.

5 Experimental Setup

5.1 Model Update Assumptions

To analyze the impact of model updates we consider parameter-efficient fine-tuned models using Low-Rank Adapters (LoRA) (Hu et al., 2021). Compared to previous work on continuous learning and model updates (Qin et al., 2022, 2023), we do not limit model updates to be produced by only data updates, but consider different kinds of updates shown in Table 2. We include updates due to data, increased parameters, or different training strategies. We include a wide range of downstream tasks to evaluate model compatibility, including *generative tasks* as summarized in Table 2. For all tasks, we learn the LoRA adapter autoregressively (next-token prediction).

5.2 Task Adapter Training

For each task and each old model $\mathcal{M}_{v1}$ and new model $\mathcal{M}_{v2}$, we train a LoRA adapter on all linear layers with $r = 128$ and $\alpha = 256$. We use a 0.8/0.2 training/validation split for 10 epochs. We choose the model by cross-entropy validation loss. More information on hyperparameters is shown in Table 8.

5.3 Compatibility Adapter Training

We keep all hyper-parameters the same for task adapter training, and train compatibility adapter with the $\mathcal{L}_{comp}^m$ loss defined in Eq. (1). We analyze the statistical significance of the results on a subset of compatibility adapter training for Phi 1 to Phi 1.5 updates on PIQA using 3 random seeds. We observe a standard deviation of accuracy of 0.0012.

We compare $\mathcal{L}_{comp}^m$ with different masking strategies m in the ablation studies in Section 6.5, but find that the version introduced in Section 4.1 works best for most model updates. For model updates with large performance gaps between $\mathcal{M}_{v1}$ and $\mathcal{M}_{v2}$ we find that an auxiliary cross-entropy loss enhances the stability of training for PIQA and SAMSum from Phi 1 to Phi 1.5, and all Phi updates in GSM8k and Hellaswag (more in Appendix A.4).

5.4 Similarity Metrics

We evaluate multiple-choice tasks with classification-like approaches by choosing the maximum log-likelihood of the possible answers (Gao et al., 2023). For math tasks, we use exact-match accuracy of the final calculated result. For summarization, we cannot evaluate in a classification-like manner. We use ROUGE-1 (Lin, 2004), given that we do not observe relative ranking differences for different ROUGE-n.

6 Results

6.1 Negative Flips Occur in Model Updates

Fig. 4 shows that significant negative flips (up to more than 60%) exist for a variety of base model update scenarios and downstream tasks. We observe negative flips in updates within one model family (e.g. Llama/Vicuna, and Phi models). We find more negative flips for model updates with a smaller delta in performance gain. For generative tasks like SAMsum dialogue summarization, we observe a large number of negative flips as continuous metrics are more sensitive to small changes when updated.

	$\mathcal{M}_{v1}^{\text{base}}$	$\mathcal{M}_{v2}^{\text{base}}$	$\text{acc}_{\mathcal{M}_{v1}}$	$\text{acc}_{\mathcal{M}_{v2}}$	acc_c	$\Delta\text{acc}_c \uparrow$	NFR	$\Delta\text{NFR}_c \downarrow$	$\Delta_\%\text{NFR}_c \downarrow$
HellaSwag	Llama 1	Llama 2	72.74	72.91	79.53	6.62	10.27	-4.17	**-40.60**
	Phi 1.5	Phi 2	69.76	78.02	77.86	-0.16	3.09	0.24	7.77
	Phi 1	Phi 1.5	32.38	69.76	69.86	0.10	2.47	-0.13	**-5.26**
	Vicuna 1.3	Vicuna 1.5	72.19	71.54	78.10	6.56	10.48	-4.06	**-38.74**
PIQA	Llama 1	Llama 2	74.86	74.76	79.27	4.51	11.59	-3.97	**-34.25**
	Phi 1.5	Phi 2	74.65	78.78	78.94	0.16	6.20	-0.11	**-1.77**
	Phi 1	Phi 1.5	59.03	74.65	74.70	0.05	7.78	-0.05	**-0.64**
	Vicuna 1.3	Vicuna 1.5	74.70	75.52	78.89	3.37	9.90	-2.88	**-29.09**

Table 4: Compatible task adapter trained with MUSCLE (corresponding to metrics with suffix c) reduces negative flip rate on the test sets of multiple-choice language tasks. We see most improvements for model updates that have small performance differences.

$\mathcal{M}_{v1}^{\text{base}}$	$\mathcal{M}_{v2}^{\text{base}}$	$\text{EM}_{\mathcal{M}_{v1}}$	$\text{EM}_{\mathcal{M}_{v2}}$	EM_c	$\Delta\text{EM}_c \uparrow$	NFR	$\Delta\text{NFR}_c \downarrow$	$\Delta_\%\text{NFR}_c \downarrow$
Llama 1	Llama 2	24.45	33.09	36.66	3.57	8.49	-0.91	**-10.72**
Phi 1.5	Phi 2	30.02	48.18	50.68	2.50	5.88	-1.71	**-29.08**
Phi 1	Phi 1.5	3.41	30.02	26.99	-3.03	2.01	-0.04	**-1.99**
Vicuna 1.3	Vicuna 1.5	26.72	29.91	31.84	1.93	11.60	-0.99	**-8.53**

Table 5: GSM8K math evaluation with exact match (EM) over the test dataset. For compatibility adapter (corresponding to metrics with suffix c), we observe a decreased negative flip rate while mostly maintaining performance gains.
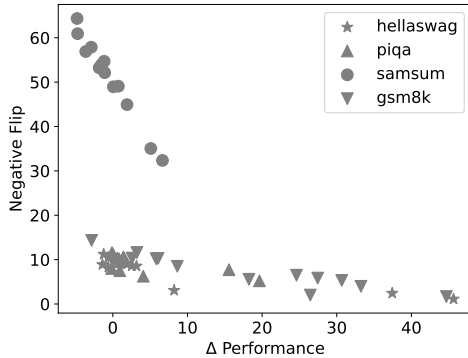

Figure 4: When updating LLM models (e.g. Llama 1 → Llama 2), we observe negative flips in different tasks. The smaller the performance gap from an old model to a new model, the more negative flips we observe. We indicate the performance gap by the difference in exact match for GSM8K, Rouge-1 for SAMSum, and log-likelihood-based accuracy for PIQA and HellaSwag. When evaluating continuous metrics with absolute ROUGE-1 value for summarization on SAMSum, we observe a large fraction of negative flips. We show the exact models analyzed in Table 9.

6.2 Reduced Negative Flips in Classification

In Table 4, we observe that MUSCLE decreases negative flips when compared to regular model updates without compatibility-specific training ($\mathcal{M}_{v2}$). Specifically, we see a reduction of NFR by 40% for the update of LLama 1 to Llama 2 and 39% for Vicuna 1.3 to Vicuna 1.5. For updates with

a large performance gap (for example Phi 1 to Phi 1.5), we observe a less strong enhancement with a negative flip rate reduction by 1-5%. In addition to the reduction of negative flips, we also observe an increased accuracy of up to 7% for Llama and Vicuna updates.

In exact match (EM) evaluation, we match the final result of the math question from the prediction to ground truth. Results are shown in Table 5. In this case, we observe that we can reduce the number of negative flips by 29% for Phi 1.5 to Phi 2. When the version 1 model is significantly less accurate (e.g., 3.4% exact-match accuracy for Phi 1), we observe a reduction in accuracy with the compatibility adapter while only being able to decrease negative flips by 2%. For all other updates, MUSCLE increases expected-match accuracy while reducing negative flips.

6.3 Increased Consistent Behavior

When we cannot achieve a positive flip (switching from an incorrect to a correct answer), we might prefer to at least maintain consistent behavior to the old model to avoid unexpected behavior for the user. We evaluate negative flips rate (NFR) and inconsistency flips rate (NFR_{mc}). For model updates that have a large performance gap and a small number of negative flips to begin with (Phi models), we see a limited reduction in inconsistency

$\mathcal{M}_{v1}^{\text{base}}$	$\mathcal{M}_{v2}^{\text{base}}$	$R1_{\mathcal{M}_{v1}}$	$R1_{\mathcal{M}_{v2}}$	$R1_c$	$\Delta R1_c \uparrow$	$\widetilde{\text{NFR}}$	$\Delta\widetilde{\text{NFR}}_c \downarrow$	$\Delta_{\%}\widetilde{\text{NFR}}_c \downarrow$	$\Delta m_g \uparrow$	$\Delta m_r \downarrow$
Llama 1	Llama 2	32.06	32.28	34.79	2.51	48.96	-7.81	**-15.95**	0.64	**-1.13**
Phi 1.5	Phi 2	37.53	36.15	40.69	4.54	54.70	-15.02	**-27.46**	0.24	**-3.39**
Phi 1	Phi 1.5	30.92	37.53	38.76	1.23	32.60	-5.74	**-17.61**	-0.38	**-0.59**
Vicuna 1.3	Vicuna 1.5	30.32	30.88	34.08	3.20	49.69	-10.98	**-22.10**	-0.02	**-1.85**

Table 6: For summarization (SAMsum) generative task we reduce model update regression of ROUGE-1 score (R1) by up to 27%.
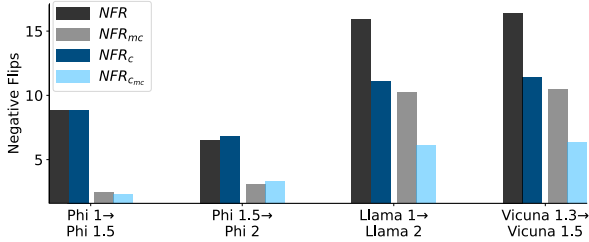

Figure 5: Comparison of NFR vs NFR_{mc} metrics to evaluate inconsistency when updating LLMs for HellaSwag task. We see that using our compatibility adapter (denoted by $_c$), we can reduce inconsistency for Llama and Vicuna models.

	ΔNFR_c	$\Delta_{\%}\text{NFR}_c$	ΔPFR_c	Δacc_c
$\mathcal{L}_{comp}^{\mathcal{M}_{v2}^C \neq y}$	**-3.97**	**-34.25**	0.54	4.51
$\mathcal{L}_{comp}^{\mathcal{M}_{v1} = y}$	-2.29	-19.76	-0.60	1.68
$a_{\mathcal{M}_{v1}}$	-2.99	-25.80	-0.82	2.17
$\text{CE}+\mathcal{L}_{comp}^{\mathcal{M}_{v2}^C \neq y}$	-3.10	-26.75	0.38	3.48
$\text{CE}+\mathcal{L}_{comp}^{LL_L}$	-2.12	-18.29	-0.49	1.63
$\text{CE}+\mathcal{L}_{comp}^{LL_S}$	-1.96	-16.91	0.05	2.01

Table 7: A model update from Llama 1 → Llama 2 on PIQA. Different ablations and masking strategies and their impact on negative flips, positive flips, and accuracy improvement.

flips. However, we observe that we can reduce the inconsistency flips with our method for model updates with small accuracy gaps such as the updates for Llama and Vicuna (Fig. 5) on the HellaSwag dataset.

6.4 Reduced Model Update Regression in Generative Tasks

When evaluating generative tasks, we can reduce model update regression of ROUGE-1 score performance (Table 6). We can reduce negative flips by 18% for updates with weaker version 1 models (Phi 1 → Phi 1.5) and 22% for smaller model updates (Vicuna 1.3 → Vicuna 1.5) and 27% for Phi 1.5 → Phi 2. We see that we decrease ROUGE-1 model update regression by 1-3%, while maintaining gain.

6.5 The Effect of Different Masking Strategies

We analyze different training and masking strategies to evaluate our design choices. On the PIQA dataset and model update Llama 1 → Llama 2, we compare MUSCLE with different masking strategies. Intuitively, instance or token-wise likelihood-based masking strategies can be useful for tasks that are evaluated with log-likelihoods (e.g. PIQA, HellaSwag) where multiple choices are available. We compare likelihood-based masking for example on individual tokens,

$$m_i = LL_L = \mathbf{1}[\sigma(z_{\mathcal{M}_{v2}^C,i}) < \sigma(z_{\mathcal{M}_{v1},i})] \quad (2)$$

to check if the likelihood of the ground-truth next token in the current model is smaller than the old model. Only in this case, we align to $\mathcal{M}_{v1}$, and align to $\mathcal{M}_{v2}$ for every other token. Alternatively, we can also compare the likelihood of the entire sequence

$$m_i = LL_S = \mathbf{1}[\sum_i [\sigma(z_{\mathcal{M}_{v2}^C,i}) < \sigma(z_{\mathcal{M}_{v1},i})]] \quad (3)$$

such that we mask all tokens per sequence to get instance-wise masking. For both of these strategies, we see a reduction in negative flips. We note that likelihood-based masking requires auxiliary cross-entropy loss for stability (A.4). Aligning to the old model only if it is already correct with a mask $\mathcal{L}_{comp}^{\mathcal{M}_{v1}=y}$ with $m_i = \mathbf{1}[\sigma(z_{\mathcal{M}_{v1},i}) = y_i]$ or without masking with KL ($a_{\mathcal{M}_{v1}}$) leads to a small reduction in negative flips. Our approach that aligns to the old model if the current prediction is incorrect ($\mathcal{L}_{comp}^{\mathcal{M}_{v2}^C \neq y}$), leads to the best reduction in negative flips while providing the biggest accuracy gain and increased positive flips (Table 7). This strategy has the additional advantage that it takes into account extended inconsistent flips explained in Fig. 5, as it aligns to $\mathcal{M}_{v1}$ whenever $\mathcal{M}_{v2}$ is incorrect. For this best-performing strategy, we see that including cross-entropy loss ($\text{CE}+\mathcal{L}_{comp}^{\mathcal{M}_{v2}^C \neq y}$) does not lead to additional performance gains.

6.6 Behavior for Different Model Pairs

A general point in the analysis of negative flips is its connection with accuracy: if accuracy is 100%, the NFR must be 0%. This means that compatible model updates become harder when updating $\mathcal{M}_{v1} \to \mathcal{M}_{v2}$ with lower $\mathcal{M}_{v2}$ accuracy. We report observed compatibility when using vanilla adapter learning, and show a reduction in NFR when using the proposed method (MUSCLE). When $\text{acc}(\mathcal{M}_{v2}) > \text{acc}(\mathcal{M}_{v1})$ for a task, the observed NFR is small (Fig. 4). This has been the case irrespective of the model update scenario and task. We find that MUSCLE reduces the observed NFR the most when $\text{acc}(\mathcal{M}_{v2}) \approx \text{acc}(\mathcal{M}_{v1})$. In such cases, the incompatibility between vanilla model updates is not dominated by the fact that one model version is more accurate than the other, making it an easier use case to be improved by our proposed distillation loss. With multiple frequent incremental model updates in academia and industry, this is a very relevant use case to focus on.

7 Conclusion

In this work, we study the task-specific compatibility problem when updating LLMs. We show that LLM updates with different scenarios, e.g., changes in model architecture, optimization, or training dataset, exhibit significant negative flips – instances previously classified or generated correctly, and incorrectly after the model update. We extend the negative flip metric for discriminative and, for the first time, generative tasks, and report results for various models and tasks.

We propose a novel method (MUSCLE) to train task-specific compatibility adapters when updating an old LLM to a new LLM to reduce negative flips while maintaining performance gain. Our proposed method does not require a modification to the base model training and is only based on adapter training. Further, as opposed to previous works, the proposed solution does not require both versions of the model in memory to enable compatibility, which is often infeasible due to the large size of LLMs.

We observe a mitigation of negative flips of 40% for multiple-choice type evaluations, and 27% for continuous summarization evaluation. We also show insights into model properties that facilitate transfer, finding that our alignment masking strategy provides best results with the additional benefit of mitigating inconsistent update behavior.

8 Limitations, Risks and Future Work

We do not consider a model update that includes changes in tokenization and/or vocabulary size (e.g. Llama 2 (Touvron et al., 2023b) to LLama 3 (AI@Meta, 2024)). Future work can explore compatible vocabulary mapping strategies before learning from prior model versions.

Tackling Large Performance Gaps When there is a large performance gap between M_{v1} and M_{v2}, loss hyperparameter weighting could be an interesting avenue to explore. We unsuccessfully experimented with simple weighting based on average accuracy. We hypothesize that instance-based loss weighting could be a more promising approach to tackle this model update case. As this would require intensive experimentation, we leave this as an avenue for future work. In general, the utility of aligning to prior model versions is limited by the performance of the prior model version. For example, see the update from Phi 1 to Phi 1.5 in Table 5, where Phi 1 only has an accuracy of 3%. In this case, it is arguable if an alignment to $\mathcal{M}_{v1}$ is desired and if the strive for compatibility outweighs a possible performance drop.

MUSCLE Performance Improvements In our work, we observe an interesting overall performance improvement following knowledge transfer, which is an intriguing finding regarding the possibilities of this line of work. Previous works on model compatibility using distillation-like losses have also observed such a phenomenon (e.g., (Jaeckle et al., 2023)). Given that we initialize the compatibility adapter with M_{v2} (an independently trained task adapter) and continue fine-tuning it with knowledge transfer loss from M_{v1}, one can argue that the observed performance improvement demonstrates an ensemble knowledge effect (i.e., knowledge from both M_{v2} and M_{v1} is aggregated into our compatibility adapter).

Ethical Considerations and Risks One potential risk of the proposed approach for compatible task-specific LLM updates is the transfer of potential biases from the old model, $\mathcal{M}_{v1}$, to the new model trained through knowledge transfer. We did not explore this aspect in the current study.

9 Acknowledgements

References

AI@Meta. 2024. Llama 3 model card.

Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. 2019. Beyond accuracy: The role of mental models in human-ai team performance. In *Proceedings of the AAAI conference on human computation and crowdsourcing*, volume 7, pages 2–11.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. PIQA: reasoning about physical commonsense in natural language. *CoRR*, abs/1911.11641.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Andrea Caciolai, Verena Weber, Tobias Falke, Alessandro Pedrani, and Davide Bernardi. 2023. Regression-free model updates for spoken language understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 538–551.

Deng Cai, Elman Mansimov, Yi-An Lai, Yixuan Su, Lei Shu, and Yi Zhang. 2022. Measuring and reducing model update regression in structured prediction for nlp. *Advances in Neural Information Processing Systems*, 35:19384–19397.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hui Huang, Yingqi Qu, Jing Liu, Muyun Yang, and Tiejun Zhao. 2024. An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge models are task-specific classifiers.

Florian Jaeckle, Fartash Faghri, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. 2023. Fastfill: Efficient compatible model update. *arXiv preprint arXiv:2303.04766*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*.

Yi-An Lai, Elman Mansimov, Yuqing Xie, and Yi Zhang. 2023. Improving prediction backward-compatiblility in nlp model upgrade with gated fusion. *arXiv preprint arXiv:2302.02080*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023a. Textbooks are all you need ii: phi-1.5 technical report.

Zenan Li, Maorun Zhang, Jingwei Xu, Yuan Yao, Chun Cao, Taolue Chen, Xiaoxing Ma, and Jian Lü. 2023b. Lightweight approaches to dnn regression error reduction: An uncertainty alignment perspective. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 1187–1199. IEEE.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Wanqin Ma, Chenyang Yang, and Christian Kästner. 2023. (why) is my prompt getting worse? rethinking regression testing for evolving llm apis. *arXiv preprint arXiv:2311.11123*.

Ryuta Matsuno and Keita Sakuma. 2023. A robust backward compatibility metric for model retraining. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4190–4194.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Yujia Qin, Cheng Qian, Xu Han, Yankai Lin, Huadong Wang, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. Recyclable tuning for continual pre-training.

Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Elle: Efficient lifelong pre-training for emerging data. *arXiv preprint arXiv:2203.06311*.

Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. 2020. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*.

Vivek Ramanujan, Pavan Kumar Anasosalu Vasu, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. 2022. Forward compatible training for large-scale embedding retrieval systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19386–19395.

Lingmin Ran, Xiaodong Cun, Jia-Wei Liu, Rui Zhao, Song Zijie, Xintao Wang, Jussi Keppo, and Mike Zheng Shou. 2023. X-adapter: Adding universal compatibility of plugins for upgraded diffusion model.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Karsten Roth, Lukas Thede, Almut Sophia Koepke, Oriol Vinyals, Olivier Hénaff, and Zeynep Akata. 2024. Fantastic gains and where to find them: On the existence and prospect of general knowledge transfer between any pretrained model.

Tomoya Sakai. 2022. A generalized backward compatibility metric. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1525–1535.

Raphael Schumann, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao, and Yi Zhang. 2023. Backward compatibility during data updates by weight interpolation. *arXiv preprint arXiv:2301.10546*.

Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. 2020. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6368–6377.

Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. 2020. An empirical analysis of backward compatibility in machine learning systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3272–3280.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2022. Contrastive representation distillation.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

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

Frederik Träuble, Julius Von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Peter Gehler. 2021. Backward-compatible prediction updates: A probabilistic approach. *Advances in Neural Information Processing Systems*, 34:116–128.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:170p7.06209*.

Yuqing Xie, Yi-An Lai, Yuanjun Xiong, Yi Zhang, and Stefano Soatto. 2021. Regression bugs are in your model! measuring, reducing and analyzing regressions in nlp model updates. *arXiv preprint arXiv:2105.03048*.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. 2021. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14299–14308.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Binjie Zhang, Yixiao Ge, Yantao Shen, Yu Li, Chun Yuan, Xuyuan Xu, Yexin Wang, and Ying Shan. 2021. Hot-refresh model upgrades with regression-free compatible training in image retrieval. In *International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yue Zhao, Yantao Shen, Yuanjun Xiong, Shuo Yang, Wei Xia, Zhuowen Tu, Bernt Schiele, and Stefano Soatto. 2022. Elodi: Ensemble logit difference inhibition for positive-congruent training. *arXiv preprint arXiv:2205.06265*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

A   Appendix

A.1   Training Hyperparameters and Evaluation

We show an overview of the design choices of our LoRA adapter for task and compatibility training in Table 8. We learn about the entire context and answer tokens during fine-tuning. We find that successfully training the compatibility adapter generally requires a larger LoRA rank. Task adapters can be trained with lower rank, but also experience performance boost with higher rank. We use the training splits of the respective datasets where we use 80% of the training set as training data, and 20% as validation data for selecting the best model based on validation loss. We evaluate for selection after each epoch. All compatibility adapter models are initialized with the previously trained task adapter models for version 2 model. KL divergence temperature is set to 2. We use deepspeed for optimization (Rasley et al., 2020). For benchmark eval-

Hyperparameter	Value
Epochs	10
Learning Rate	1e-4
Grad. acc. steps	8
LoRA α	256
LoRA rank	128
Dropout	0.0
Adapter Layers	All linear
Warmup Steps	500
Weight decay	0.0

Table 8: Hyperparameters for the training setup of the task and compatibility adapters.

uation, we use LM evaluation harness (Gao et al., 2023), where most of the benchmarks (GSM8k, HellaSwag, PIQA) are already defined, and we use them as-is. We add an evaluation for SAMsum summarization evaluating ROUGE-1 score with a no-repeat N-gram size of 2, and generation stop words of "Dialogue:", "Summary:" (which are the keywords for the context and answer behavior), "</s>" and double new lines. All of our tasks are based on the English language.

A.2   Training Cost

All Experiments were run on NVIDIA A100 and H100 GPUs with an overall compute budget of 720x8 GPUh. Assuming hourly rates of 2.5$, this would amount to 14,400$. Compared to regular parameter-efficient fine-tuning, our compatibility method requires both model versions in GPU memory during training, hence fewer data batches can be processed per time step. However, as the compatibility adapters are initialized with the task adapters, we can view it as a continued training. There are no differences for inference time and costs compared to regular parameter-efficiently fine-tuned LoRa adapters.

A.3   Model Update Evaluation

In the evaluation of model update regression for different model updates (Fig. 4), we consider the model pairs shown in Table 9.

A.4   Auxiliary Cross-Entropy Loss

For model updates with large performance gaps between $\mathcal{M}_{v1}$ and $\mathcal{M}_{v2}$, we find that an auxiliary cross-entropy loss enhances the stability of training. When the performance of $\mathcal{M}_{v1}$ is greatly inferior to $\mathcal{M}_{v2}$, we assume that the inferior $\mathcal{M}_{v1}$ introduces errors that steer the model away too much from the ground truth. We can account for this by adding the cross-entropy loss to align with the ground truth.

$\mathcal{M}_{v1}^{\text{base}}$	$\mathcal{M}_{v2}^{\text{base}}$
Phi 1	Phi 1.5
Phi 1	Phi 2
Phi 1.5	Phi 2
Vicuna 1.3	Vicuna 1.5
Llama 1	Llama 2
Llama 1	Vicuna 1.5
Llama 1	Vicuna 1.3
Vicuna 1.3	Llama 2
Llama 2	Vicuna 1.5
Llama 2	Llama 3
Llama 1	Llama 3
Vicuna 1.3	Llama 3
Vicuna 1.5	Llama 3

Table 9: Model pairs used to analyse model update regression.

Dataset	Train	Test
HellaSwag (Zellers et al., 2019)	40k	10k
PIQA (Bisk et al., 2019)	16k	3k
GSM8k (Cobbe et al., 2021)	7.5k	1.3k
SAMsum (Gliwa et al., 2019)	14.7k	819

Table 10: Dataset size for our experiments.

Using Eq. (1), we add a $\mathcal{L}_{CE}$ for training (scaled with hyperparameter λ) .

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\log(\sigma(z_{\mathcal{M}_{v2}^C,i,k})) \quad (4)$$

$$\mathcal{L} = \lambda\mathcal{L}_{comp}^m + (1-\lambda)\mathcal{L}_{CE} \quad (5)$$

We find that likelihood-based masking (Section 6.5) also requires auxiliary cross-entropy loss for stability. Even though one model version might provide a higher likelihood than the other, this likelihood does not necessarily mean a suitable probability distribution over the vocabulary, hence adding alignment to the ground truth helps to provide better results.

A.5 Downstream Tasks

We describe the datasets used in our experiments. Each dataset poses unique challenges and targets different aspects of language understanding and reasoning. Dataset sizes are shown in Table 10.

GSM8K The Grade School Math 8K (GSM8K) dataset is designed to evaluate mathematical reasoning capabilities. It consists of grade school-level math word problems. These problems require the application of mathematical concepts and the ability to reason about quantities and operations in a textual format. The dataset is structured to test the performance of models on arithmetic, algebra, geometry, and statistics problems, reflecting

a wide range of mathematical knowledge in education. We use this dataset as a representative case to test problem-solving cases. We evaluate model update regression in this task with exact match, which is a strict evaluation on a fraction of produced tokens, but does not account for the reasoning process.

SAMSum The SAMSum dataset consists of dialogue summaries designed to facilitate the evaluation of automatic conversational summarization models. It contains dialogue instances, paired with human-written summaries. These conversations mimic real-life scenarios to enable learning to generate coherent and concise summaries. We use this dataset as a representative case to test language generation behavior. We evaluate model update regression in this task with ROUGE-1.

HellaSwag HellaSWAG is a dataset for assessing common sense reasoning and predictive text modeling. It builds on the SWAG (Zellers et al., 2018) dataset by providing more challenging distractors. HellaSWAG consists of multiple-choice scenarios where a model must predict the most plausible continuation among four options, focusing on everyday activities, scenarios, and interactions. We use this dataset as a representative case to test abilities that require not just linguistic understanding but also real-world knowledge and common sense reasoning. We evaluate model update regression in this task with log-likelihoods for the correct answer, as multiple choices are given.

PIQA The Physical Interaction Question Answering (PIQA) dataset tests the understanding of physical and causal interactions in the real world through textual descriptions. It contains scenarios that require reasoning about physical properties, actions, and outcomes. Each scenario is presented as a question with two possible solutions, where the model must choose the most physically plausible one. We use this dataset as a representative case for evaluating models on tasks that require an understanding of the physical world and its governing principles. We evaluate model update regression in this task with log-likelihoods for the correct answer, as different choices are given.