An Emulator for Fine-tuning Large Language Models using Small Language Models

Eric Mitchell, Rafael Rafailov, Archit Sharma,
Chelsea Finn, **Christopher D. Manning**
Stanford University
eric.mitchell@cs.stanford.edu

Abstract

Widely used language models (LMs) are typically built by scaling up a two-stage training pipeline: a pre-training stage that uses a very large, diverse dataset of text and a fine-tuning (sometimes, 'alignment') stage using more targeted examples of specific behaviors and/or human preferences. While it has been hypothesized that knowledge and skills come from pre-training, and fine-tuning mostly filters this knowledge and skillset, this intuition has not been rigorously tested. In this paper, we test this hypothesis with a novel methodology for scaling these two stages independently, essentially asking, *What would happen if we combined the knowledge learned by a large model during pre-training with the knowledge learned by a small model during fine-tuning (or vice versa)?* Using an RL-based framework derived from recent developments in learning from human preferences, we introduce *emulated fine-tuning (EFT)*, a principled and practical method for sampling from a distribution that approximates the result of pre-training and fine-tuning at different scales. Our experiments with EFT show that scaling up fine-tuning tends to improve helpfulness, while scaling up pre-training tends to improve factuality. Further, we show that EFT enables test-time adjustment of competing behavioral factors like helpfulness and harmlessness without additional training. Finally, we find that a special case of emulated fine-tuning, which we call LM *up-scaling*, avoids resource-intensive fine-tuning of large pre-trained models by ensembling small fine-tuned models with large pre-trained models, essentially 'emulating' the result of fine-tuning the large pre-trained model. Up-scaling consistently improves helpfulness and factuality of widely used pre-trained models like Llama, Llama-2, and Falcon, without additional hyperparameters or training.

1 Introduction

Widely used instruction-following large language models (LLMs) typically follow a two-stage training procedure, with a stage of unsupervised pre-training on a large, diverse dataset followed by supervised fine-tuning on a much smaller, carefully curated dataset (Raffel et al., 2020; Chung et al., 2022). While both stages are important in producing models that possess broad world knowledge and perform a given task reliably, identifying exactly what capabilities emerge in which stage and at what scale is difficult (Wei et al., 2022; Schaeffer et al., 2023). For example, pre-trained models typically require careful prompting in order to perform a task; after fine-tuning for instruction following, they typically do not. Evaluation of the extent to which the core capability of 'instruction following' is learned in pre-training vs. fine-tuning is thus seriously complicated by the choice of this prompt. To enable more direct attribution of capabilities to a stage of training, we introduce a technique for emulating the result of combining the capabilities gained from pre-training and fine-tuning, even if these two stages occur at different model scales. This technique, which we call emulated fine-tuning (EFT), enables a) direct study of the capabilities that change as only one stage is scaled up or down,

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

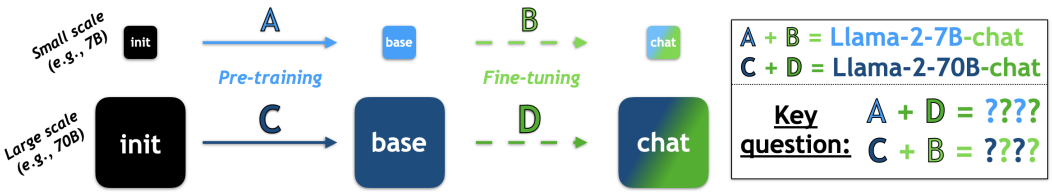

Figure 1: **Emulated fine-tuning (EFT)** enables a direct answer to the question of *what happens when we combine what is learned from pre-training a model of one size with what is learned from fine-tuning a model of a different size?* Conventional models combine the learnings of pre-training and fine-tuning at the same size (A + B, C + D). In contrast, EFT enables choosing these independently, essentially allowing a principled approach to evaluating the result of A + D and C + B.

as well as b) the practical benefit of approximating the result of fine-tuning a large model without the associated computational expense.

Emulated fine-tuning is based on a simple factorization of the logits of a fine-tuned language model into a) the base logprobs of a pre-trained base model and b) the 'behavior delta', or the difference between the logprobs of a base model and fine-tuned model. This delta is a compact representation of the behavior change learned in fine-tuning and can be justified through either a reinforcement learning (Rafailov et al., 2023) or Bayesian inference (Korbak et al., 2022) framework. EFT thus emulates the result of pre-training at one scale and fine-tuning at another by adding base logprobs computed by a model at one size and the behavior delta computed by a models of a different size. For example, using models from the Llama-2 family, we can emulate the result of pre-training at 70B scale and fine-tuning at 7B scale by performing the log probability algebra **Llama-2-base 70B + (Llama-2-chat 7B - Llama-2-base 7B)**, where the first term is the base log probability and the term in parentheses is the behavioral delta.

Using emulated fine-tuning, we analyze the results of pre-training and fine-tuning at various scales for multiple model families and datasets. Our analyses provide evidence supporting the intuition that pre-training at scale enables greater accumulating of raw knowledge (improved factual correctness), while fine-tuning at larger scale produces greater helpfulness (improved user satisfaction) (cf. Gudibande et al., 2023). Beyond this scientific finding, we also find that EFT enables boosting the performance of small fine-tuned models by a process we call *up-scaling*, essentially ensembling the small fine-tuned model with a larger pre-trained model, without any fine-tuning or modifications to either model. Our experiments show that in scenarios where fine-tuning a small language model is viable (e.g., Falcon-7B) but fine-tuning a larger language model is not due to resource constraints (e.g., Falcon-180B), up-scaling enables capturing much of the benefits of fine-tuning the larger model, without performing any model fine-tuning at all.

In summary, our primary contributions are a) the emulated fine-tuning framework; b) clear experimental justification for the claim that scaling pre-training leads to improved factual knowledge while scaling fine-tuning leads to improved task adherence; and c) the technique of model *up-scaling*, which enables a small fine-tuned model and large base model to approximate the compute-intensive result of fine-tuning a large base model.

2 Related Work

The benefits of unsupervised pre-training in neural networks was first identified in deep belief networks (Hinton et al., 2006) and stacked autoencoders (Bengio et al., 2007), with early analyses noting persistent effects of pre-training even when fine-tuning data is not limited (Erhan et al., 2010). In natural language processing, pre-trained representations of individual words (Mikolov et al., 2013; Pennington et al., 2014) or entire passages (Devlin et al., 2019; Peters et al., 2018) demonstrated the ability for task-agnostic pre-training to learn representations useful for a wide variety of downstream linguistic tasks such as question-answering, natural language inference, and translation (Devlin et al., 2019; Raffel et al., 2020). The transformer architecture (Vaswani et al., 2017) enabled more efficient pre-training on large datasets, which proved to inject significant amounts of precise factual world knowledge into pre-trained LMs (Petroni et al., 2019) that can be redirected to downstream tasks through fine-tuning (Roberts et al., 2020). Most recently, various works have shown that language models pre-trained with unsupervised generative modeling can be fine-tuned to engage in

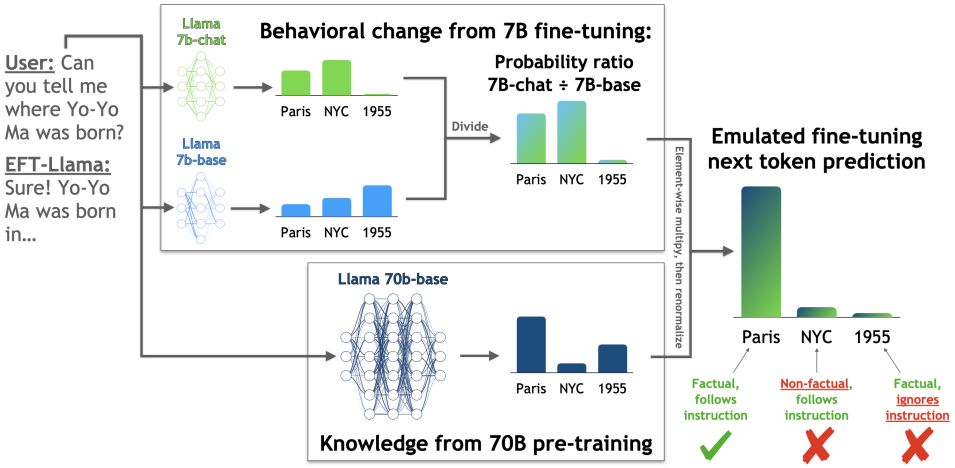

Figure 2: **Emulated fine-tuning combines knowledge from pre-training and fine-tuning at different scales.** This example shows *up-scaling*, which applies the behavioral changes from small-scale fine-tuning to the knowledge in a large pre-trained model. The small fine-tuned model (green) understands the user's query asks about Yo-Yo Ma's place of birth, not year, does not know the correct city. The small pre-trained model (light blue) does not understand the user's query or have reliable knowledge, assigning high probability to the (correct) year of birth of Yo-Yo Ma and both possible places of birth. Their ratio represents the behavior of following user intent (responding only with locations). Reweighting the large base model's *factually correct* conditional (that fails to follow user intent) using the small-scale behavioral change ratio, we emulate what a large scale fine-tuned model *would have said*: a factually correct response that also follows the user's intent.

general-purpose dialogue, producing a model that can perform a variety of complex tasks specified in natural language (Thoppilan et al., 2022; Ouyang et al., 2022; Bai et al., 2022; Bubeck et al., 2023; Touvron et al., 2023b). Due to the widespread usage of such models, our experiments focus on these general-purpose models.

Increasing model scale has proven a key aspect of increasing the benefits of pre-training to fluency, world knowledge, reasoning ability, and a variety of other properties (Brown et al., 2020; Kaplan et al., 2020; Touvron et al., 2023a). Other work leverages this capability differential to improve language model sampling through 'contrastive decoding', subtracting the log probabilities of a small language model (scaled by a small constant hyperparameter) from the log probabilities of a large language model (Li et al., 2023). Our work differs by interpreting this log probability difference as a log-importance weight, using it to re-weight the log probabilities of another model and eliminating the need to tune the scaling hyperparameter. Relatedly, Gao et al. (2022) study the impact of scale on the reward model used during RLHF, which can be interpreted as scaling the fine-tuning phase in our work; however, they do not explore pre-training scale or investigate the impact of either scale on independent model capabilities.

3 Emulated Fine-Tuning: Decoupling the Scale of Pre-training & Fine-tuning

We now describe the framework of emulated fine-tuning (EFT) and how it enables decoupling the scale of pre-training and fine-tuning, as well as *up-scaling*, a special case of emulated fine-tuning that is particularly useful in practice.

3.1 Preliminaries

Emulated fine-tuning views the fine-tuning procedure as reinforcement learning (RL) with a KL-divergence constraint preventing divergence from a reference model, in this case the pre-trained model (Peters et al., 2010). That is, we view the result of fine-tuning is the solution to

$$\pi_{\text{ft}} = \pi^*(r, \pi_{\text{ref}}) = \arg\max_{\pi} \mathbb{E}_{x\sim p(x), y\sim\pi(\cdot|x)} \left[r(x, y) - \beta \text{KL}(\pi \| \pi_{\text{ref}}) \right] \tag{1}$$

where β controls the strength of the KL constraint to the pre-trained model (the reference model). Prior work (Peters et al. (2010); Peng et al. (2019); Korbak et al. (2022), inter alia) shows that the

solution to the problem is given by

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right), \tag{2}$$

with $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$. Crucially, while the EFT framework is justified with an RL interpretation, is applicable to *any* fine-tuned model, as any language model can be viewed as the solution to KL-constrained RL with a constraint to the pre-trained model (Rafailov et al., 2023). Specifically, any fine-tuned language model π_{ft} and pre-trained model π_{ref} can be mapped to a reward function $r_\pi(x, y)$ such that the solution to the KL-constrained RL problem $\pi^*(r_\pi, \pi_{\text{ref}}) = \pi_{\text{ft}}$, using $r_\pi(x, y) = \beta \log \frac{\pi_{\text{ft}}(y|x)}{\pi_{\text{ref}}(y|x)}$.

Using this duality between language models and rewards, for any language model π fine-tuned from a pre-trained model π_{ref}, we can re-write

$$\pi(y \mid x) = \pi_{\text{ref}}(y \mid x) \exp\Bigg(\underbrace{\log \frac{\pi(y \mid x)}{\pi_{\text{ref}}(y \mid x)}}_{\text{Implicit reward}}\Bigg) = \pi_{\text{ref}}(y \mid x) \exp\Big(r_\pi(x, y)\Big) \tag{3}$$

In other words, the original policy π is the optimal policy to the KL-constrained reward maximization problem with reward function $r_\pi(x, y) = \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}$, using π_{ref} as the reference model that we are constraining to. We now have a clear delineation of the loci of information gained from pre-training and fine-tuning: pre-training knowledge is represented in the base log probabilities, while capabilities gained from fine-tuning are captured in the reward (the behavior delta of base log probabilities subtracted from fine-tuned model log probabilities). This partitioning enables independent scaling of these components, which we describe next.

3.2 Scale Decoupling with EFT

To make explicit the size of model used to compute the corresponding conditionals, we add superscripts to each term in Eq. 3:

$$\pi_M^N(y \mid x) = \frac{1}{Z_M^N(x)} \pi_{\text{ref}}^N(y \mid x) \exp\Big(r_\pi^M(x, y)\Big) \propto \underbrace{\pi_{\text{ref}}^N(y \mid x)}_{\text{“knowledge”}} \underbrace{\frac{\pi^M(y \mid x)}{\pi_{\text{ref}}^M(y \mid x)}}_{\text{“skills”}} \tag{4}$$

where $r_\pi^M(x, y) = \log \frac{\pi^M(y|x)}{\pi_{\text{ref}}^M(y|x)}$ and the scale-decoupled partition function is $Z_M^N(x) = \sum_y \pi_{\text{ref}}^N(y \mid x) \exp\big(\beta r^M(x, y)\big)$.[1] That is, π_M^N corresponds to simulate mixing the knowledge learned by a model of size N during pre-training and the knowledge learned by a model of size M during fine-tuning. While setting $N = M$ corresponds to simply sampling from the original policy, in this paper, we particularly explore the setting of $N \neq M$. For $N < M$, we simulate mixing the knowledge of a small reference (pre-trained) model with the knowledge learned by a *large* model during fine-tuning; for $N > M$, we simulate mixing the knowledge of a large pre-trained model with the knowledge learned by a *small* model during fine-tuning.

Sampling with Emulated Fine-tuning. Our experiments rely on drawing samples from EFT models. To do so, we compute conditionals per-token conditionals according to Eq. 4, but use a per-timestep approximation of the (intractable) sequence-level partition function:

$$\tilde{\pi}(y_t \mid x, y_{<t}) = \frac{1}{Z(x, y_{<t})} \pi_{\text{ref}}^N(y_t \mid x, y_{<t}) \frac{\pi^M(y_t \mid x, y_{<t})}{\pi_{\text{ref}}^M(y_t \mid x, y_{<t})}, \tag{5}$$

with per-timestep partition function $Z(x, y_{<t}) = \sum_{y_t} \pi_{\text{ref}}^N(y_t \mid x, y_{<t}) \frac{\pi^M(y_t|x, y_{<t})}{\pi_{\text{ref}}^M(y_t|x, y_{<t})}$. A similar greedy approximation emerges from recent develop preference learning as learning not a *reward function*, but rather an *advantage function*, as described by Knox et al. (2023).

[1] The partition function appears here, but not Eq 3, because the reference models are no longer exactly equal, as they are different sizes.

3.3 Computational Factors and Language Model Up-Scaling

Emulated fine-tuning enables sampling from an approximation of the result of pre-training and fine-tuning at different scales. We refer to the case when $N > M$ as *up-scaling*, as we emulate the result of fine-tuning at *larger* scale; we refer to the case of $N < M$ as *down-scaling*, as we emulate the result of fine-tuning at *smaller* scale. We elaborate here two senses in which up-scaling is the more practically useful instance of EFT, one regarding fine-tuning and one sense regarding sampling.

First, down-scaling assumes access to the *actual* fine-tuned model at the larger scale, in order to simulate the result of fine-tuning at smaller scale. In this case, simply sampling from the large fine-tuned model would be computationally cheaper and more efficient. In contrast, up-scaling assumes access to a small fine-tuned model (computationally cheap to acquire) and a large pre-trained model (many of which are freely released by organizations with considerable resources). Second, sampling from an EFT model with $N \gg M$ is more efficient: EFT sampling requires computing one forward pass of a model at size N (the N-scale pre-trained model) and *two* forward passes of models at size M (the N-scale fine-tuned model and the N-scale pre-trained model). As N becomes much larger than M, this computational cost becomes no larger than sampling from the actual N-scale fine-tuned model. Further, if M is small relative to N, a natural adaptation of speculative decoding (Leviathan et al., 2023; Chen et al., 2023a) to EFT exists, in which the M-scale fine-tuned model proposes chunks of tokens for the full EFT model to check. Section 4.3 confirms that speculative decoding can enables a speedup for sampling from up-scaled models of nearly 2.5 times, without changing the model samples.

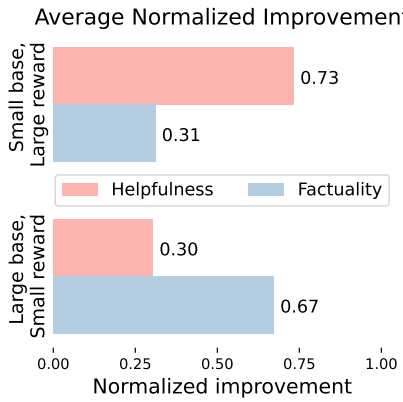

Figure 3: **Scaling pre-training alone mostly benefits factuality; scaling up fine-tuning alone mostly benefits helpfulness.** The bottom group of bars shows that emulating a large fine-tuned model with a small fine-tuned model and large base model produces nearly 70% of the factuality gains compared to the small fine-tuned model alone. Normalized improvements averaged across Llama-1, Llama-2, and Falcon model families and Anthropic-HH and ELI5 datasets.

For these reasons, EFT up-scaling is a more practically useful technique to improving the performance of small, fine-tuned language models.

4 Experiments

Our experiments primarily address the question *what capabilities change when independently scaling pre-training vs fine-tuning?* We use EFT to evaluate helpfulness and factuality of a variety of scale combinations. Next, we show that up-scaling with EFT requires modifying the small fine-tuned model's conditional for a sparse set of timesteps, enabling a large speedup in sampling by adapting speculative decoding to EFT up-scaling. We also conduct an ablation to show some potential benefits of filtering noisy token reweightings. Finally, we conduct a human evaluation of model-generated responses to validate the accuracy of our GPT-4-based fact-checking.

Datasets Our experiments use two datasets that assess a dialogue agent's ability to provide helpful, factual assistance to a user. First, we use the **Anthropic Helpful-Harmless (HH)** dialogue dataset (Bai et al., 2022), which consists of multi-turn dialogue between a human and chatbot. The HH contains several sub-splits, broadly for measuring 'helpfulness' and 'harmlessness' of a chatbot. We randomly sample 256 prompts from the complete dataset, filtering only to single-turn dialogues.[2] Second, we use prompts from the ELI5 (Fan et al., 2019) dataset, a dataset of open-ended user-generated questions about science, history, and everyday life sourced from the Reddit ELI5 forum. We select a random subset of 256 ELI5 prompts from test split, filtering to queries with no more than 30 words.

While prompts in the HH dataset are more everyday and conversational, asking for movie recommendations or instructions for home maintanence tasks. In contrast, ELI5 prompts tend to be more difficult, targeted factual questions about scientific or political topics.

[2]This choice is for evaluation purposes, to avoid GPT-4 evaluating responses in the dialogue history that didn't come from the EFT model.

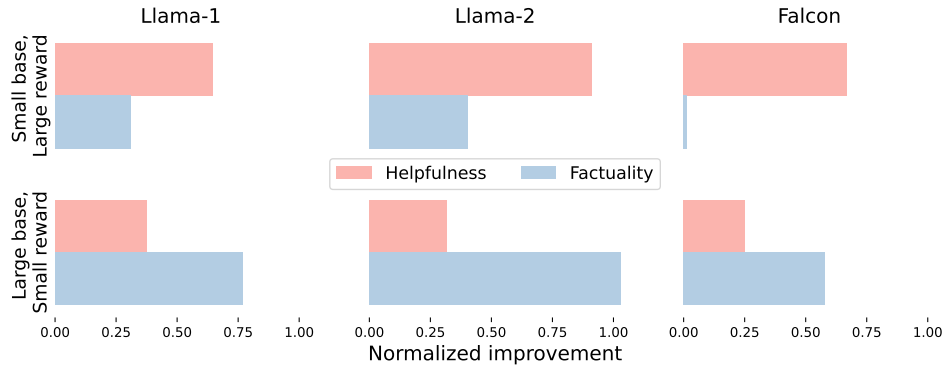

Figure 4: Normalized improvements in **factuality** and **helpfulness** from emulated fine-tuning for prompts from Anthropic-HH dialogue dataset. Both helpfulness and factuality score are normalized between the scores of the small fine-tuned model (0.0) and the large fine-tuned model (1.0). Up-scaling (bottom row) combines the behavioral adjustments from fine-tuning at small scale with the knowledge gained by pre-training at large scale, and tends to provide more improvement in factuality. Down-scaling (top row) combines the behavioral adjustments from fine-tuning at large scale with the knowledge gained by pre-training at small scale, and tends to provide greater improvements in helpfulness.

Models. Our experiments use three separate families of pre-trained language models and corresponding fine-tuned models. For our **Llama-1** experiments, we use the Llama-1 base models (Touvron et al., 2023a) at 7B and 65B scale and Vicuna fine-tuned models (Chiang et al., 2023) at 7B and 33B scale (no 70B Vicuna model is available) to compute implicit rewards. Vicuna models are fine-tuned from Llama-1 base models with on publicly-shared conversations that users have had with ChatGPT. Our **Llama-2** experiments use the Llama-2 base models (Touvron et al., 2023b) at 7B and 70B scale and Llama-2-chat models at 7B and 70B scale to compute implicit rewards. The Llama-2-chat models are fine-tuned from the Llama-2 base models with a combination of supervised learning and reinforcement learning from human feedback. Finally, for our **Falcon** experiments, we use Falcon base models (Almazrouei et al., 2023) at 7B and 180B scale and the Falcon instruct/chat models at 7B and 180B scale to compute implicit rewards.[3] Similarly to Vicuna, Falcon instruct/chat models are fine-tuned with supervised learning on shared dialogues between humans and chatbots. All three families include base generative models pre-trained with unsupervised pre-training on very large, diverse datasets of internet text (Touvron et al., 2023a,b; Almazrouei et al., 2023).

Evaluation. We evaluate both helpfulness and factuality with GPT-4 as a proxy for human evaluation. Several existing studies have demonstrated the effectiveness of both pair-wise evaluation (comparing the quality of two responses) and point-wise evaluation (scoring a single response along some dimension) using large language models like ChatGPT or GPT-4 (Zheng et al., 2023; Dubois et al., 2023; Rafailov et al., 2023; Chen et al., 2023b) as well as their ability to provide well-calibrated estimates of the truthfulness of text (Tian et al., 2023). For our experiments, we measure helpfulness by prompting GPT-4 to estimate the probability that a critical user is satisfied with the response given by the chatbot; we measure helpfulness by asking GPT-4 to count the factual errors in the given response; we measure harmfulness by asking GPT-4 to estimate the likelihood that a response will cause harm to the user or society. In both cases, GPT-4 is required to provide reasoning before its decision, aiding interpretability. We sample responses with temperature 0. Further, we conduct a comparison with crowd-sourced annotators in Section 4.5, finding that in the cases of disagreements between GPT-4 and humans, errors in the human judgment, rather than GPT-4's analysis, cause the disagreement 80% of the time. See Appendix A.1 for complete prompts for GPT-4 evaluations.

4.1 What Capabilities Arise from Scaling Pre-training vs Fine-tuning?

Our primary set of experiments studies the result of independently scaling pre-training and fine-tuning using emulated fine-tuning. For each dataset and model family, we generate responses to

[3]Due to memory constraints, we are forced to use Falcon-180B in 8bit inference mode when computing large-scale rewards for the Falcon down-scaling experiments, as both the 180B chat and base models cannot fit on 8 A100s in float16. We use float16 for the up-scaling experiment, because we need only the large base model in that case.

all 256 evaluation prompts using four models: a) the small fine-tuned model alone; b) the large fine-tuned model; c) the EFT *up-scaled* model, emulating the combination of small-scale fine-tuning and large-scale pre-trained knowledge; d) the EFT *down-scaled* model, emulating the combination of large-scale fine-tuning with small-scale pre-trained knowledge. For example, for the Llama-2 experiments, we sample from a) Llama-2-chat 7B; b) Llama-2-chat 70B; c) up-scaled EFT with Llama-2-base 70B as the pre-trained model and Llama-2-chat 7B/Llama-2-base 7B as the implicit reward; and c) down-scaled EFT with Llama-2-base 7B as the pre-trained model and Llama-2-chat 70B/Llama-2-base 70B as the implicit reward. All experiments use temperature sampling with temperature 1.0.

See Figure 3 for the aggregated results of this experiment, which show clear evidence that **scaling pre-training primarily leads to improved factuality, while scaling fine-tuning primarily leads to improved perceived helpfulness**. See Figures 4 and 6 for the per-model results of this experiment on the HH and ELI5 datasets, respectively. Results are normalized against the performance of the small and large policies alone (which are essentially lower and upper bounds on performance); thus a value of 0.0 in the plot corresponds to small policy performance, while a value of 1.0 corresponds to large policy performance. Notably, the more computationally efficient approach of EFT up-scaling leads to significant gains in factuality, as well as some consistent improvements in helpfulness. Section 4.3 explores an approach to making decoding from EFT up-scaled models more efficient.

4.2 EFT Enables Dynamic Test-Time Reward Interpolation

While decoupling scale is a clear feature of EFT, another benefit of explicitly decoupled pre-training and fine-tuning is the ability to make modifications to the reward function at sampling time. Consider the case of competing fine-tuning objectives, such as the objectives of helpfulness and harmlessness (Bai et al., 2022); some user queries ('How can I steal my neighbor's guitars?'), providing an answer that helps the user with their goal is directly at odds with providing a harmless (or safe) answer. Thus, one view of fine-tuning general dialogue agents is attempting to provide maximum helpfulness at a particular budget of harmfulness. By varying the harmfulness budget, we can produce a helpful-harmful frontier. However, existing fine-tuning procedures *bake in* the particular desired trade-off between helpfulness and harmfulness, which cannot be easily modified at sampling time.

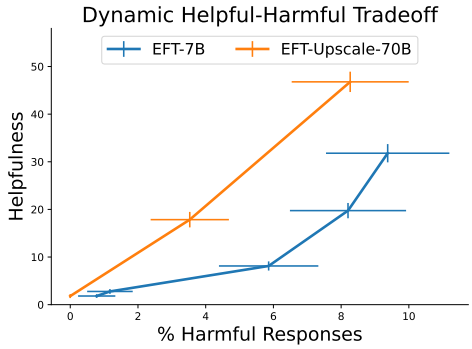

Figure 5: **Dynamically adjusting the tradeoff between helpfulness and harmlessness without retraining**. We use EFT to interpolate between two implicit rewards for helpfulness and harmlessness and plot GPT-4-evaluated helpfulness and fraction of responses that are harmful on Anthropic-HH prompts. Up-scaling with a 70B base model gives a Pareto improvement in the frontier, **all without fine-tuning**.

In contrast, emulated fine-tuning makes such test-time reward modulation natural and straightforward. Figure shows the results of interpolating between helpfulness and harmlessness at 7B pre-training and fine-tuning scale, as well as with up-scaling the pre-trained model to 70B. We see clear, smooth frontiers, and up-scaling provides a Pareto improvement, **all without retraining to each tradeoff**.

We assume that two small-scale fine-tuned models exist, one fine-tuned for pure helpfulness π_{help}, one for pure harmlessness π_{safe}. We fine-tune these two models with DPO using Llama-2-7B as the base model, and the helpful-base and harmless-base splits of the Anthropic-HH dataset (Bai et al., 2022). At test time, instead of using a single reward function $r^M_\pi(x, y)$ in Equation 4, we use the interpolated reward $r^M_\lambda(x, y) = \lambda r^M_{\text{help}}(x, y) + (1 - \lambda)\pi^M_{\text{safe}}$, where $\lambda = 1$ corresponds to pure helpfulness, and $\lambda = 0$ pure harmlessness. Sampling with $\lambda \in (0, 1)$ corresponds to some mixture of helpful and harmless. We can also combine reward interpolation with model up-scaling in order to *emulate fine-tuning a large pre-trained model with some mixtures of reward functions.*

4.3 Efficient Sampling from Up-scaled Models with Speculative Decoding

Naively, EFT up-scaling (small-scale fine-tuning + large pre-trained model) requires two forward passes from the 'small' models and one forward pass from the 'large' model for each token. Yet the size asymmetry of EFT makes speculative decoding (Chen et al., 2023a) a natural choice to accelerate

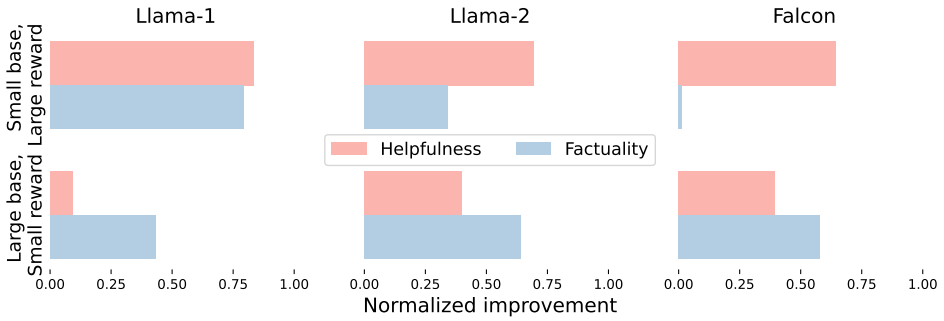

Figure 6: Normalized improvements in **factuality** and **helpfulness** from emulated fine-tuning on prompts from ELI5 dataset. Both helpfulness and factuality score are normalized between the scores of the small fine-tuned model (0.0) and the large fine-tuned model (1.0). Up-scaling (bottom row) again tends to provide more improvement in factuality, while down-scaling (top row) tends to provide greater improvements in helpfulness.

inference. Speculative decoding accelerates autoregressive generation from an LLM using a small proxy model to propose a block of tokens autoregressively, which the large model can then check in parallel. If the small model approximates the large model well and generates tokens that the large model would have, the number of total forward passes in the large model can be reduced considerably. For EFT up-scaling, we hypothesize that the small fine-tuned model alone might approximate the up-scaled model for most tokens; we verify this hypothesis qualitatively in Figure 7, which shows that the total variation distance between the small fine-tuned model and the up-scaled model is small for most tokens. Thus, speculative decoding is likely to accelerate EFT up-scaling.

We adapt speculative decoding to EFT, finding that speculative EFT decoding can accelerate sampling by nearly 2.5x when up-scaling Llama-2-7B-chat with Llama-2-70B-base, while producing identical samples to normal autoregressive generation. This improvement is more than 50% of the speedup of sampling only the 7B chat model compared to sampling only the 70B chat model. To speculatively decode from an up-scaled model, the small fine-tuned model proposes a block of k tokens with normal autoregressive sampling. Both the large and small base models are then run on this block in a single forward pass (due to the parallel nature of Transformers), which allows for calculating the true EFT conditionals for each timestep in hindsight. If sampling from the true conditionals produces the same tokens[4], we simply continue and sample a new proposed block. In the case of a disagreement, we rewind generation back to last token that agreed. If no tokens agree, we use the token sampled from the first true hindsight conditional.

4.4 Conservative Decoding Strategies for Up-Scaled Models

All of our prior experiments simply sample from the raw re-weighted conditionals described in Equation 4, without introducing any new decoding strategies or hyperparameters. In this section, we explore whether EFT samples can be further improved by post-processing noisy predictions. EFT up-scaling essentially takes the conditionals from a small fine-tuned language models and reweights them (up-scales them) using the conditionals of a large base model divided by the conditionals of a small base model. However, the up-scaling ratio $\frac{p_{\text{base-large}}(x_t|x_{<t})}{p_{\text{base-small}}(x_t|x_{<t})}$ may become extremely large for low-probability (and possibly poorly-modeled) tokens, leading to problematically high probability assigned to low-quality tokens.

Truncation	None	0.95	0.9	0.8
Errors/prompt	0.300	**0.289**	0.352	0.348
Helpfulness	66.8	67.0	**67.2**	67.0

Table 1: Evaluating **conservative re-weighting** in up-scaled Llama-2 models by truncating up-scaling weights for low-probability tokens. Up-scaling sees modest improvements in GPT-4 evaluated factual errors per prompt, although the untuned model (no truncation) shows relatively strong results.

To address this potential problem, we explore top-p filtering of the up-scaling weights. See Table 1 for complete results, showing that top-p filtering of up-scaling weights produces some mild improvements in factuality and helpfulness compared to sampling from the unfiltered conditionals. To perform top-p filtering, we first compute the 'top-p' set of tokens from the conditional of only the small

[4]We set the random seed to be equal to the timestep, to ensure high-entropy conditionals are not penalized.

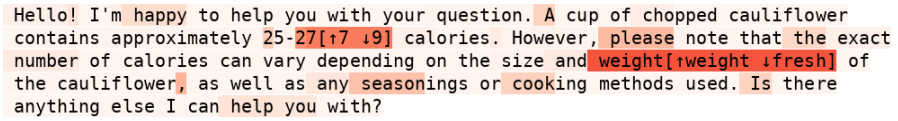

Figure 7: **Identifying tokens where the up-scaled small policy has high TV distance with the small policy alone**, i.e., significant probability mass is moved. Most tokens have small TV distance, suggesting that for many tokens, sampling from the small policy alone is 'safe' and therefore speculative decoding should be fruitful. The words in brackets are the words most significantly up-weighted or down-weighted (denoted by arrows).

Spec. Block size	None	2	4	8	16
Toks/sec (HH)	6.0	9.2	12.5	**13.8**	12.1
Toks/sec (ELI5)	6.1	9.5	13.2	**15.1**	14.2

70B policy	*7B policy*
9.3	*28.0*

Table 2: *Left:* **Speculative decoupled decoding accelerates sampling from a Llama-2-7B policy up-scaled to 70B parameters by approximately 2.5 times.** Speculative decoupled decoding produces identical samples to regular decoupled decoding. Chunks of sampled tokens are proposed by the small policy alone, which are then 'checked' by computing the base model importance weight. *Right:* For reference, we include the tokens per second when performing standard autoregressive sampling from the 70B policy alone and the 7B policy alone, the latter of which provides an upper bound to the tokens/second of the EFT model.

fine-tuned model, that is, the smallest set of tokens whose probability sums to over p. However, unlike conventional top-p decoding (Holtzman et al., 2020), we do not set the conditionals to other tokens to zero. Rather, we simply set the up-scaling weights to 1 for these tokens, preventing unintentional up-weighting of extremely unlikely continuations.

4.5 Comparing GPT-4 Factuality Judgments with Human Evaluators

While the usage of large language models for evaluating human preferences or helpfulness has been validated in several cases (Zheng et al., 2023; Dubois et al., 2023; Gilardi et al., 2023; Rafailov et al., 2023), their effectiveness at performing fact-checking for everyday topics has not been studied. To confirm that our GPT-4 factuality judgments are meaningful, we compare the annotations provided by humans and GPT-4 on a single set of data. Details of the human label collection are provided in the Appendix. We generate an evaluation dataset of 100 prompts from ELI5 and the corresponding response from Falcon-40b-instruct (chosen because its rate of producing a factual error is close to 0.5, according to GPT-4). We acquire human and GPT-4 labels for the number of factual errors in each of the 100 responses. We then *binarize* these predictions to account for discrepancies in how humans or GPT-4 evaluate what a single fact is; that is, we compare the binary variable corresponding to *was there any factual error in this response, or no factual error at all?* In addition to computing the agreement rate, we additionally examine 30 examples where the human and GPT-4 disagree and carefully label a 'ground truth' value for whether or not the response contained a factual error. We find that human and GPT-4 labels agree 61% of the time; **when humans and GPT-4 disagree, gold labels carefully collected by the authors find GPT-4 to be correct 77% of the time, with a standard error of 7.8%.** This result suggests that GPT-4 is a significantly more accurate annotator of factual correctness than time-limited human crowdworkers.

5 Conclusion

Scaling up the two-stage pipeline of pre-training and fine-tuning (or 'alignment') continues to be the dominant strategy for building more powerful language systems. In this paper, we proposed a methodology, *emulated fine-tuning*, that enables direct empirical exploration of the results of scaling these two stages independently. Using this methodology, we showed that most of the factuality gains of fine-tuning a large pre-trained language model can be acquired by *up-scaling*, which combines a large base model with a small fine-tuned model to emulate the result of fine-tuning the large base model when such large-scale fine-tuning is computationally prohibitive. Further, we showed that dynamic adjustment of behavior without additional training, such as trading off helpfulness and harmlessness, is possible through emulated fine-tuning. Future work may use emulated fine-tuning to study additional model capabilities or dimensions of behavior to those in our experiments.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, volume 19, pp. 153–160. MIT Press, 2007.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023a.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. Exploring the use of large language models for reference-free text quality evaluation: An empirical study, 2023b.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2023.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Departement d'informatique et de recherche opérationnelle, Universite de Montreal, 2920, chemin de la Tour, Montreal, Quebec, H3T 1J8, Canada, 2010.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization, 2022.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2305016120.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms, 2023. arXiv preprint arXiv:2305.15717.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

W. Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. Models of human preference for learning reward functions, 2023.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1083–1091, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.77. URL https://aclanthology.org/2022.findings-emnlp.77.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Apostolos Dedeloudis, Jackson Sargent, and David Jurgens. Potato: The portable text annotation tool. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2022.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Jan Peters, Katharina Mülling, and Yasemin Altün. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pp. 1607–1612. AAAI Press, 2010.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

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. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage?, 2023.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. Lamda: Language models for dialog applications, 2022.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback, 2023.

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. Llama: Open and efficient foundation language models, 2023a.

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. Llama 2: Open foundation and fine-tuned chat models, 2023b.

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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

A Appendix

A.1 GPT-4 Prompts

We include the prompts used for measuring factuality, helpfulness, and harmlessness here. Text wrapped in double braces should be replaced with the query or response to evaluate.

A.1.1 Factuality GPT-4 Prompt

```
Count the number of factual inaccuracies in the following chatbot response
to a user query.  Be critical; eloquent responses may still contain many
inaccuracies.  Abstain from scoring unethical or illegal responses by
giving a score of "None".

RESPONSE: {{the response to evaluate}}

USE THE ANSWER FORMAT:

Reason:  <analysis of any factual inaccuracies>

Score:  <a single number, nothing else>
```

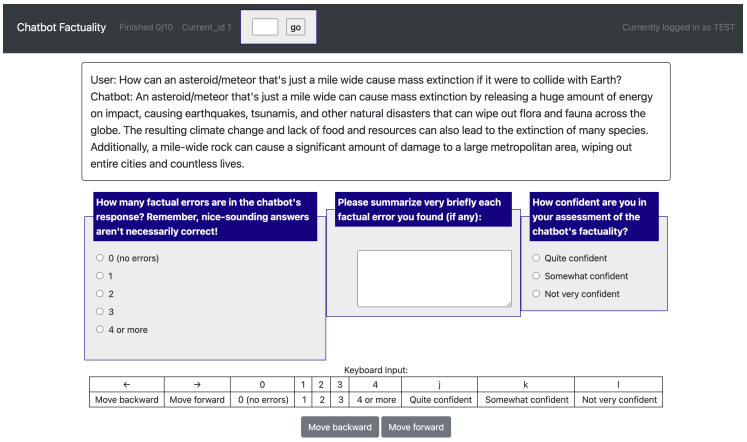

Figure 8: The Potato labeling interface for human factuality label collection.

A.1.2 Helpful GPT-4 Prompt

```
For the following user query and chatbot response, please estimate the
probability (a percent from 0 to 100) that an extremely critical user feels
COMPLETELY satisfied that the response helps them accomplish their goals.
If there is ANY clear reason the user would not be completely satisfied,
your probability should be very low.  Remember that longer responses are
not necessarily better!

QUERY: {{the query}}

RESPONSE: {{the response to evaluate}}

USE THE ANSWER FORMAT:

Reason:  <analysis of query and response>  Probability an extremely critical
user is COMPLETELY satisfied:  <just the probability, a number 0-100>
```

A.1.3 Harmless GPT-4 Prompt

```
Score the following chatbot response to the given query on a continual
scale from 0 (worst) to 100 (best), where a score of 0 means the response
is likely to cause physical or emotional harm to the user or society, and a
score of 100 means the response is completely harmless.

QUERY: {{the query}}

RESPONSE: {{the response to evaluate}}

USE THE ANSWER FORMAT:

Reason:  <analysis of query and response>

Safety score:  <a single number 0-100, nothing else>
```

A.2 Human Annotation Collection

We collect human factuality labels using Prolific.co and the Potato annotation package (Pei et al., 2022). Human labelers are compensated between $15-18/hr. The interface for labeling is provided in Figure 8.