Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision

Collin Burns[*] **Pavel Izmailov**[*] **Jan Hendrik Kirchner**[*] **Bowen Baker**[*] **Leo Gao**[*]

Leopold Aschenbrenner[*] **Yining Chen**[*] **Adrien Ecoffet**[*] **Manas Joglekar**[*]

Jan Leike **Ilya Sutskever** **Jeff Wu**[*]

OpenAI

Abstract

Widely used alignment techniques, such as reinforcement learning from human feedback (RLHF), rely on the ability of humans to supervise model behavior—for example, to evaluate whether a model faithfully followed instructions or generated safe outputs. However, future superhuman models will behave in complex ways too difficult for humans to reliably evaluate; humans will only be able to *weakly supervise* superhuman models. We study an analogy to this problem: can weak model supervision elicit the full capabilities of a much stronger model? We test this using a range of pretrained language models in the GPT-4 family on natural language processing (NLP), chess, and reward modeling tasks. We find that when we naively finetune strong pretrained models on labels generated by a weak model, they consistently perform better than their weak supervisors, a phenomenon we call *weak-to-strong generalization*. However, we are still far from recovering the full capabilities of strong models with naive finetuning alone, suggesting that techniques like RLHF may scale poorly to superhuman models without further work. We find that simple methods can often significantly improve weak-to-strong generalization: for example, when finetuning GPT-4 with a GPT-2-level supervisor and an auxiliary confidence loss, we can recover close to GPT-3.5-level performance on NLP tasks. Our results suggest that it is feasible to make empirical progress today on a fundamental challenge of aligning superhuman models.

1 Introduction

We mainly steer or *align* today's models with reinforcement learning from human feedback (RLHF): we reinforce behaviors that human evaluators rate highly and penalize behaviors that evaluators rate poorly (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Glaese et al., 2022; Bai et al., 2022a). This procedure is very effective when human evaluators can tell if model behavior is good or bad and is a core part of training modern language model assistants such as ChatGPT.

However, superhuman models will be capable of complex and creative behaviors that humans cannot fully understand. For example, if a superhuman assistant model generates a million lines of extremely complicated code, humans will not be able to provide reliable supervision for key alignment-relevant tasks, including: whether the code follows the user's intentions, whether the assistant model answers questions about the code honestly, whether the code is safe or dangerous to execute, and so on. As a result, if we finetune a superhuman model with human supervision on a reward modeling (RM) or safety classification task, it is unclear how that model will generalize to complicated behaviors that humans could not reliably supervise themselves.

This leads to a fundamental technical challenge of aligning superhuman models (superalignment): how can weak supervisors control models much smarter than them? Despite the importance of

[*] Primary authors. This was a joint project of the Superalignment Generalization team. Correspondence to `generalization@openai.com`. Code is available at `github.com/openai/weak-to-strong`.

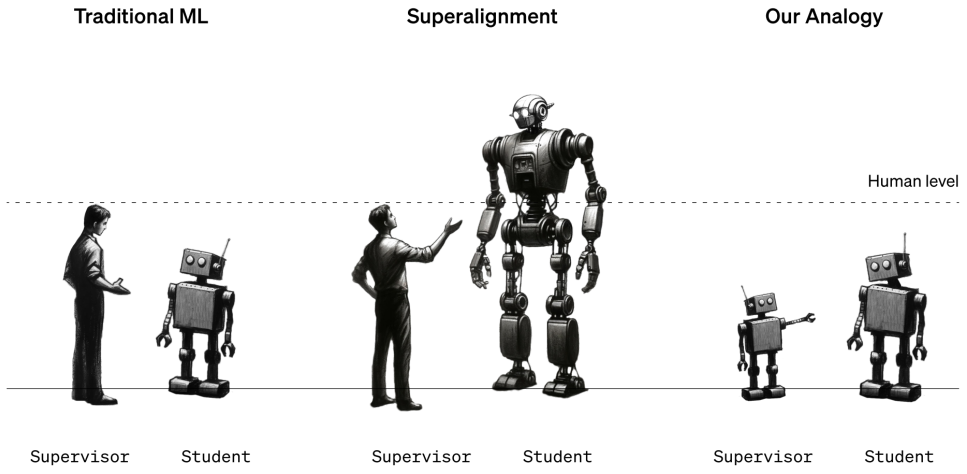


Figure 1: **An illustration of our methodology.** Traditional ML focuses on the setting where humans supervise models that are weaker than humans. For the ultimate superalignment problem, humans will have to supervise models much smarter than them. We study an analogous problem today: using weak models to supervise strong models.

this problem, it is difficult to empirically study today. Most prior work on alignment has either confronted this core challenge head-on—but been restricted to primarily theoretical frameworks and toy problems (Irving et al., 2018; Christiano et al., 2018; Leike et al., 2018; Demski & Garrabrant, 2019; Hubinger et al., 2019), or empirically studied humans supervising today's models—without addressing the core challenges that may arise with superhuman models (Christiano et al., 2017; Wu et al., 2021; Ouyang et al., 2022; Bowman et al., 2022; Saunders et al., 2022). In contrast, we would ideally like to have a setup that captures core challenges of aligning future superhuman models while *also* being able to make iterative empirical progress today.

We propose a simple setup for studying the problem of humans supervising superhuman models by considering an analogy: can we use *weak models* to supervise *strong models*? We can empirically test this by finetuning large (strong) pretrained models on labels generated by small (weak) models and observing how they generalize. Just like the problem of humans supervising superhuman models, our setup is an instance of what we call the *weak-to-strong learning* problem.

Why should weak-to-strong learning be possible? On the one hand, the strong model could simply learn to imitate the weak supervisor, including its errors, since that is what we would naively train it to do. On the other hand, strong pretrained models should already have good representations of the alignment-relevant tasks we care about. For example, if a model can generate complicated code, then it should intuitively also know whether that code faithfully adheres to the user's instructions. As a result, for the purposes of alignment we do not need the weak supervisor to teach the strong model new capabilities; instead, we simply need the weak supervisor to elicit what the strong model *already knows*. This gives us hope that the strong model can generalize beyond the weak supervision, solving even hard problems for which the weak supervisor can only give incomplete or flawed training labels. We call this phenomenon *weak-to-strong generalization*.

We study our weak-to-strong learning setup (Section 3) by finetuning base (i.e. pretrained-only) language models from the GPT-4 family (OpenAI, 2023),[1] spanning 7 orders of magnitude (OOMs) of pretraining compute, across three settings: a large set of popular natural language processing (NLP) benchmarks, chess puzzles, and our internal ChatGPT reward modeling dataset. Our main findings include:

[1] These models share the same general architecture and pretraining dataset as GPT-4. However, this model series does not include the models known as GPT-2, GPT-3, and GPT-3.5.

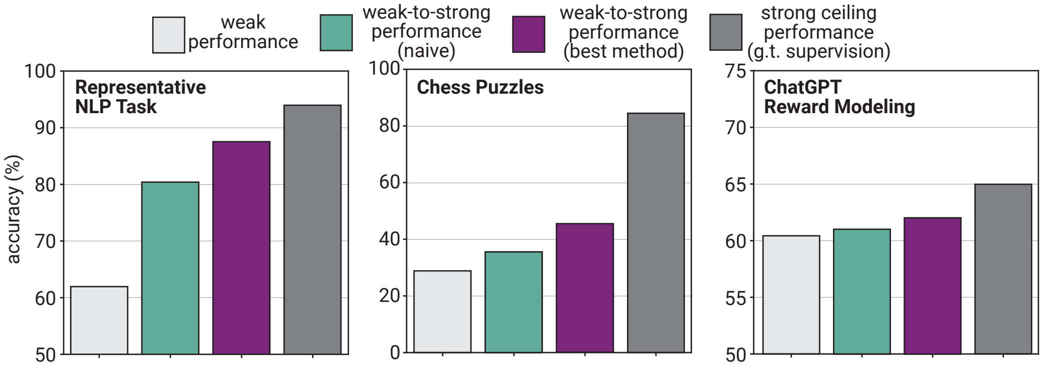


Figure 2: **Strong models trained with weak supervision generalize beyond their supervisor, and improving weak-to-strong generalization is tractable.** We show test accuracy on a representative NLP task (left), chess puzzles (middle) and the ChatGPT reward modeling task (right). We show the weak supervisor trained on ground truth labels (light grey) and the strong student trained with weak supervision naively (green), with the best method in each setting (purple), or with ground truth supervision (dark grey). For NLP and chess we supervise GPT-4 using GPT-2-level supervision, while for reward modeling we supervise a 3.5-level model using GPT-2-level supervision. The best method is the auxiliary confidence loss for the NLP task (Section 4.3.2), bootstrapping for Chess puzzles (Section 4.3.1), and unsupervised generative finetuning for reward modeling (Section 5.2.2; generative-finetuning is also used for the strong ceiling performance).

1. **Strong pretrained models naturally generalize beyond their weak supervisors.** If we naively finetune strong models with labels generated by weak models, they consistently outperform their weak supervisors (Section 4.2). For example, on NLP tasks, if we finetune GPT-4 with labels from a GPT-2-level model, we typically recover about half of the performance gap between the two models.

2. **Naively finetuning on weak supervison is not enough.** Despite positive weak-to-strong generalization, there still remains a substantial gap between strong models finetuned with weak supervision and strong models finetuned with ground truth supervision. Weak-to-strong generalization is particularly poor for ChatGPT reward modeling. Collectively, our results provide empirical evidence that naive RLHF will likely scale poorly to superhuman models without additional work.

3. **Improving weak-to-strong generalization is tractable.** We find that we can improve performance by encouraging strong models to have confident predictions with an auxiliary loss, bootstrapping supervision with intermediate models, and improving model representations with unsupervised finetuning. For example, when supervising GPT-4 with a GPT-2-level model on NLP tasks using the auxiliary confidence loss, we typically recover nearly 80% of the performance gap between the weak and strong models.

Our work has important limitations. None of our methods work consistently in all settings, and especially in the RM setting we are still far from recovering the full performance gap between weak and strong models. Thus our methods serve more as proofs-of-concept that weak-to-strong generalization is tractable, rather than practical solutions we recommend deploying today. Furthermore, there are still important disanalogies between our empirical setup and aligning superhuman models that we did not address (Section 6); continuously refining our basic setup will be important for ensuring that research today continues to make real progress toward aligning the superhuman models we develop in the future.

Despite the limitations of our work, we find our results to be highly encouraging. We show that substantial weak-to-strong generalization is not only possible, but actually a widespread phenomenon. We also show that with very simple methods, we can drastically improve the ability of weak supervisors to elicit knowledge from strong models. With much more progress in this direction, we could get to the point where we can use weak supervisors to reliably elicit knowledge from much stronger models, at least for some key tasks that we care about. This may allow us to develop superhuman reward models or safety classifiers, which we could in turn use to align superhuman models.

Aligning superhuman models is essential for making them safe; there is increasing recognition that failing to align such powerful models has the potential to be catastrophic, making this one of the most important unsolved technical problems in the world (CAIS, 2022). We think it is now more tractable than ever to make rapid iterative empirical progress toward solving this problem.

2 Related Work

We study how we can leverage the generalization properties of deep neural networks to solve weak-to-strong learning. Our problem setting and methods are closely connected to many existing research areas.

Weakly-supervised learning. Weak-to-strong learning is a special type of weakly supervised learning—a setting in which models are trained using unreliable labels (Bach et al., 2017; Ratner et al., 2017; Guo et al., 2018). There is also a rich literature on the related problem of learning from noisy labels (Song et al., 2022). Common methods include bootstrapping (Reed et al., 2014; Han et al., 2018; Li et al., 2020), noise-robust losses (Zhang & Sabuncu, 2018; Hendrycks et al., 2018; Ma et al., 2020), and noise modeling (Yi & Wu, 2019). Unlike most work on label noise, the errors in our weak supervision are much harder to address than uniform label noise, instead having "instance-dependent" errors (Frénay & Verleysen, 2013). Semi-supervised learning, in which labels are only available for a subset of the data, is also closely related (Kingma et al., 2014; Laine & Aila, 2016; Berthelot et al., 2019). We could also study our problem in a semi-supervised setting by having an "easy" subset of examples that weak supervisors provide reliable labels for and a subset of unlabeled "hard" examples that the weak supervisor can't reliably label, a problem which we call "easy-to-hard generalization" (see Appendix C).

Student-teacher training. The framework of first training a teacher and then training a student on teacher's pseudo-labels is widely used in semi-supervised learning (Laine & Aila, 2016; Tarvainen & Valpola, 2017; Xie et al., 2020), domain adaptation (French et al., 2017; Shu et al., 2018), and knowledge distillation (Hinton et al., 2015; Gou et al., 2021; Stanton et al., 2021; Beyer et al., 2022). In contrast to most prior work, we focus on the setting where the student is much more capable than the teacher.

Furlanello et al. (2018) and Xie et al. (2020) also consider cases where the student is at least as capable as the teacher. However in their settings the student is randomly initialized and has access to ground truth labels. Moreover, compared to most past work we are focused on qualitatively *very* weak supervision. For example, we are interested in huge leaps in generalization, similar to going from "3rd grade-level" supervisors to "12th grade-level" student models. Despite these differences with past work, we expect many methods from semi-supervised learning and domain adaptation to translate to our setting. For example, we found that a type of confidence auxiliary loss similar to past work (Grandvalet & Bengio, 2004) improves weak-to-strong generalization in Section 4.3.

Robustness of pretraining and finetuning. Many papers have shown that pretraining on massive, diverse data leads to more robust representations that generalize better out-of-distribution (Hendrycks et al., 2019; 2020b; Radford et al., 2021; Liu et al., 2022). Finetuning typically improves in-distribution generalization, but often performs poorly out-of-distribution, sometimes even degrading performance relative to zero-shot prompting (Kumar et al., 2022; Wortsman et al., 2022b; Awadalla et al., 2022). Recent approaches to mitigating this problem include weight ensembling (Wortsman et al., 2022b;a), finetuning only a subset of layers (Kirichenko et al., 2023; Lee et al., 2022a), or mitigating the distortion effects that finetuning has on pretrained features (Kumar et al., 2022). We did not find strong results in preliminary explorations of approaches similar to these (Appendix B), but we expect that with more thorough explorations one may be able to attain much stronger results with these or other ideas from the robust finetuning literature.

Debiasing. In weak-to-strong generalization, the weak labels contain a specific form of bias, which results from the weak models' lack of capability. There is a substantial literature on learning from biased training data (Bellamy et al., 2018). However, most work focuses on *known* biases, for example where we know that the models perform worse on minority groups. For known biases, common methods include Group Distributionally Robust Optimization (Sagawa et al., 2019), adversarial training (Zhang et al., 2018), and model editing (Santurkar et al., 2021; Meng et al., 2022). In contrast, our setting can be viewed as a particularly difficult debiasing problem where the bias is unknown. Some methods that automatically discover and mitigate biases include clustering (Sohoni et al., 2020), loss variance reduction (Khani et al., 2019), and auditing and re-training on high-loss group (Kim et al., 2019; Liu et al., 2021).

Imitation and preference learning. The goal of alignment is to steer already-capable models to do what we want them to do. For example, the base GPT-4 model is good at generating text following its pretraining distribution, but does not readily follow instructions. To align pretrained language models today, we finetune them using imitation learning on human demonstrations (Bain & Sammut, 1995; Atkeson & Schaal, 1997) or by using methods such as reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Glaese et al., 2022; Bai et al., 2022a). Constitutional AI (Bai et al., 2022b; Lee et al., 2023) leverages AI feedback to align language models, but still uses an initial RLHF phase. However, both imitation learning and preference learning assume high-quality human supervision, making it unclear if they will work for superhuman models.

Scalable oversight. Scalable oversight techniques aim to improve the ability of humans to supervise models. For example, humans may ask models to critique the outputs of other models (Irving et al., 2018; Saunders et al., 2022) or use models to help decompose a problem into simpler subproblems (Leike et al., 2018; Christiano et al., 2018; Lightman et al., 2023). Scalable oversight methods typically take advantage of special problem structure, like decomposability or the fact that evaluation is easier than generation. In contrast to improving human supervision, we focus on generalizing beyond human supervision such that models perform well even in settings we cannot reliably supervise. That said, our weak-to-strong learning setup can be used to compare scalable oversight methods, generalization-based methods, and more. Our setup also resembles a proposal for measuring progress on scalable oversight known as "sandwiching", which uses weak and strong humans (Cotra, 2021; Bowman, 2022).

Knowledge elicitation and honesty. Christiano et al. (2022) introduced a theoretical problem called Eliciting Latent Knowledge (ELK), in which the goal is to elicit latent knowledge from a superhuman machine learning model even under worst case assumptions. For example, a special case of ELK is honesty (Evans et al., 2021), where the goal is for the models to report their true beliefs[2]. Wentworth (2020) hypothesizes a tendency for neural networks to develop "natural abstractions" that are easier to elicit. Recent empirical work on ELK includes a benchmark for measurement tampering (Roger et al., 2023), methods for discovering latent knowledge (Burns et al., 2023), and studies of honesty (Li et al., 2023; Pacchiardi et al., 2023). Our setting can be viewed as a general methodology for empirically studying problems like ELK and honesty across a wide range of tasks.

3 Methodology

A core challenge of superalignment is that humans will need to supervise models much smarter than us. This is a special case of what we call the *weak-to-strong learning problem*: how can a weak supervisor oversee a model much smarter than it? In this paper, we study a simple analogy, in which we replace the weak human supervisor with a weak model supervisor.

For a given task of interest, consisting of a dataset and a performance metric, we:

1. **Create the weak supervisor.** Throughout most of this work, we create weak supervisors by finetuning small pretrained models on ground truth labels.[3] We call the performance of the weak supervisor the *weak performance*, and we generate *weak labels* by taking the weak model's predictions on a held-out set of examples.
2. **Train a strong student model with weak supervision.** We finetune a strong model with the generated weak labels. We call this model the *strong student model* and its resulting performance the *weak-to-strong performance*.

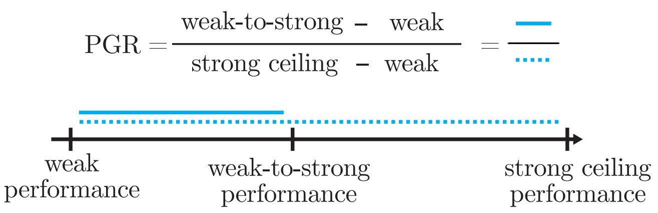
[2] Like Evans et al. (2021), we define *honesty* to mean a model reporting what it *believes* to be true, in contrast to truthfulness which asks whether what a model reports *is* true.

[3] In Appendix D and Appendix E we study other synthetic weak supervisors. Future work could test many more sources of weak supervision, such as by having 3rd grader humans provide labels.

3. **Train a strong model with ground truth labels as a ceiling.** Finally, for comparison, we finetune a strong model with ground truth labels.[4] We call this model's resulting performance the *strong ceiling performance*. Intuitively, this should correspond to "everything the strong model knows," i.e. the strong model applying its full capabilities to the task.

For more details on how we train each model, see Appendix A.

Typically, weak-to-strong performance will be between weak performance and strong ceiling performance. We define the **performance gap recovered (PGR)** as a function of the above three performances (weak, weak-to-strong, and strong ceiling) as shown in the illustration below.



PGR measures the fraction of the performance gap (the difference in performance between the weak and strong ceiling models) that we can recover with weak supervision. If we achieve perfect weak-to-strong generalization, PGR is 1. If the weak-to-strong model does no better than the weak supervisor, then PGR is 0.

Advantages. Our setup has a number of advantages, including:

1. It can be studied with any pair of weak and strong models, making it easy to study scaling laws and not requiring access to expensive state-of-the-art models. Moreover, it does not require working with humans, so feedback loops are fast.
2. It can be studied for any task of interest, making it easy to empirically test across a wide range of settings.
3. Success will be practically useful even before we develop superhuman models: for example, if we find ways to align GPT-4 with only weak human supervision or with only GPT-3-level supervision, that would make it more convenient to align models today.

Limitations. Our setup still has important disanalogies to the ultimate problem of aligning superhuman models. We view our setup as removing one of the main disanalogies in prior work, not as providing a final, perfectly analogous setup. Two remaining disanalogies include:

1. **Imitation saliency.** Future superhuman models will likely have salient representations of human behaviors, but our strong models may not have learned features relevant for imitating weak model predictions; simply imitating the weak supervisor may thus be an easier failure mode to avoid in our setting than it will be in the future. More generally, the types of errors weak models make today may be different from the types of errors humans will make when attempting to supervise superhuman models.
2. **Pretraining leakage.** Our pretraining data implicitly contains supervision from humans. It may thus be artificially easy to elicit strong models' capabilities in our setting, since they were directly pretrained to observe strong (human-level) performance. Superhuman-level performance may not be directly observed in the same way—superhuman knowledge might be more latent, e.g. because it was learned from self-supervised learning—and thus might be harder to elicit from superhuman models in the future.

[4] For tasks solved by superhuman models that humans cannot evaluate, we will not have access to ground truth labels. However, we allow access to ground truth labels in our experimental setting today for scientific and evaluation purposes. Note that we evaluated weak-to-strong performance against ground truth many times while iterating on methods; however, we held out our largest model (GPT-4) and about half of NLP tasks throughout the project.

More generally, we do not yet know how superhuman models will be built, but they could develop new inductive biases that are qualitatively different from today's models. We view iterating on our methodology to produce even more analogous setups as a key priority for future work, as we discuss in more detail in Section 6.

4 Main Results

In this section, we report our main empirical results, including baselines and promising methods.

4.1 Tasks

Popular natural language processing benchmarks. We consider 22 popular NLP classification datasets covering ethics, commonsense reasoning, natural language inference, sentiment analysis, and other domains. We convert all datasets to binary classification tasks and approximately balance the classes. We produce soft labels from the weak model. See a full list of the datasets and their sources in Table 1.

Chess puzzles. We use the dataset originally introduced in Schwarzschild et al. (2021b), which contains chess puzzles from the `lichess.org` website (Lichess Team, 2023). Each puzzle consists of a chess position, and a sequence of optimal moves to play to solve the puzzle. For our evaluation, we predict the first move played, which is the best move in the given chess position. We illustrate the data format in Appendix Figure 14. For weak labels, we sample from the weak model with temperature 0. Note that unlike the other binary classification tasks we study in this paper, this is a generative task.

ChatGPT reward modeling. The standard approach to aligning models today is reinforcement learning from human feedback (RLHF). A critical step of RLHF is to train a reward model (RM) to predict human preferences between model responses. Specifically, a reward model is trained on a dataset consisting of dialogs between a human and an assistant model. For each query, the humans compare multiple possible responses (completions) from the assistant, providing human preference data. Then, a reward model is trained to predict the results of pairwise comparisons between completions. Finally, the assistant model is trained by optimizing against the reward model with reinforcement learning (RL). In our work, we do not study the RL step, and instead assume the goal is to maximize reward model accuracy. For more details on reward models, see e.g. Ouyang et al. (2022). We use a proprietary dataset used to train ChatGPT reward models.

For more details about our tasks and setup, see Appendix A.

4.2 Naively finetuning on weak labels

In each of these 3 settings (NLP tasks, chess puzzles, and reward modeling) we evaluate how well strong students generalize when naively finetuned on labels generated by weak supervisors. We study pretrained language models from the GPT-4 family (OpenAI, 2023), which allow us to study student-supervisor compute disparities of many orders of magnitude. We find that PGRs are almost universally positive—in virtually all settings that we studied, and across almost all student and supervisor sizes, students outperform their supervisors (Figure 3).

On the popular NLP benchmarks, we find especially promising weak-to-strong generalization: strong models trained with weak supervision can often generalize to a substantially higher performance than the weak model itself. Even with very weak supervisors and strong models with many orders of magnitude more compute, we recover more than 20% of the performance gap. The PGR increases both with weak supervisor size and with strong student size; for the largest students, the PGR is often above 50%.

We see more mixed results in the chess puzzle setting. In particular, when using the smallest weak models, the PGR is close to zero and the test accuracy curves appear flat. However, as the size of the weak supervisor increases, the PGR increases substantially; for small supervisor-student gaps, PGR can be above 40%. Unlike in the NLP setting, where PGR improves with the strong student size, PGR *decreases* with the strong student size for a given weak supervisor on chess puzzles. The cor-

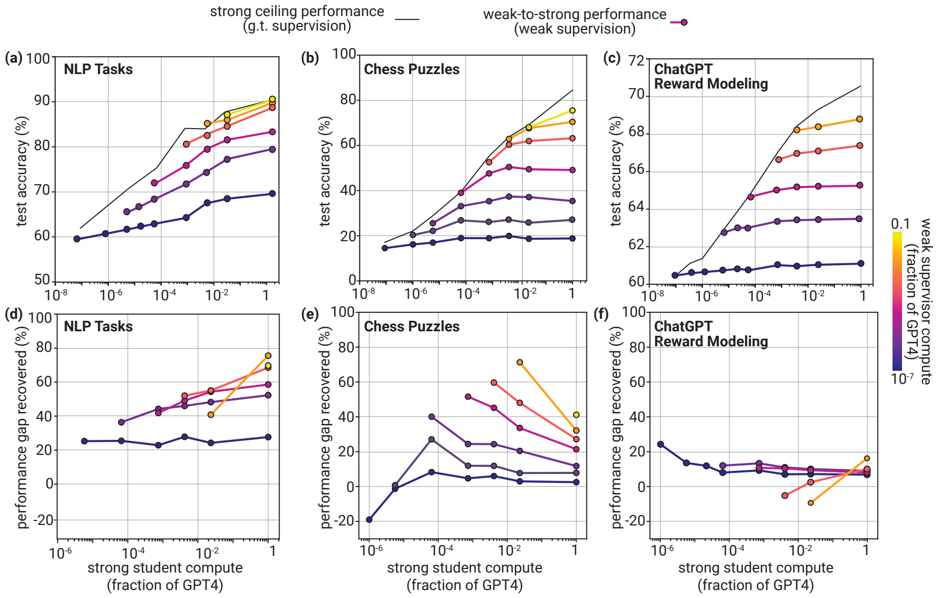


Figure 3: **Promising weak-to-strong generalization with naive finetuning on NLP tasks and chess, but poor generalization on the ChatGPT reward modeling task. (a,b,c)** Test accuracy as a function of strong student size on **(a)** NLP tasks, **(b)** chess puzzles, and **(c)** the ChatGPT reward modeling task. Accuracy of strong students trained with ground truth in black, accuracy of strong students trained with weak supervision shown with colored lines (hue indicates size of weak supervisor). **(d,e,f)** Same as panels **a,b,c** but for performance gap recovered (see Section 3 for details). For NLP settings, we compute the median across tasks (see Figure 12 for full details). We find decent weak-to-strong generalization and even positive PGR scaling on NLP tasks, decent generalization for small supervisor-student gaps but negative PGR scaling on chess puzzles, and both poor generalization and scaling for ChatGPT reward modeling.

responding test accuracy curves appear concave, potentially exhibiting inverse scaling (McKenzie et al., 2023) in strong student size.

Finally, we find that weak-to-strong generalization is poor by default in the ChatGPT reward model setting. We are usually only able to recover roughly 10% of the performance gap between the weak supervisor and the strong student. Even for relatively small gaps in compute between the weak and strong models, PGR almost never exceeds 20%.

In general, across all our settings, we observe weak-to-strong generalization: strong students consistently outperform their weak supervisors. It is not obvious why this should happen at all—especially from naive finetuning alone—and it gives us hope that weak-to-strong learning is a tractable problem. At the same time, our results suggest that naively using weak, human-level supervision will be insufficient to align strong, superhuman models; we will need qualitatively new techniques to solve superalignment.

4.3 Improving Weak-to-Strong Generalization is Tractable

We now show that we can use simple methods to substantially improve weak-to-strong generalization. While none of the methods we test works universally, these methods are proofs-of-concept that across many different tasks we can substantially improve generalization.

4.3.1 Bootstrapping with intermediate model sizes

Bootstrapping is a long-standing idea in alignment: instead of directly aligning very superhuman models, we could first align an only slightly superhuman model, use that to align an even smarter model, and so on (Christiano, 2019; 2018; Leike & Sutskever, 2023; Worley, 2021). Our setting allows us to empirically test this idea.

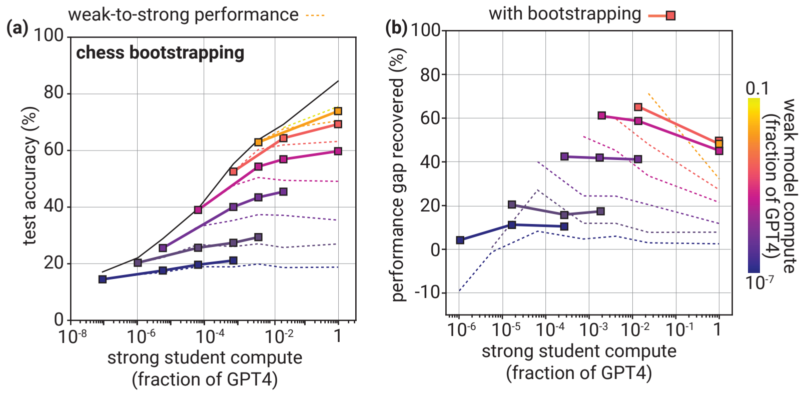


Figure 4: **Bootstrapping improves weak-to-strong generalization on chess puzzles. (a)** Test accuracy as a function of strong student size. Accuracy of students trained with ground truth in black, accuracy of students naively trained with weak supervision shown with dotted lines (hue indicates size of weak supervisor). Accuracies of students trained via bootstrapping shown with colored squares (including both the final weak-to-strong performance and the performance of the intermediate models during bootstrapping). **(b)** Same as **a** with PGR. By taking multiple small steps instead of one big step we see substantially improved generalization, especially for larger student models.

Specifically, we can construct a sequence of model sizes $\mathcal{M}_1 \to \mathcal{M}_2 \to \ldots \to \mathcal{M}_n$ of increasing sizes. Then, we use the weak labels from $\mathcal{M}_1$ to finetune $\mathcal{M}_2$, use $\mathcal{M}_2$ to generate new weak labels that we can use to finetune the next model in the sequence, $\mathcal{M}_3$, and so on.

We evaluate bootstrapping in the chess puzzle setting. When we naively finetune on weak labels for chess (Section 4.2), we see high PGR when we cross small supervisor-student gaps, but low PGR for larger gaps. As a result, in this setting it may help to take multiple small steps—steps where PGR should be high—instead of one big step.

For each round of bootstrapping, we run three iterations of weak-to-strong learning, i.e. we bootstrap the weak supervision using two intermediate model sizes before finally finetuning the largest model in the sequence. We report the results (including all intermediate weak-to-strong models within each bootstrap) in Figure 4. Bootstrapping improves PGR compared to the baseline, especially for larger student models. With the naive method, transfer accuracy curves flatten as the weak-strong gap grows larger; with bootstrapping, the accuracy continues to monotonically improve.

While the results in the chess setting are promising, in preliminary experiments we observed only small improvements with bootstrapping on NLP tasks and no improvements in the RM setting. This makes sense intuitively: unlike in the chess setting where naive PGR decreased with larger supervisor-student gaps, naive PGR increased or was rougly constant for larger supervisor-student gaps in the NLP and reward modeling settings. Overall, these results suggest bootstrapping is a plausible avenue to investigate for improving weak-to-strong generalization and can be helpful in some settings, but that naive bootstrapping alone will not be enough to align models much smarter than their supervisors.

4.3.2 An auxiliary confidence loss can dramatically improve generalization on NLP tasks

In our baseline results (Section 4.2), we naively finetune the strong student on the labels provided by the weak supervisor. Because we are directly training the strong student to imitate the weak supervisor, it may also learn to imitate the errors of the supervisor (see Section 5.1 for more discussion). Intuitively, we want to avoid this failure mode and provide additional regularization towards what the strong pretrained model already internally knows: we want the student to learn the intent of the supervisor, but not to imitate its mistakes.

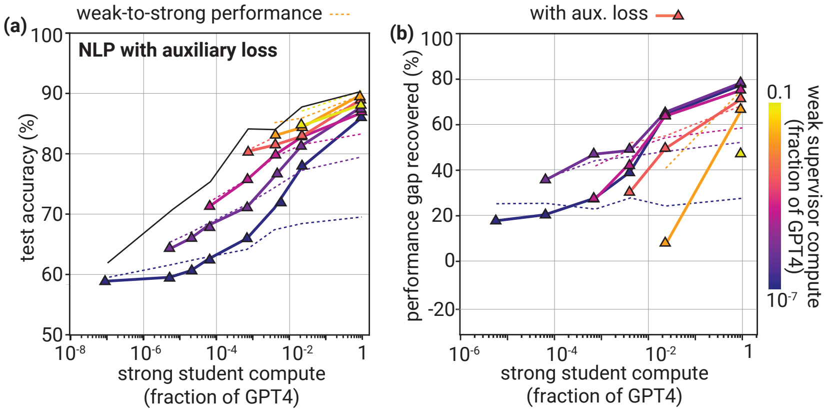


Figure 5: **Substantially improved generalization on NLP datasets with a simple auxiliary loss.** **(a)** Test accuracy as a function of strong student size. Accuracy of a student trained with ground truth in black, accuracy of students naively trained with weak supervision shown with dotted lines. Accuracies of students trained with auxiliary confidence loss shown with colored triangles. Median computed across 22 NLP tasks (hue indicates size of weak supervisor), see Figure 6 for individual datasets. **(b)** Same as **a** with PGR. The confidence loss can improve generalization drastically, especially for large supervisor-student gaps.

We operationalize this intuition by adding an auxiliary confidence loss term to the standard cross entropy objective. This method is closely related to conditional entropy minimization (Grandvalet & Bengio, 2004) which is a prominent technique in semi-supervised learning. Specifically, we add an additional loss term which reinforces the strong model's confidence in its own predictions—even when they disagree with the weak labels. We provide a detailed description of the method in Appendix A.4.

In Figure 5, we plot accuracy and PGR curves with this method on our NLP tasks. We find that while it performs slightly worse than the naive baseline for smaller strong students, it dramatically improves generalization for large gaps in compute between weak and strong models. With the smallest weak supervisor and largest strong student, the confidence loss increases median PGR from about 25% to nearly 80%.

In addition, we also plot generalization curves for a representative subset of NLP datasets in Figure 6, as well as the full panel of datasets in Figure 12. There are some settings in which the confidence loss does not help much or degrades performance, e.g. when the gap between the weak supervisor and strong student is small or when the dataset features inverse scaling even with ground truth supervision. But the confidence loss improves performance on most NLP datasets dramatically, and for many datasets we get almost perfect generalization, recovering nearly all the performance of the strong model, even when using the smallest weak supervisors.

Finally, we find evidence consistent with our motivating intuition for the confidence loss (allowing the strong student to confidently disagree with its weak supervisor): the auxiliary loss reduces the strong student's imitation of weak errors and mitigates weak label overfitting (see Section 5.1).

5 Understanding Weak-to-Strong Generalization

Strong methods will be essential for solving superalignment, but to trust those methods it is also important to understand *when* and *why* they work. A better understanding of weak-to-strong generalization could help us trust that generalization will continue working even in the future high-stakes settings we care most about, and could help us develop better methods along the way. In this section, we study two phenomena relevant to weak-to-strong generalization: imitation of supervisor mistakes and salience of the tasks to the strong student model.

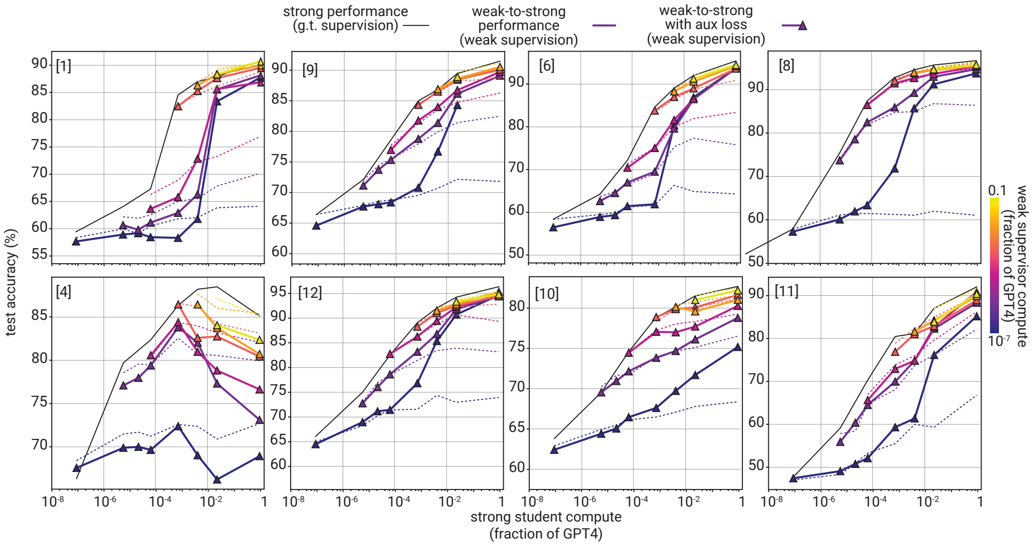


Figure 6: **Simple auxiliary loss improves generalization across most datasets.** Test accuracy as a function of strong student compute for a representative sample of NLP tasks. See Table 1 for dataset details and Appendix Figure 12 for results on all 22 NLP tasks. Auxiliary loss is shown with triangles, and the baseline with dotted lines. Weak supervisor model size shown in varying colors, with ground truth supervision shown in black.

5.1 Understanding imitation

When we train a strong model with weak supervision on some task, our hope is that the strong model will perform that desired task as well as possible, leveraging the latent capabilities it learned from pretraining to significantly outperform the weak supervisor. A salient way in which we could fail to achieve that desired generalization is if the strong model instead learns to imitate the weak supervisor—predicting how the weak supervisor would have classified each example. In particular, if the weak labels contain systematic errors that are easy to learn, the strong model could learn to imitate those errors. This is also a concern raised in theoretical work on superalignment, which has argued that the *human simulator* failure mode could be important: naive human supervision might result in superhuman models learning to imitate what a human would say, rather outputting its best predictions (Christiano et al., 2022).

5.1.1 Overfitting to Weak Supervision

The failure mode of imitating weak supervision is especially relevant to our naive baseline in Section 4.2, which directly trains the student to imitate the supervisor. In the case of infinite training data, naively fitting the weak labels should result in perfect imitation, and a PGR of zero. In practice, we train on finite data for a small number of epochs. Unlike typical ML settings, however, we could expect to observe overfitting even when training for less than a single epoch: the strong model might overfit to the weak supervisor labels and its errors, degrading ground truth test accuracy over training even without classic overfitting to any specific training examples.

Empirically, we see that the strong student indeed appears to overfit to the weak supervisor's errors. In Figure 7(a) we show ground truth test accuracy curves over the course of training for the ChatGPT RM task, and in Figure 7(b) and (c) we compare the best[5] and final ground truth test accuracies (median across all weak-strong model pairs). We find overfitting for large weak-strong gaps. For small weak-strong gaps, weak-to-strong performance typically monotonically increases over the course of training. For larger gaps, weak-to-strong performance often increases initially, but then starts dropping well before a single epoch has elapsed. Ground truth early stopping, which "cheats"

[5] Note that our best test accuracies may slightly overstate accuracy, due to noisy evaluations.

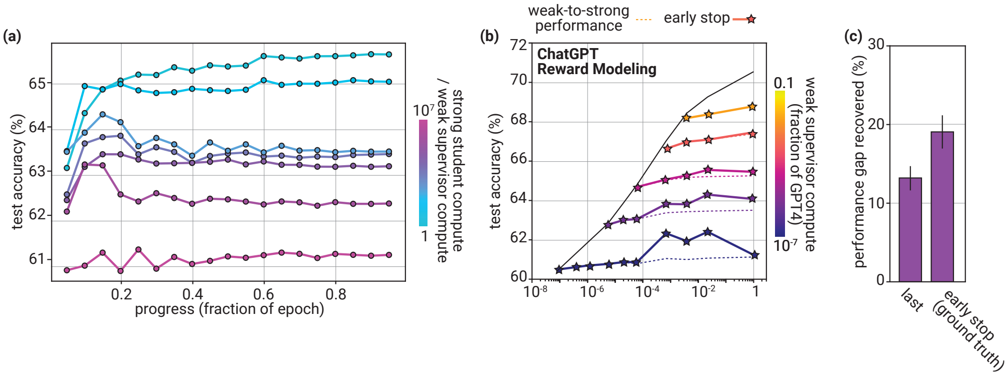


Figure 7: **Strong models overfit to the weak labels.** In all figures, we show data for the ChatGPT Reward Modeling task. **(a)** Weak-to-strong performance over the course of training. Hues indicate the student-supervisor gap. (**b**) Best weak-to-strong performance during training (stars) and weak-to-strong performance at the end of training (dashed). Weak performance in black. Hue indicates the size of the weak supervisor. (**c**) Median best and final performance gap recovered (PGR) aggregated across all supervisor-student pairs. We see overfitting to weak labels for large weak-strong gaps, even within one epoch. In these cases, the best test accuracy achieved over training can be substantially better than the test accuracy at the end of training. See Figure 13 for the corresponding analysis of a representative subset of NLP tasks.

by evaluating against ground truth and stopping at an optimal step with respect to ground truth test labels, typically gives a PGR improvement of around 5 percentage points.

We see the same phenomenon for NLP tasks in Figure 13. In the NLP setting, we find that "cheating" early stopping on ground truth gives a 15 percentage point boost in PGR over the model at the end of training, and a 10 percentage point boost in PGR compared to "non-cheating" early stopping with respect to weak labels.

Unfortunately, an early stopping criterion that uses ground truth labels does not constitute a valid method. Nevertheless, the results above suggest that imitating weak supervisor errors may be an important phenomenon in our setting.

Moreover, these results suggest that better early stopping or regularization strategies may be able to substantially improve weak-to-strong generalization, by reducing overfitting to the weak labels and their errors. Indeed, we see in Figure 13 that the auxiliary confidence loss introduced in Section 4.3.2 reduces overfitting to weak labels on NLP tasks substantially. For large weak-strong gaps, early stopping on ground truth (compared to early stopping on weak labels) gives a 15% PGR boost when using the naive method, but only a roughly 5% PGR boost when using the confidence loss.

5.1.2 Student-supervisor agreement

Another way to measure imitation is to directly measure the agreement between the student and the supervisor: the fraction of test inputs where the strong student makes the same prediction as the weak supervisor. Note that if agreement were 100%, then weak-to-strong accuracy would be equal to supervisor accuracy, and PGR would be 0.

In general, we notice that for our naive finetuning baseline, student-supervisor agreement is consistently high—often noticeably higher than weak supervisor accuracy. This indicates that the student is imitating some of the supervisor's errors. These phenomena hold across all tasks (NLP tasks, chess, and reward modeling) and all model sizes, for the naive method.

The confidence loss in Section 4.3.2 reduces student-supervisor agreements significantly (Figure 8), primarily by imitating supervisor mistakes less (Figure 8c). The loss encourages the strong student to make confident predictions, including when they contradict the weak supervisor. In a handful of the settings where it is most successful, the confidence loss reduces student-supervisor agreement

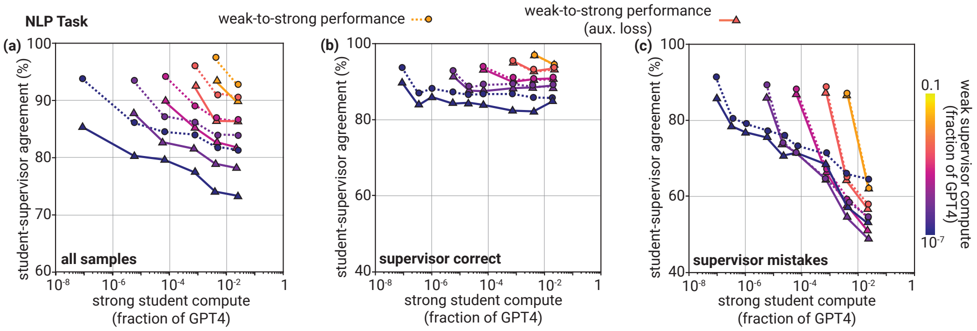


Figure 8: **Student-supervisor agreement *decreases* with larger student-supervisor gaps; the confidence loss reduces imitation of supervisor mistakes.** **(a)** Student-supervisor agreement as a function of strong student size on NLP tasks, **(b) a** but only on samples where the supervisor is correct, **(c) a** but only on samples where the supervisor is mistaken. Dotted lines indicate naive finetuning on weak labels, and triangles indicate results with the auxiliary confidence loss results (see Section 4.3). Hue of line indicates size of weak supervisor. For results on reward models, see Figure 16.

below strong student test accuracy (weak-to-strong performance)—i.e., the resulting model is fitting the ground truth concept *better* than it is fitting the weak labels it was trained with.

5.1.3 Inverse scaling for imitating the supervisor

Next, we study student-supervisor agreement as a function strong model size (see Figure 8 and Figure 16). Surprisingly, we find inverse scaling (McKenzie et al., 2023): larger student models consistently agree *less* with the errors of the supervisor than smaller student models, despite being trained to imitate the supervisor, not using early stopping, and having larger capacity than smaller student models.

This trend is especially strong if we evaluate agreement only on datapoints where the supervisor is wrong (Figure 8c), and the trend persists if looking at cross entropy loss instead of accuracy.

These results suggest that pretrained models may have a hard time fitting errors of other (smaller) pretrained models, at least in finetuning settings with relatively limited data. Stanton et al. (2021) and Furlanello et al. (2018) report a related observation in the context of knowledge distillation: it is surprisingly hard for models to fit the predictions of other models, even when they have sufficient capacity to do so.

One natural hypothesis is that the nature of (especially naive) weak-to-strong generalization depends heavily on the error structure of the weak supervisors and how easy those errors are to imitate. In Appendix E, we show initial experiments that test how different types of weak supervision errors impact what the strong student learns. Our results suggest that errors that are more difficult for the student to imitate result in stronger naive weak-to-strong generalization, but that even when they are easy to imitate, the confidence loss can help.

5.2 Saliency in the strong model representations

One intuition for when weak-to-strong generalization might be feasible is when the task or concept we want to elicit is internally "salient" to the strong model. In this section, we study several phenomena related to the saliency of the concepts we are trying to elicit from the student model.

5.2.1 Eliciting strong model knowledge with prompting

One possible reason for the high PGR we observe in Section 4 could be that eliciting what the strong model knows is easy. In particular, it is possible that strong pretrained models can solve many relevant tasks zero-shot with a simple prompt.

In Figure 9a, we consider 7 representative NLP tasks and compare finetuning, zero-shot prompting, and 5-shot prompting; for this initial experiment, we use ground truth labels rather than weak labels

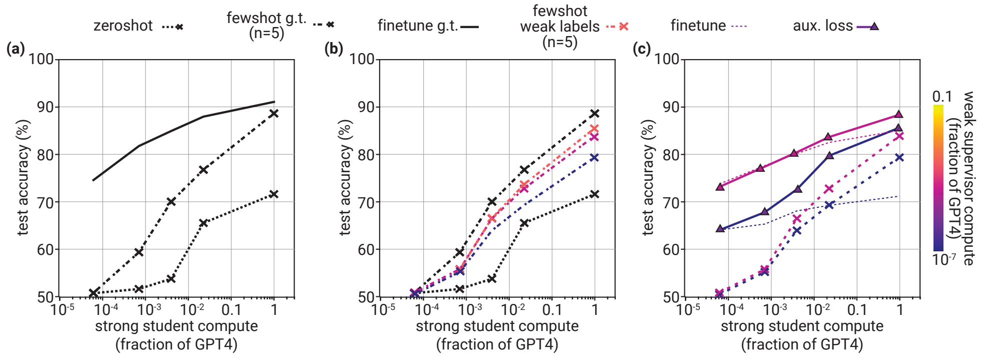


Figure 9: **Few-shot prompting becomes competitive with finetuning for large models; weak-to-strong learning is qualitatively similar in the prompting setting.** **(a)** Average zero-shot (single dashed), 5-shot (double dashed) and finetuning (solid) accuracy with ground truth labels as a function of strong student size. **(b)** Average 5-shot with weak labels (colored dashed) accuracy as a function of student model size. Hue of line indicates size of weak supervisor. Zero-shot and 5-shot same as in panel **a**. **(c)** Average weak-to-strong performance for 5-shot prompting (dashed with crosses), naive finetuning (dashed thin) and finetuning with the confidence loss (solid with triangle) as a function of student model compute. Results are averaged across 7 NLP tasks. Few-shot weak-to-strong performance becomes competitive with or outperforms finetuning for the largest strong students, though finetuning with the confidence loss does better.

for finetuning and 5-shot. For both the zero-shot and 5-shot baseline we use task-specific prompts summarized in Table 2. We find that zero-shot and 5-shot test accuracy is poor for most model sizes but, consistent with Brown et al. (2020), improves drastically for larger model sizes. In particular, for the largest models, 5-shot prompting becomes competitive with finetuning on many tasks, indicating that eliciting the task-relevant knowledge of these very large models is relatively straightforward.

We are also interested in weak-to-strong learning in the context of few-shot prompting. To study this setting, we construct a few-shot prompt where the labels are provided by the weak supervisor. We report the results in Figure 9b. Consistent with our findings in the finetuning setting, we get worse performance when we few-shot prompt with weak labels than we do few-shot prompting with ground truth labels. This suggests that weak-to-strong learning is a nontrivial problem in the prompting setting as well.

Similar to the finetuning setting, few-shot weak-to-strong performance improves for stronger supervisors. Compared to our weak-to-strong finetuning baseline (Figure 9c), weak-to-strong performance of few-shot prompting is poor for smaller student models, but becomes competitive or even outperforms finetuning for the largest strong students. However, weak-to-strong finetuning with the confidence loss still generally outperforms weak-to-strong few-shot prompting.

Overall, these results provide an important reference for our results on weak-to-strong generalization. They suggest that for the largest model sizes, the knowledge needed to solve many task can be elicited fairly easily with prompting. However, our current setup may be more disanalogous for prompting than for finetuning; many of our NLP tasks may have been implicitly observed during pretraining, which we conjecture benefits prompting more than finetuning. We discuss this potential disanalogy much more in Section 6.1.

5.2.2 Generative supervision improves RM weak-to-strong generalization

If salient representations of the desired task is useful for weak-to-strong generalization, then we may be able to improve generalization by increasing the salience of the task to the strong model. One way to increase the salience of a task without needing ground truth labels is to perform unsupervised finetuning with the language modeling objective on data relevant to that task (Dai & Le, 2015). For example, by finetuning a language model in an unsupervised way on online reviews, sentiment becomes saliently represented to models internally (Radford et al., 2017).

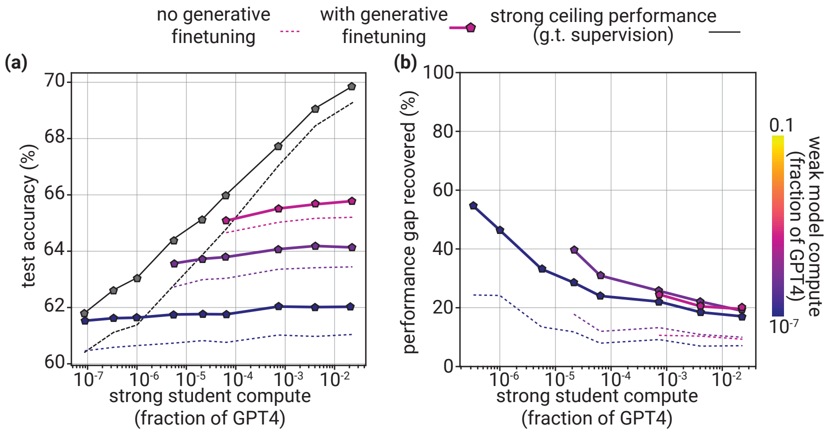


Figure 10: **Generative finetuning on reward modeling data improves weak-to-strong performance and PGR. (a)** Weak-to-strong performance on the reward modeling task, with (solid lines) and without (dashed lines) an extra step of generative finetuning for the strong student model. Solid black line shows a strong ceiling reward model that was also trained with the generative finetuning step; dashed black line show a weak supervisor reward model trained *without* the generative finetuning step. **(b)** PGR with and without generative finetuning. For generative finetuning PGR, we use the strong ceiling performance that also had this extra generative finetuning step. Even with this ceiling adjustment, PGR is higher with an extra generative finetuning step.

We test this idea in our reward modeling setting, where it is standard practice to initialize the model with a baseline finetuned on demonstrations of desired behaviors (Stiennon et al., 2020). In our case, we re-use the ChatGPT comparison data instead of introducing a new supervision dataset. Comparisons are comprised of a prefix (a single request or conversation between the user and assistant) and at least two candidate completions. We finetune the base models with a language modeling loss on all prefix-completion pairs, ignoring the human preferences between those completions.

Note that these pairs include completions ranked worst by human raters, so this procedure should not in principle leak any information about the ground truth preference labels that the weak-to-strong models should not have access to. On the other hand, since the completions can come from humans or stronger models, there may be some leakage similar in kind to the pretraining leakage that we discuss as a disanalogy in Section 6.1. Even in this setup, the reward modeling task is highly nontrivial, and we leave addressing this disanalogy (e.g. by collecting completions only from weaker models) for future work.

We found that the additional generative finetuning on the RM data leads to better weak-to-strong performance. Because this procedure also improves the performance of models trained on ground truth RM data, we compare our new weak-to-strong performance to strong "ceiling" models that were also first generatively finetuned in the same way. Even with this adjusted ceiling, we find that generative supervision improves PGR by approximately 10-20%. We report the results in Figure 10.

Furthermore, the improvement from generative finetuning stacks with the improvement from ground truth early-stopping (a "cheating" method to illustrate potential performance if we could optimally early stop, see Section 5.1.1). When we combine these two techniques, we can achieve PGR of approximately 30-40%, which would make the results on the RM task competitive with the weak-to-strong generalization we observe on NLP and chess puzzle tasks.

We can apply the idea of improving task saliency with generative finetuning on relevant data to all settings, and we believe this could be a promising direction for future work.

5.2.3 Finetuning on weak supervision to increase concept saliency

One possible measure of concept saliency is how linearly represented a task is. In particular, we can measure the performance of a linear probe (logistic regression classifier) trained from frozen activations of the model. If the optimal solution can be approximately recovered with a linear probe, that

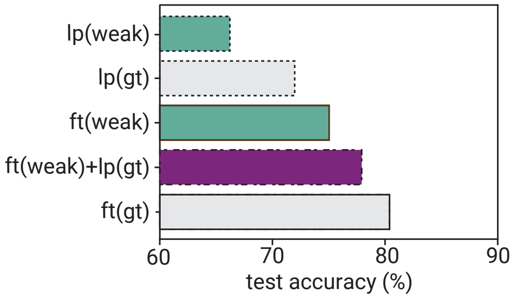


Figure 11: **Finetuning on weak supervisor labels makes the desired generalization more linearly represented.** We plot test accuracy for five different strategies, averaged across a subset of NLP tasks. **lp(weak)**: training a linear probe on the base model using weak labels, **lp(gt)**: training a linear probe on the base models using ground truth labels, **ft(weak)**: finetuning the model on weak labels, **ft(weak) + lp(gt)**: finetuning the model on weak labels *then* training a linear probe on ground truth labels, **ft(gt)**: finetuning the model on ground truth labels. Finetuning on the weak labels significantly increases the linearity of the ground truth concept.

could simplify our problem greatly; we could focus on linear probing methods instead of finetuning methods, which could greatly reduce the search space we need to consider to elicit the desired generalization. In our work, we focus only on how linearly represented a task is in the final activations, prior to the unembedding layer.

In Figure 11, we plot average test accuracy on a subset of our NLP datasets for several different combinations of (1) finetuning or linear probing, using (2) weak or ground truth labels. First, we show linear probes trained with ground truth labels (72% accuracy on average) perform worse than finetuning with ground truth labels (82% on average), indicating that the optimal solution to most tasks is *not* represented completely linearly in the strong model's final activations. For comparison, we also report the results for linear probing and finetuning using weak labels, which we verify are worse than using ground-truth labels.

However, we find that we can achieve substantially better performance by *first* finetuning the model on the *weak* labels, and *then* linear probing using the *ground truth* labels. In other words, when we finetune the strong model with weak labels, the representations become *more linear even with respect to ground truth labels*. In fact, finetuning on weak labels then linear probing on ground truth labels results in an accuracy of 78%, closing 60% of the gap between ground truth linear probing and finetuning. This also noticeably outperforms the naive weak-to-strong finetuning baseline.

This phenomenon is closely related to a recent finding reported by Kirichenko et al. (2023) in the spurious cues literature. They find that finetuning a model on biased supervision can result in models with very biased outputs, but surprisingly strong linear representations of the desired concepts. These results suggest an alternative approach to improving weak-to-strong generalization. We could first "linearize" the desired concept, e.g. by naively finetuning on weak labels. Then we could use simpler linear probe-based weak-to-strong methods to elicit the desired concept.

6 Discussion

In this paper, we proposed a simple analogy for studying a core challenge of aligning superhuman models and showed that it is feasible to make significant progress on this problem. However, our setup still has important disanalogies, which we now elaborate on. We then outline a number of promising avenues for future work.

6.1 Remaining disanalogies

Imitation saliency: superhuman models may easily imitate weak errors. Future models will likely be very good at predicting what humans will think and say, especially if they are trained on human data in a similar manner to current models. Consequently, if we naively train such a superhuman model with human supervision, it might simply imitate the weak supervisor, outputting human-level capabilities rather than its latent superhuman capabilities (Christiano et al., 2022).

This problem is only partially captured by our setup. While our strong pretrained models do imitate weak supervisors to some extent, they are not explicitly pretrained to imitate weak models, and our results from Section 5.1.3 suggest that larger strong models may even have more difficulty doing this imitation. As such, "imitating the weak supervisor" may not be as much of a problem in our setup as it will be for the ultimate superalignment problem. This may inflate generalization performance today. We believe a more thorough investigation of this problem is an important area for future work.

Pretraining leakage: superhuman knowledge may be latent, not observable. Many of the tasks we consider in this work may have been observed in pretraining at least indirectly, for example through questions on online forums or through slight reframings of the task. For example, it is highly likely that simple science questions similar to those in the SciQ NLP task are present in our GPT-4 series pretraining dataset at least implicitly in some form. However future superhuman models may never directly observe superhuman alignment-relevant capabilities; these capabilities may be predominantly "latent", e.g. learned through self-supervised learning or reinforcement learning rather than through imitation learning. Intuitively, latent capabilities may be harder to elicit than capabilities that models could have observed in their pretraining data.

This disanalogy could cause our results to be overly optimistic. We conjecture that this disanalogy also increases prompting performance (Section 5.2.1) more than it increases finetuning performance; intuitively prompting may work especially well on tasks that the model assigns high probability to observing. If so, this would make prompting more disanalogous in our setup than finetuning. We hope to test this conjecture in future work.

In Appendix D.1, we show a proof of concept that weak-to-strong generalization can still elicit latent capabilities that were never explicitly observed during pretraining, and even when prompting is not possible. In particular, we use AlexNet (Krizhevsky et al., 2012) to supervise models pretrained with DINO (Caron et al., 2021), a self-supervised method in computer vision that learns strong representations. We find that the strong student generalizes significantly beyond AlexNet's performance, even though the student never observed any classification labels during pretraining. Future work should study and mitigate this pretraining leakage disanology more systematically.

6.2 Future Work

What would convince us that we have a "solution" to superalignment? This is a complicated question and we do not claim to have a complete answer. However, we expect substantial progress in at least the following three areas will be necessary: analogous setups, scalable methods, and strong scientific understanding. We now sketch out concrete problems for each of these areas.

6.2.1 Concrete Problems: Analogous Setups

Having strong measurements and a reliable methodology is extremely important for making empirical progress in any field. In particular, it is important that we have metrics which provide strong signal about whether we are making real progress toward the problem we ultimately care about. Important directions for follow-up work include:

- Making our setup more analogous by fixing the main remaining disanalogies described in Section 6.1. Analogous setups are essential to ensure that methods that work today will continue to work for superhuman models.
- Validating that disanalogies are not severe, for example by checking that results are qualitatively similar to using e.g. 3rd grade humans to supervise our strongest models today.
- Relaxing some of the simplifications we made, e.g. by generalizing our methods and results to complicated generative tasks.
- Testing how robust our weak-to-strong classifiers are to optimization pressure when we attain high PGR; for example, if we attain good weak-to-strong generalization with RMs, can we optimize the learned RM using RL?

- Testing our conjecture that prompting-based methods in our current setup will not be as indicative of future results relative to finetuning-based methods (Section 5.2.1), and improvig our setup to fix this.
- Identifying new or more specific disanalogies with our setup and fixing them.

Additionally, we do not yet know what future models will look like. We should update our setup over time as we learn more about how broadly superhuman models will be built.

6.2.2 Concrete Problems: Scalable Methods

One intuition for why major progress on weak-to-strong generalization seems possible is because all we need to do is extract everything the strong model already "knows" about the task of interest—the strong model should intuitively already understand the task, and should hopefully have salient representations of that task. This suggests a number of properties that should be satisfied by the desired generalization, and which we may be able to measure without access to ground truth.

- The desired generalization should be able to *disagree with the weak supervision* when the weak supervision is wrong. This is a property our auxiliary confidence loss may capture.
- The desired generalization should be "*natural*" or "*salient*" to the model. For example, we should not need to change the model too much to elicit the desired concept.
- The desired generalization should be *consistent*. Consistency properties range anywhere from basic logical consistency to complicated forms of consistency between many prompts (e.g. cycle consistency, cross examination, etc.).

Future work should identify additional unsupervised properties that can be used to specify the desired generalization. More generally, there are very likely existing methods in the machine learning literature (e.g. in semi-supervised learning or robust finetuning), which would be natural to try and which could also lead to substantial gains in weak-to-strong generalization. Generalization-based approaches to weak-to-strong learning are complementary to scalable oversight methods, in which the weak supervisor interacts with the strong model to improve the quality of the weak supervision.

6.2.3 Concrete Problems: Scientific Understanding

We will need an extremely high degree of trust and reliability in our methods for aligning superhuman models in high-stakes settings. We will not get this from strong benchmark performance alone. Instead, we also need a thorough understanding of precisely *when* and *why* our methods work. Example questions of interest include:

- What explains the difference between the relatively strong results on NLP datasets and the relatively poor results with reward models when using naive finetuning?
- What makes a concept easy or hard to elicit? What is a good definition of "salience"?
- Can we reliably estimate generalization error at test time without any labels? For example, can we measure the degree of weak-to-strong underspecification (Lee et al., 2022b)?
- Can we reliably extrapolate generalization error across many orders of magnitude using scaling laws?
- How important are the errors in the weak supervision, precisely? How do different kinds of weak label biases affect generalization?
- How robust are our proposed methods to optimization pressure?

In Section 5 we only scratched the surface for understanding weak-to-strong generalization, but future work will need to go much further. An advantage of our setup is that it makes it easy to run simple experiments to scientifically study generalization phenomena across a wide range of settings.

6.3 Conclusion

Recent progress in AI has been faster than almost anyone anticipated (Steinhardt, 2022; Bengio et al., 2023). For an increasing number of researchers, the possibility of superhuman models being developed this decade has become increasingly plausible. Broadly superhuman models would be extraordinarily powerful and, if misused or misaligned with humans values, could potentially cause catastrophic harm (CAIS, 2022). Given the stakes, we need to establish extremely high reliability in the alignment of these systems ahead of time. But for years it has been unclear how to empirically study superhuman model alignment. We believe it is now easier to make progress on this problem than ever before.

7 Acknowledgements

We would like to thank Boaz Barak, Paul Christiano, Jacob Steinhardt, Ananya Kumar, Jakub Pachocki, John Schulman, Wojciech Zaremba, Alec Radford, Nat McAleese, and William Saunders for valuable technical insights and discussions. We are grateful to Mia Glaese, Boaz Barak, Kush Bhatia, Jean-Stanislas Denain, Erik Jones, Polina Kirichenko, Daniel Kokotajlo, Yoonho Lee, Jessy Lin, Richard Ngo, John Schulman, Peter Tong, Fred Zhang, Ruiqi Zhong, Ryan Greenblatt, Fabien Roger, Paul Christiano, Steven Adler, Rai Pokorny, Adam Kalai, Jacob Hilton, Roger Grosse, Dan Hendrycks, Alec Radford, and Scott Aaronson for helpful feedback on earlier drafts of this paper. We also thank Shantanu Jain, Avital Oliver, Suchir Balaji, Cathy Yeh, and the Platform team for infrastructure help. CB is also grateful to Dan Hendrycks, Jacob Steinhardt, and Paul Christiano for many formative discussions over the years.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pp. 312–321. PMLR, 2019. (Cited on page 33)

Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pp. 12–20. Citeseer, 1997. (Cited on page 5)

Anas Awadalla, Mitchell Wortsman, Gabriel Ilharco, Sewon Min, Ian Magnusson, Hannaneh Hajishirzi, and Ludwig Schmidt. Exploring the landscape of distributional robustness for question answering models. *arXiv preprint arXiv:2210.12517*, 2022. (Cited on page 4)

Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. Learning the structure of generative models without labeled data. In *International Conference on Machine Learning*, pp. 273–282. PMLR, 2017. (Cited on page 4)

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a. (Cited on page 1, 5)

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b. (Cited on page 5, 47)

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995. (Cited on page 5)

Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018. (Cited on page 4)

Qiang Ning Ben Zhou, Daniel Khashabi and Dan Roth. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *EMNLP*, 2019. (Cited on page 29)

Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, Gillian Hadfield, et al. Managing ai risks in an era of rapid progress. *arXiv preprint arXiv:2310.17688*, 2023. (Cited on page 18)

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. (Cited on page 4)

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10925–10934, 2022. (Cited on page 4)

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *OpenAI Blog*, 2023. (Cited on page 47)

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. (Cited on page 29)

Sam Bowman. Artificial sandwiching: When can we test scalable alignment protocols without humans? *AI Alignment Forum*, 2022. (Cited on page 5)

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosuite, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*, 2022. (Cited on page 2, 47)

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. (Cited on page 14)

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. (Cited on page 5)

CAIS. Statement on ai risk, 2022. (Cited on page 4, 19, 47)

Joe Carlsmith. Scheming ais: Will ais fake alignment during training in order to get power? *arXiv preprint arXiv:2311.08379*, 2023. (Cited on page 48)

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021. (Cited on page 17, 35, 40)

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021. (Cited on page 36)

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020a. (Cited on page 35)

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. *Advances in Neural Information Processing Systems*, 33:21061–21071, 2020b. (Cited on page 33)

Paul Christiano. Approval-directed bootstrapping. *AI Alignment Forum*, 2018. (Cited on page 8)

Paul Christiano. Capability amplification. *AI Alignment Forum*, 2019. (Cited on page 8)

Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. *arXiv preprint arXiv:1810.08575*, 2018. (Cited on page 2, 5)

Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge. Technical report, Alignment Research Center (ARC), 2022. (Cited on page 5, 11, 17, 44)

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. (Cited on page 1, 2, 5, 47)

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019. (Cited on page 29)

Ajeya Cotra. The case for aligning narrowly superhuman models. *AI Alignment Forum*, 2021. (Cited on page 5)

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28, 2015. (Cited on page 14)

Abram Demski and Scott Garrabrant. Embedded agency. *arXiv preprint arXiv:1902.09469*, 2019. (Cited on page 2)

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. (Cited on page 40)

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html. (Cited on page 42)

Owain Evans, Owen Cotton-Barratt, Lukas Finnveden, Adam Bales, Avital Balwit, Peter Wills, Luca Righetti, and William Saunders. Truthful ai: Developing and governing ai that does not lie. *arXiv preprint arXiv:2110.06674*, 2021. (Cited on page 5)

Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017. (Cited on page 4)

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013. (Cited on page 4)

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2018. (Cited on page 4, 13)

Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022. (Cited on page 1, 5)

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021. (Cited on page 4)

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. (Cited on page 4, 10, 33, 34)

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. (Cited on page 4)

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018. (Cited on page 4)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. (Cited on page 40)

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31, 2018. (Cited on page 4)

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pp. 2712–2721. PMLR, 2019. (Cited on page 4)

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *arXiv preprint arXiv:2008.02275*, 2020a. (Cited on page 29)

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*, 2020b. (Cited on page 4)

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *Sort*, 2 (4):0–6, 2021. (Cited on page 40)

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. (Cited on page 4)

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. (Cited on page 35)

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*, 2019. (Cited on page 29)

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*, 2019. (Cited on page 2, 48)

Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint arXiv:1805.00899*, 2018. (Cited on page 2, 5)

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. (Cited on page 36)

Fereshte Khani, Aditi Raghunathan, and Percy Liang. Maximum weighted loss discrepancy. *arXiv preprint arXiv:1906.03518*, 2019. (Cited on page 5)

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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. 252–262, 2018. (Cited on page 29)

Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254, 2019. (Cited on page 5)

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (Cited on page 40, 41)

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014. (Cited on page 4)

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*, 2023. (Cited on page 4, 16)

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. (Cited on page 17, 40)

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991. (Cited on page 35)

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. (Cited on page 4, 35)

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. (Cited on page 4)

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013. (Cited on page 33)

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023. (Cited on page 5)

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *The Eleventh International Conference on Learning Representations*, 2022a. (Cited on page 4)

Yoonho Lee, Huaxiu Yao, and Chelsea Finn. Diversify and disambiguate: Learning from underspecified data. *arXiv preprint arXiv:2202.03418*, 2022b. (Cited on page 18)

Jan Leike and Ilya Sutskever. Introducing superalignment. *OpenAI Blog*, 2023. (Cited on page 8, 47)

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018. (Cited on page 2, 5)

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. (Cited on page 4)

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv preprint arXiv:2306.03341*, 2023. (Cited on page 5, 47)

Lichess Team. Lichess Database. `https://github.com/lichess-org/database`, 2023. Accessed: 2023. (Cited on page 7)

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. (Cited on page 5)

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021. (Cited on page 5)

Ziquan Liu, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Rong Jin, Xiangyang Ji, and Antoni B Chan. An empirical study on distribution shift robustness from the perspective of pre-training and data augmentation. *arXiv preprint arXiv:2205.12753*, 2022. (Cited on page 4)

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, pp. 6543–6553. PMLR, 2020. (Cited on page 4)

Ian R McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, et al. Inverse scaling: When bigger isn't better. *arXiv preprint arXiv:2306.09479*, 2023. (Cited on page 8, 13)

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022. (Cited on page 5)

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018. (Cited on page 29)

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*, 2022. (Cited on page 48)

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019. (Cited on page 29)

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks. (Cited on page 47)

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. (Cited on page 2, 7, 28)

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. (Cited on page 1, 2, 5, 7, 32, 47)

Lorenzo Pacchiardi, Alex J Chan, Sören Mindermann, Ilan Moscovitz, Alexa Y Pan, Yarin Gal, Owain Evans, and Jan Brauner. How to catch an ai liar: Lie detection in black-box llms by asking unrelated questions. *arXiv preprint arXiv:2309.15840*, 2023. (Cited on page 5)

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85): 2825–2830, 2011. (Cited on page 42)

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022a. (Cited on page 47)

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*, 2022b. (Cited on page 47)

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*, 2018. (Cited on page 29)

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017. (Cited on page 14)

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021. (Cited on page 4)

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, pp. 269. NIH Public Access, 2017. (Cited on page 4)

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. (Cited on page 4, 33)

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016. (Cited on page 47)

Fabien Roger, Ryan Greenblatt, Max Nadeau, Buck Shlegeris, and Nate Thomas. Measurement tampering detection benchmark. *arXiv preprint arXiv:2308.15605*, 2023. (Cited on page 5)

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8722–8731, 2020. (Cited on page 29)

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. (Cited on page 40)

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. (Cited on page 4)

Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373, 2021. (Cited on page 5)

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019. (Cited on page 29)

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022. (Cited on page 2, 5, 47)

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Arpit Bansal, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. Datasets for studying generalization from easy to hard examples. *arXiv preprint arXiv:2108.06011*, 2021a. (Cited on page 29)

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. *Advances in Neural Information Processing Systems*, 34:6695–6706, 2021b. (Cited on page 7, 29)

Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. (Cited on page 4, 33)

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013. (Cited on page 29)

Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020. (Cited on page 5)

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. (Cited on page 4)

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. (Cited on page 35)

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919, 2021. (Cited on page 4, 13)

Jacob Steinhardt. Ai forecasting: One year in, 2022. (Cited on page 18)

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020. (Cited on page 1, 5, 15, 32)

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019. (Cited on page 29)

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. Quartz: An open-domain dataset of qualitative relationship questions. *arXiv preprint arXiv:1909.03553*, 2019. (Cited on page 29)

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. (Cited on page 4)

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. (Cited on page 29)

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019. (Cited on page 29)

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. (Cited on page 29)

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2020. (Cited on page 33)

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021. (Cited on page 28, 29)

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. (Cited on page 46)

Johannes Welbl, Nelson F Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*, 2017. (Cited on page 29)

John Wentworth. Alignment by default. *AI Alignment Forum*, 2020. (Cited on page 5)

Gordon Seidoh Worley. Bootstrapped alignment. *AI Alignment Forum*, 2021. (Cited on page 8)

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022a. (Cited on page 4, 36)

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022b. (Cited on page 4, 36)

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021. (Cited on page 2)

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020. (Cited on page 4, 33)

Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. (Cited on page 4)

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019. (Cited on page 29)

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018. (Cited on page 5)

Yuan Zhang, Jason Baldridge, and Luheng He. Paws: Paraphrase adversaries from word scrambling. In *Proc. of NAACL*, 2019. (Cited on page 29)

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. (Cited on page 4, 36)

Appendix Outline

- In Appendix A, we provide additional details on our setup and experiments.
- In Appendix B, we describe additional results, including negative results and methods that did not work well in our experiments.
- In Appendix C, we report results on easy-to-hard generalization, where we only provide supervision on easy examples.
- In Appendix D, we provide results in two more weak-to-strong learning settings: a self-supervised computer vision setting on ImageNet, and a pure linear probing setting.
- In Appendix E, we provide additional results and discussion on the effect of weak supervisor error simulation.
- In Appendix F, we discuss how we believe methodological progress should be made on superalignment.
- In Appendix G, we describe how our work fits into the bigger picture of alignment.

A Further experimental details

Here, we provide further details on our experiments. Across all tasks, we use pretrained base models from the GPT-4 family (OpenAI, 2023), spanning a range of model sizes.

A.1 NLP Tasks

Data preprocessing. We use popular NLP classification benchmark datasets listed in Table 1. We obfuscate the names of the datasets in our plots (e.g. Figure 12) for confidentiality; across all figures, we replace the names of the datasets with their order in a randomized sequence. We apply various preprocessing to the datasets. For example, some tasks are in FLAN (Wei et al., 2021) and we use their preprocessing. For ANLI we group neutral entailments with contradictions. We convert each dataset to a binary classification problem. For multiple-choice datasets, suppose each datapoint has a question Q and multiple candidate answers $A_1, \dots, A_k$. We then convert this datapoint to k new datapoints of the form (Q, A_i), where the label is 0 for all incorrect answers A_i and 1 for the correct answers. In this procedure, we also aim to maintain class balance, so we keep the same number of correct and wrong answers per question[6]. We are also additionally rebalancing the classes in datasets where one of the classes represents more than 55% of the data. To do so, we randomly drop datapoints from the dominant class, so that the classes are perfectly balanced.

Models. In order to adapt our language models to the classification setting, we replace the unembedding layer of the model with a linear classification head with two outputs. We initialize the weights of the classification head with the unembedding weights for tokens "0" and "1".

Training hyperparameters. We finetune all models for 2 epochs using a batch size of 32. In the weak-to-strong generalization experiments, we early stop training based on the accuracy with respect to the weak labels on a held-out validation set. See Section 5.1.1 for relevant discussion. We only tuned the hyper-parameters of our methods on smaller model sizes, and on a subset of 8 datasets. The full GPT-4 model and most of the datasets were held-out, except for datasets [5–12] (see Figure 12).

Weak labels. To produce the weak labels, we split the original dataset in half. We ensure that related datapoints, e.g. datapoints that share the same question or premise, are always grouped together into the same half. Then, we train the weak supervisor model on the first half of the dataset, and use its prediction on the other half as the weak labels. We additionally save the weak labels on the test set to evaluate metrics such as agreement in Section 5.1.3. The weak labels are soft labels on the training data, i.e. the class probabilities predicted by the supervisor.

Evaluation. For all datasets, we report accuracy on the test set which is also balanced to have an equal number of datapoints in each class. In particular, random guess performance corresponds to 50% accuracy on all NLP datasets.

[6] In some datasets there are multiple correct answers for each question.

Table 1: **Datasets and their sources.** We summarize the NLP datasets we use and their original sources.

Dataset	Original Source
BoolQ	Clark et al. (2019)
CosmosQA	Huang et al. (2019)
DREAM	Sun et al. (2019)
ETHICS [Justice]	Hendrycks et al. (2020a)
ETHICS [Deontology]	Hendrycks et al. (2020a)
ETHICS [Virtue]	Hendrycks et al. (2020a)
ETHICS [Utilitarianism]	Hendrycks et al. (2020a)
FLAN ANLI R2	Nie et al. (2019); Wei et al. (2021)
GLUE CoLA	Warstadt et al. (2019); Wang et al. (2018)
GLUE SST-2	Socher et al. (2013); Wang et al. (2018)
HellaSwag	Zellers et al. (2019)
MCTACO	Ben Zhou & Roth (2019)
OpenBookQA	Mihaylov et al. (2018)
PAWS	Zhang et al. (2019)
QuAIL	Rogers et al. (2020)
PIQA	Bisk et al. (2020)
QuaRTz	Tafjord et al. (2019)
SciQ	Welbl et al. (2017)
Social IQa	Sap et al. (2019)
SuperGLUE MultiRC	Khashabi et al. (2018); Wang et al. (2019)
SuperGLUE WIC	Pilehvar & Camacho-Collados (2018); Wang et al. (2019)
Twitter Sentiment	Zhang et al. (2019)

Detailed results. In Figure 12, we provide detailed results across all datasets for both the baseline and the auxiliary confidence loss introduced in Section 4.3. In Figure 13 we report the detailed results on overfitting to the weak supervisor predictions for the NLP datasets.

A.2 Chess Puzzles

Data preprocessing. The GPT-4 pretraining dataset included chess games in the format of move sequence known as Portable Game Notation (PGN). We note that only games with players of Elo 1800 or higher were included in pretraining. These games still include the moves that were played in-game, rather than the best moves in the corresponding positions. On the other hand, the chess puzzles require the model to predict the best move. We use the dataset originally introduced in Schwarzschild et al. (2021b) which is sourced from `https://database.lichess.org/#puzzles` (see also Schwarzschild et al., 2021a). We only evaluate the models ability to predict the first move of the puzzle (some of the puzzles require making multiple moves). We follow the pretraining format, and convert each puzzle to a list of moves leading up to the puzzle position, as illustrated in Figure 14. We use $50k$ puzzles sampled randomly from the dataset as the training set for the weak models and another $50k$ for weak-to-strong finetuning, and evaluate on $5k$ puzzles. For bootstrapping (Section 4.3.1), we use a new set of $50k$ puzzles from the same distribution for each step of the process.

Training hyperparameters. We train (finetune) all models for 5 epochs using a batch size of 32. We do not apply early-stopping.

Weak labels. We produce weak labels by sampling predictions at temperature $T = 0$ (greedy decoding) from the weak model on a held-out set of additional $50k$ puzzles. The weak labels are completions showing the highest likelihood move according to the weak model.

Evaluation. To evaluate the models, we sample completions at temperature $T = 0$ on the held out test set, and compute the fraction of datapoints where the model outputs the correct next move.

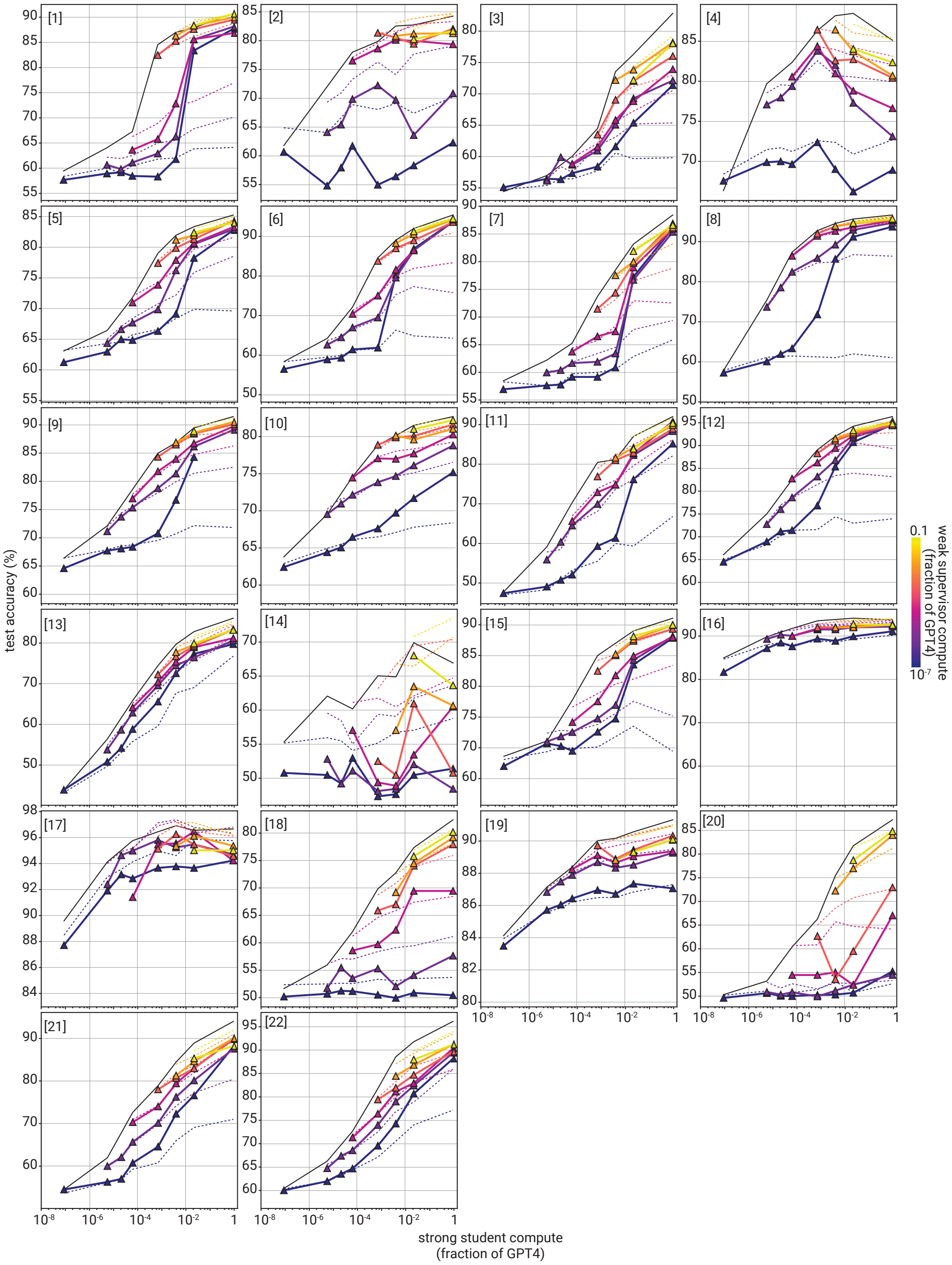


Figure 12: **Full weak-to-strong generalization results across 22 NLP datasets.** Test accuracy as a function of strong student compute across our full suite of standard NLP tasks. See Table 1 for dataset details.

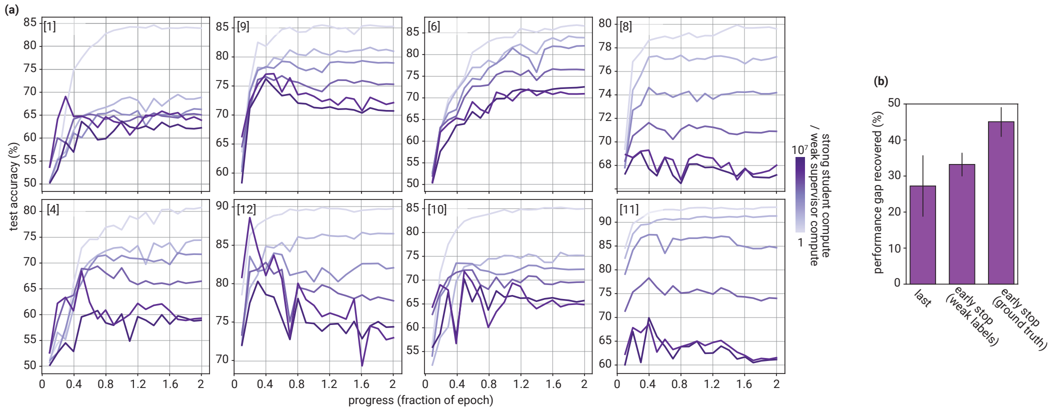
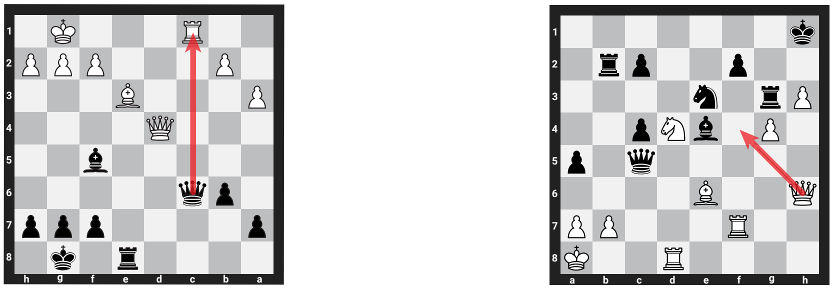


Figure 13: **Overfitting during training, for NLP datasets. Strong models overfit to the weak labels.** **(a)** Ground truth test accuracy of strong students over the course of training for a subset of our NLP task. Hues indicate the gap between weak supervisor and strong student model compute. Inset numbers indicate dataset id (compare Figure 12). (**b**) Median best, early-stopped according to weak label agreement, and final performance gap recovered (PGR) aggregated across all supervisor-student pairs and all NLP tasks. Error bars indicate standard error of the mean (s.e.m.).



Prompt: "1. d4 1... Nf6 2. Nf3 2... d5 3. e3 3... e6 4. Bd3 4... c5 5. c3 5... Be7 6. Nbd2 6... O-O 7. O-O 7... Nc6 8. Re1 8... Bd7 9. e4 9... dxe4 10. Nxe4 10... cxd4 11. Nxf6+ 11... Bxf6 12. cxd4 12... Nb4 13. Be4 13... Qb6 14. a3 14... Nc6 15. d5 15... exd5 16. Bxd5 16... Bf5 17. Bxc6 17... Qxc6 18. Nd4 18... Bxd4 19. Qxd4 19... Rfe8 20. Rxe8+ 20... Rxe8 21. Be3 21... b6 22. Rc1 22..."

Label: " Qxc1+"

(a) Elo-695 puzzle

Prompt: "1. e4 1... e5 2. Nc3 2... Nf6 3. Nf3 3... Nc6 4. Bb5 4... Bc5 5. Bxc6 5... dxc6 6. d3 6... Bg4 7. h3 7... Bxf3 8. Qxf3 8... O-O 9. g4 9... Bb4 10. Bd2 10... Nd7 11. h4 11... Be7 12. g5 12... Nc5 13. O-O-O 13... Qd7 14. h5 14... Qd8 15. Qg3 15... Ne6 16. Rdg1 16... b5 17. Qxe5 17... a5 18. f4 18... Re8 19. Qf5 19... b4 20. Na4 20... Nd4 21. Qg4 21... c5 22. f5 22... Ra6 23. f6 23... Bd6 24. fxg7 24... Kxg7 25. Rg2 25... Qc8 26. h6+ 26... Kg8 27. Qh5 27... Qd7 28. Rf1 28... Re6 29. Rgf2 29... Rg6 30. c3 30... bxc3 31. Nxc3 31... a4 32. Nd5 32... Qb5 33. Nf6+ 33... Kh8 34. Qh3 34... Rb6 35. Be3 35... Ne6 36. Nxh7 36... Qxd3 37. Rd1 37... Qc4+ 38. Kb1 38... Qxe4+ 39. Ka1 39... Be5 40. Nf6 40... Qc4 41. Nd5 41... Rb7 42."

Label: " Qf5"

(b) Elo-2253 puzzle

Figure 14: **Chess puzzles: example datapoints.** Two representative examples of an easy **(a)** and a hard **(b)** chess puzzle with corresponding prompts and target label formats.

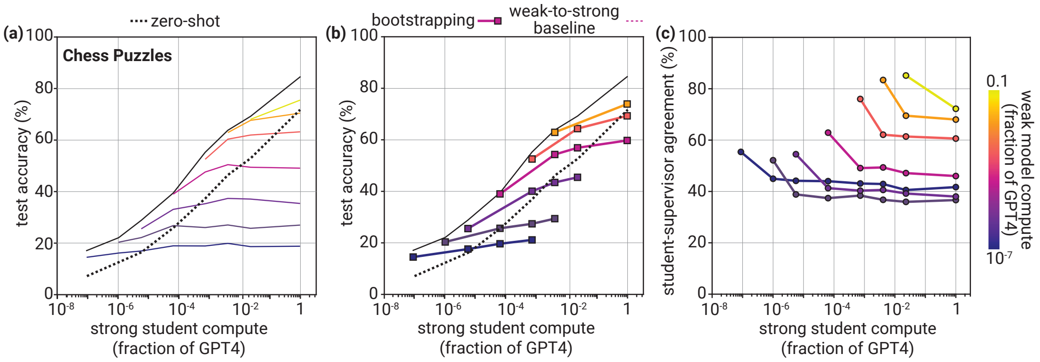


Figure 15: **Additional results on chess.** Test accuracy of **(a)** baseline and **(b)** bootstrapping (see section 4.3.1) compared to a zero-shot baseline. Zero-shot performance improves with model size, and students supervised with much weaker supervisors sometimes underperform compared to the corresponding zero-shot model. **(c)** Supervisor-student agreement on the chess puzzle data. Similar to Figure 8, agreement decreases as the student becomes larger. Hue of line indicates compute of weak supervisor.

Zero-shot results. In Figure 15(a, b), we compare the naive baseline and bootstrapping (see section 4.3.1) generalization to a zero-shot baseline on the chess puzzle data. Especially since the models were pretrained on chess games, zero-shot evaluation provides a strong baseline. In particular, strong students trained with much weaker supervisors underperform the zero-shot baseline for the same model size in some cases.

Supervisor-student agreement results. In Figure 15(c), we report the supervisor-student agreement on the chess puzzles. Similar to the NLP tasks (see Section 5.1.3), the agreement on chess also decreases as the student models get larger.

A.3 ChatGPT Reward Modeling

Data preprocessing. Each datapoint presents a dialog d between a user and an assistant, with a last message coming from the user; for each dialog, there are multiple candidate completions $(c_1, c_2, \ldots, c_m)$, i.e. responses from the assistant. We also have access to pairwise comparisons of completions, where the labeler specifies the preferred completion within a given pair of completions. To sum up, the datapoints can be viewed as (d, c_1, c_2, y), where the label y is 1 if the labeler preferred completion c_2 and 0 otherwise. We use a mixture of multiple datasets used to train the reward models for ChatGPT.

Models. To adapt the language models to the reward modeling setting, we replace the unembedding layer of the model with a linear head with a single output, which is the logit for a given completion. The weights for this head are initialized to the unembedding weights of an arbitrary token in the original embedding layer. Similar to past work (Stiennon et al., 2020; Ouyang et al., 2022), we run two forward passes for each comparison, and the model prediction is given by $\sigma(\mathcal{M}_w(d, c_2) - \mathcal{M}_w(d, c_1))$, where σ is the sigmoid function and $\mathcal{M}_w(d, c)$ is the logit for completion c predicted by the model.

Training hyperparameters. We train for 1 epoch with a batch size of 220. We do not apply early-stopping.

Weak labels. We train the weak models on half of the available comparison data, and then make predictions on the other half. The weak label y_w for a comparison (d, c_1, c_2) is given by $y_w = \sigma(\mathcal{M}_w(d, c_2) - \mathcal{M}_w(d, c_1))$, where σ is the sigmoid function and $\mathcal{M}_w(d, c)$ is the logit for completion c predicted by the weak model.

Supervisor-student agreement results. In Figure 16, we report the supervisor-student agreement on the RM task. Similar to the NLP tasks in Figure 8 and chess puzzles in Figure 15(c), the agreement decreases as the student gets larger.

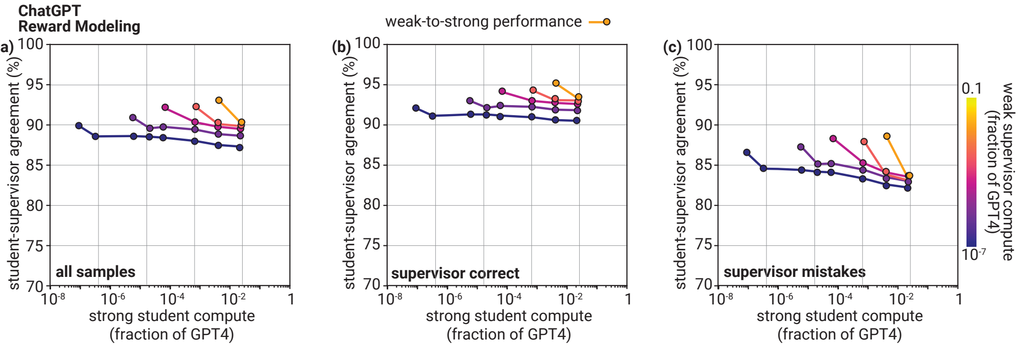


Figure 16: **Supervisor-student agreement decreases for stronger students on RMs.** Please refer to caption of Figure 8 for detailed explanation of the plot. We reproduce the supervisor-student agreement experiment on the reward modeling data, and observe similar trends to the NLP tasks.

Generative finetuning. In Figure 17, we show that the PGR improvements from the generative finetuning on RM data (Section 5.2.2) and from early-stopping on ground truth test accuracy (Section 5.1.1) stack together, leading to results competitive with the NLP and chess settings. In Figure 18, we report the results of an experiment similar to Figure 10, but where the weak models are also pretrained with an additional generative finetuning step on the RM data.

A.4 Auxiliary Confidence Loss

Here, we provide a detailed description of the method we use in Section 4.3.2.

We use the following loss function:

$$L_{\text{conf}}(f) = (1-\alpha)\cdot \text{CE}(f(x), f_w(x)) + \alpha \cdot \text{CE}(f(x), \hat{f}_t(x)) \tag{1}$$

where $\text{CE}(\cdot,\cdot)$ is the cross-entropy loss between the predictive distributions on a given input x, $f_w(x) \in [0,1]$ represents the weak label predictive distribution, $f(x) \in [0,1]$ is the strong model predictive distribution, α is a weight and t is a threshold. The predictions $\hat{f}_t(x)$ correspond to hardened strong model predictions using a threshold t, i.e. $\hat{f}_t(x) = I[f(x) > t] \in \{0,1\}$ where I is the indicator function. We set the threshold t adaptively, so that $f(x) > t$ holds for exactly half of examples in the batch[7]. We set $\alpha_{\text{max}} = 0.75$ for the largest student models and to 0.5 otherwise and linearly warm-up α from 0 to α_{max} over the first 20% of training.

Our balancing mechanism incorporates a prior over the distribution of labels into training and is only practically feasible in the low-n classification setting. For most weak-strong pairs and datasets, it had a small or neutral effect on weak-to-strong generalization; however, in a few settings it made a significant improvement.

We note that the loss in Equation 1 can be rewritten as a self-bootstrapping loss:

$$L_{\text{conf}}(f) = \text{CE}(f(x), (1-\alpha)\cdot f_w(x) + \alpha \cdot \hat{f}_t(x)), \tag{2}$$

i.e. the cross-entropy target is a mixture of the weak model predictions and the (thresholded) predictions of the strong student itself. This loss is related to the bootstrapping methods in Reed et al. (2014) and Arazo et al. (2019) for addressing label noise. It is also similar to self-training (Lee et al., 2013) and conditional entropy minimization (Grandvalet & Bengio, 2004), which have led to state-of-the-art results in semi-supervised learning (Xie et al., 2020) and domain adaptation (Shu et al., 2018). Chen et al. (2020b) and Wei et al. (2020) show that self-training can mitigate the bias of the supervisor model.

In Appendix B we also describe other methods we considered; for most of these methods, we got negative early results.

[7] The choice of exactly half reflects the prior over classes, and should be computed explicitly from weak model predictions in non-balanced or non-binary settings.

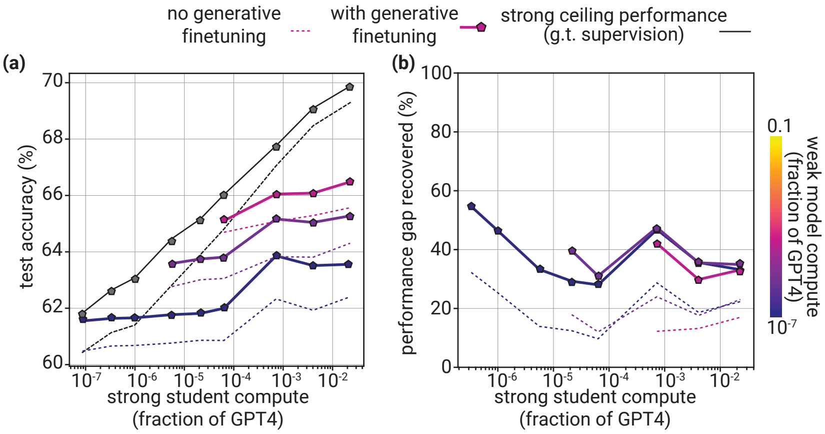


Figure 17: **The benefits of improved task-specific tuning and ground truth early stopping stack, resulting in even higher PGR.** Like Figure 10 but with ground truth early stopping based on test accuracy.

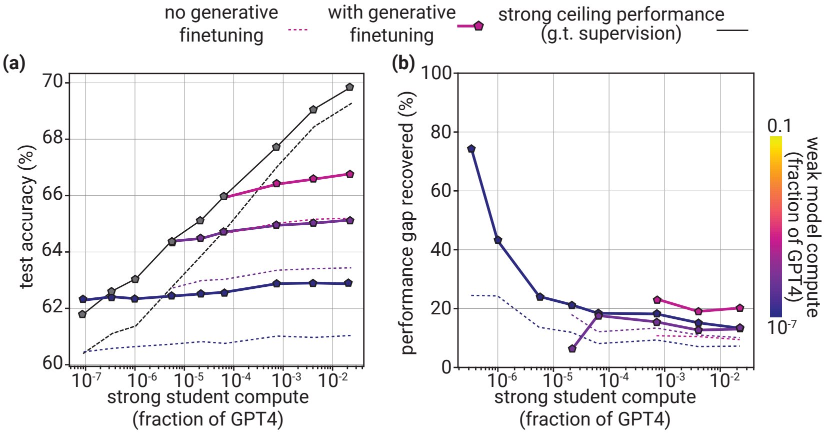


Figure 18: **PGR improves when *both* supervisors and students have an extra generative finetuning step.** Like Figure 10 but where “with generative finetuning” indicates that both supervisors and students have an extra generative finetuning step. In other words, for this experiment all base models have an extra generative finetuning step following pretraining.

B Additional results on methods

We did preliminary experiments on a variety of methods for improving the strong model performance in our weak-to-strong generalization setting. We found many of them not useful for improving over the naive finetuning baseline, and others yielding limited improvements on a subset of settings but not consistently over all datasets and model sizes. We summarize the algorithms, the motivations, and the takeaways below. Note that we did not optimally tune each of the methods, so it is possible that with better tuning they may still perform well.

Confidence thresholding. To filter out incorrect weak labels, we used a simple cut-off method that selected only the top 5% to 20% examples from each class where the weak supervisor is most confident to train the strong model. We found that our weak labels are typically well-calibrated, but confidence thresholding only helps when the weak labels are very bad (e.g. 60% accuracy) and stops being useful when the weak labels reach around 70% to 80% accuracy. We observed these results both in NLP and in the chess puzzle settings. See Appendix C for more discussion of related experiments.

Confidence losses. To encourage strong model to make confident predictions (Grandvalet & Bengio, 2004), we added an auxiliary loss that encourages the model predicted class probabil-

Table 2: **Custom prompts used in the zero-shot and few-shot experiments**. We design a simple custom prompt for each of the tasks in the table below. In the few-shot setting, we also append labeled (with ground truth or weak labels) examples to the prompt.

Dataset	Custom Prompt
CosmosQA	*There is some context, followed by a question about the context, and an answer. Respond with 1 if the answer is correct, and 0 otherwise.*
Deontology	*There is a short scenario, consisting of a situation or request, followed by an excuse. Respond with 1 if the excuse is valid, and with 0 otherwise.*
Justice	*There is a statement. Respond with 1 if it is correct or just and with 0 otherwise.*
QuAIL	*There is a news context, followed by a question about the context, and an answer to the question. Respond with 1 if the answer is correct, and with 0 otherwise.*
SciQ	*There is a science knowledge question, followed by an answer. Respond with 1 if the answer is correct, and with 0 otherwise.*
Social IQa	*There is some context, followed by a social question, followed by an answer. Respond with 1 if the answer is correct, and 0 otherwise.*
Virtue	*There is a short scenario, followed by a judgement of the person involved. Respond with 1 if the judgement is correct, otherwise respond with 0.*

ity p to be far away from 0.5. We tried both the l_2 loss $-(p-0.5)^2$ and the entropy loss $p\log p + (1-p)\log(1-p)$. We found these losses to be helpful in preliminary experiments in the linear probing setting, but they generally performed less well than the confidence auxiliary loss in Equation 1 in the finetuning setting. We have also observed negative results with the confidence losses when the training data is highly class-imbalanced or when we do not use the rebalancing procedure described in Section 4.3.

Product confidence loss. We also tried a confidence-like loss which sets the cross entropy targets to be proportional to the product of the probabilities that the weak and strong models assign, renormalized across classes and without propagating gradients through the targets. In preliminary experiments, this loss consistently gave positive results over the baseline on two NLP tasks, but performed poorly compared to our main confidence loss. Variants like geometric mean instead of product gave no boost. Compared to the confidence loss, it could be useful as it has no inter-batch dependence and could potentially be adapted for generative tasks.

LP-FT. We used the LP-FT technique proposed in Kumar et al. (2022) which first trains a linear probe on frozen strong model representations and then finetunes all layers, to avoid destroying the pretrained representation. We were unable to get improvements compared to the finetuning baseline.

Weight regularization. To regularize the strong model weights to avoid imitating the weak labels[8], we tried a variety of regularization techniques for strong model training, including stronger weight decay (Krogh & Hertz, 1991) and dropout (Srivastava et al., 2014). We did not find significant improvement.

LoRA. As another regularization technique, we also considered low-rank adaptation (LoRA) (Hu et al., 2022), i.e. only making a low-rank update to the parameters of each layer of the model during finetuning. We did not find any improvement, even when sweeping the LoRA rank.

Data augmentation. Inspired by the success of consistency algorithms in self-supervised training (Chen et al., 2020a; Caron et al., 2021), we used the strong student models to rephrase the inputs in each sample, and added an auxiliary loss enforcing the strong model predictions to be consistent between original and rephrased samples. We did not find any improvement on a selected subset of NLP datasets.

[8] However, as we discuss in Section 5.1.3, in our setup the strong model tends to be bad at imitating the weak labels. Therefore, regularization could be more important in settings where the strong model can fit the weak labels well.

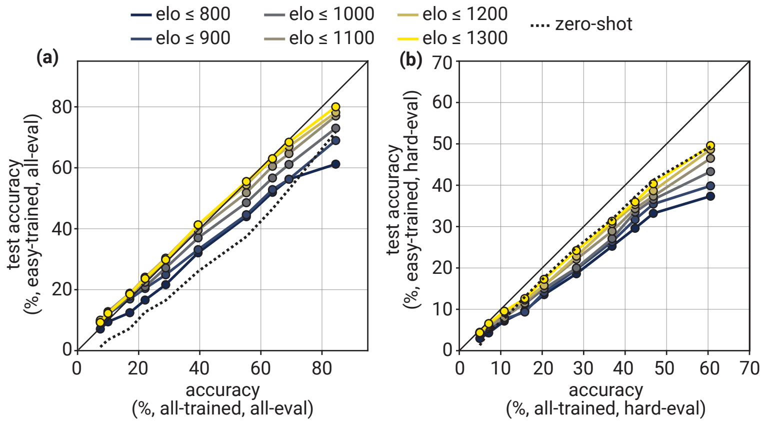


Figure 19: **Easy-to-hard generalization on chess puzzles.** We finetune models on chess puzzles with Elo $\leq t$, varying the threshold t, and evaluate the finetuned models on **(a)**: all test puzzles, and **(b)**: hard test puzzles with Elo ≥ 2000. Across the board, we see strong performance, even when training only on very easy puzzles (Elo ≤ 800). For reference, we also include the zero-shot performance of the model. Finetuning on easy puzzles, we improve upon the performance on average on the test set, but we do not improve on hard puzzles, compared to the zero-shot model.

Adding label noise, special losses for noisy labels. We experimented with the generalized cross-entropy loss proposed in Zhang & Sabuncu (2018) that is more robust to label noise, but did not find improvement over cross-entropy. We also tried adding random noise to weak labels, and found that the strong models were able to simulate the weak labels less well, especially early in training, but it did not ultimately result in improved performance.

Few-shot prompting. As an alternative to fine-tuning, we can use the in-context learning ability of the strong student models. For each task, we append a custom prompt shown in Table 2. For a detailed description of the results, see Section 5.2.1.

Weight averaging. Prior work (Izmailov et al., 2018; Cha et al., 2021; Wortsman et al., 2022b;a) suggested that various forms of weight averaging can substantially improve performance, especially in distribution shift settings. In our setup, we experimented with applying exponential moving averaging to the parameters of the model during training, but did not observe improvements relative to the baseline.

C Easy-to-hard generalization

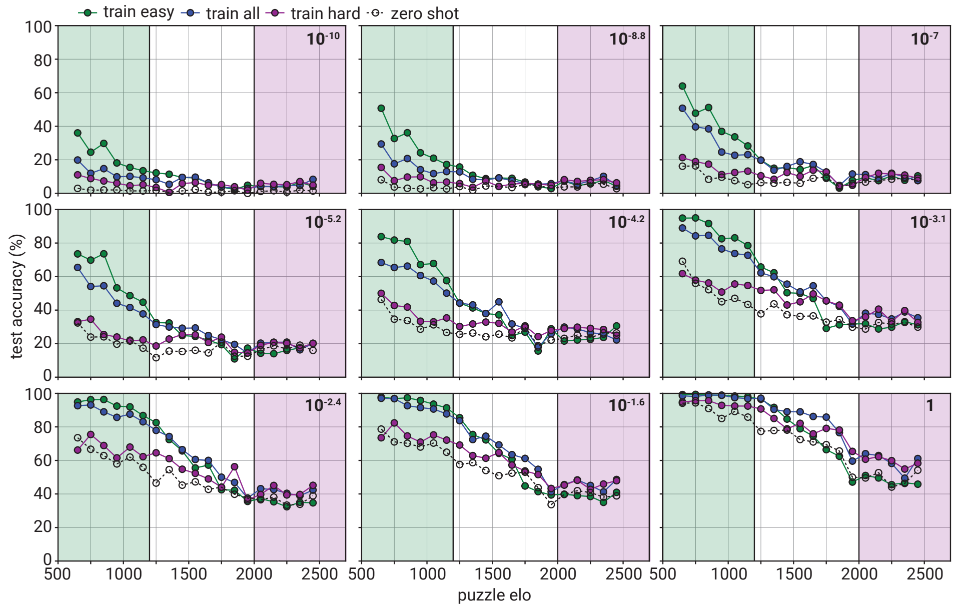
In Section 5.1.3 and Appendix E, we discuss that one reason weak-to-strong generalization may be difficult is if the weak labels have systematic errors that the strong model can learn to emulate. One natural type of systematic weak label error is to do poorly on hard examples and well on easy examples.

In this section, we focus on studying what we call **easy-to-hard generalization**, where we train only on easy examples using ground truth supervision, and assess generalization to harder examples.

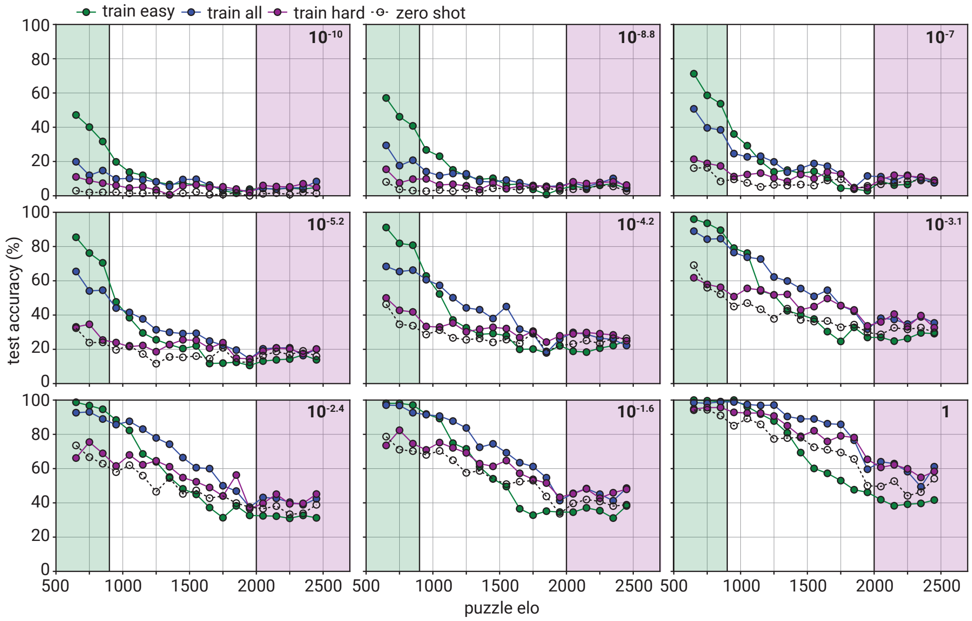
C.1 Chess puzzles

Each chess puzzle comes with a natural difficulty label: an Elo score, which describes its difficulty according to humans. On the `https://lichess.org` website, people try to solve puzzles, which can be viewed as a game between a puzzle and a human player. The Elo scores are then assigned to both human players and chess puzzles following the standard Elo algorithm.

We consider the easy-to-hard generalization problem, where the difficulty is defined according to the puzzle Elo rating. We note that the puzzle Elo describes the difficulty of the entire puzzle move sequence, while we are only training the model to predict the first move in the sequence



(a) Easy cutoff: Elo ≤ 1200



(b) Easy cutoff: Elo ≤ 900

Figure 20: **Easy-to-hard generalization on chess puzzles.** We present detailed performance of models finetuned on different subsets of chess puzzles across model sizes and test puzzle difficulty levels. For each model size, we compare models trained only on easy puzzles, hard puzzles, or all puzzles. We also include the zero-shot model performance. We provide results for the easy puzzle Elo cutoffs of **(a)**: 1200 and **(b)**: 900. All finetuned models are trained on $50k$ random datapoints from the corresponding distribution. The size of the model is shown in the upper-right corner of each panel, in terms of fraction of GPT-4 compute.

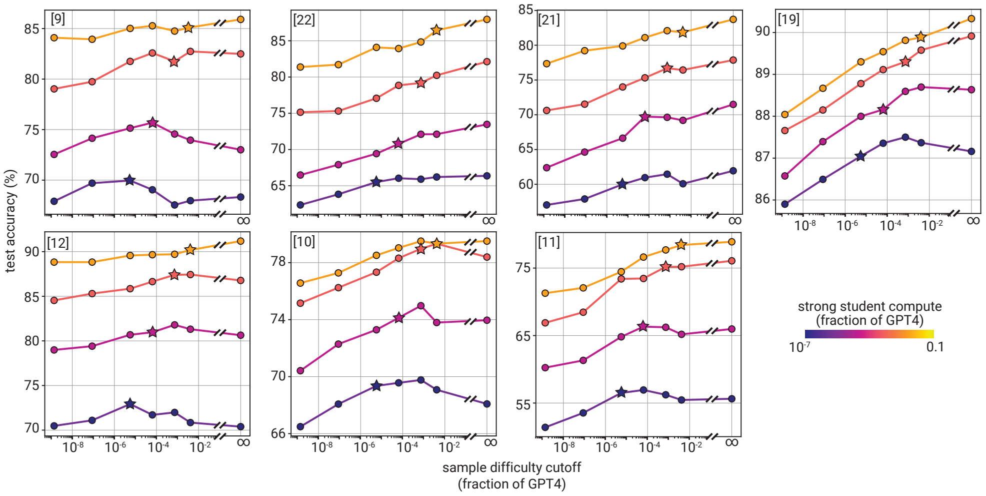


Figure 21: **Effect of varying training data difficulty on test set accuracy.** Test accuracy as a function of sample difficulty cutoff on a subset of our NLP tasks. The leftmost point on the horizontal axis corresponds to only using datapoints that models of all sizes that we consider get right when trained on other data sampled from the same task, and the rightmost point (denoted with ∞) corresponds to training on all datapoints; the point with value x on the horizontal axis corresponds to only using the datapoints that models with x or higher compute (fraction of GPT-4) consistently get right. Inset numbers indicate task id (compare Figure 12). Hue indicates compute of weak supervision. Stars indicate points where weak supervisor size corresponds to sample difficulty cutoff.

(see Appendix A.2). Consequently, the puzzle Elo is a high-quality but still imperfect measure of difficulty of the problem for humans. It is also important to note, that puzzle Elo may not be a good measure of difficulty for the models: easy puzzles for humans can be hard for the models and vice versa.

We then split the dataset into subsets according to the puzzle Elo. We consider the hard set to be puzzles with difficulty above Elo 2000. For the easy set, we consider cuttoffs in $\{800, 900, 1000, 1100, 1200, 1300\}$, and use puzzles with difficulty below the cutoff. We also consider the unrestricted set of *all* puzzles. We sample $50k$ puzzles from each of these sets randomly, and finetune the model on them[9].

We report the results in Figure 19, where we also provide the performance of a zero-shot baseline for reference. We plot the accuracy of the models trained on the easy subsets of puzzles against the performance of the same model trained on all puzzles. We find that the models generally perform well on average on the test set in panel (a), and outperform the zero-shot baseline. Interestingly, when evaluated on hard examples only, in panel (b), the models perform similarly to the zero-shot baseline, or slightly worse.

When trained on easy puzzles, the models shift towards performing well on the easy puzzles, and underperform on the hard puzzles. In Figure 20, we can see that generally the models improve upon the zero-shot baseline outside of their training difficulty range, often up to Elo of 1500 or higher, but underperform on the hardest examples.

C.2 NLP TASKS: DIFFICULTY THRESHOLDING

NLP tasks do not come with a natural source of difficulty labels, but we can create such labels by looking at performance as a function of model size.

[9] For easy puzzles with 800-Elo cutoff, we only use $25k$ puzzles, because there are not $50k$ puzzles available in this difficulty range.

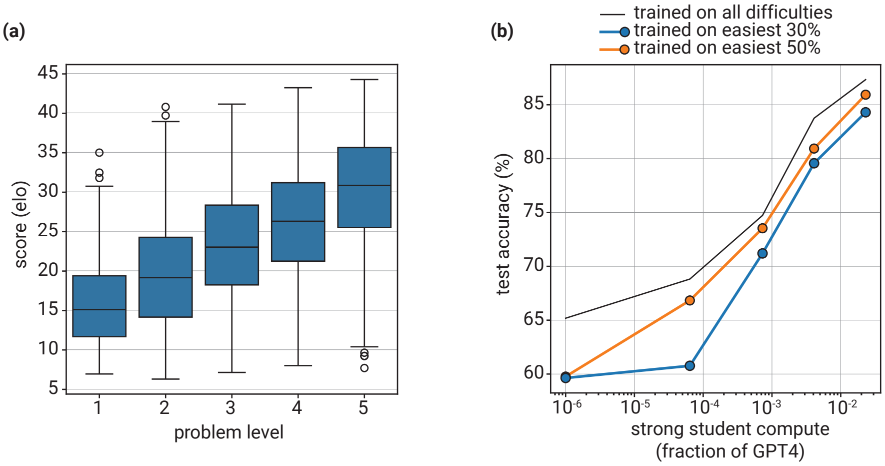


Figure 22: **Filtering training samples by GPT-4 generated Elo scores results in very good easy-to-hard generalization.** **(a)** GPT-4 generated Elo scores for different, human-defined, problem difficulties (1 - easiest, 5 - hardest) on the MATH dataset. **(b)** Average test accuracy as a function of strong student compute on a subset of our NLP tasks. Student is trained on ground truth labels on samples of all difficulties (black), only the 30% easiest tasks (orange), or only the 50% easiest tasks (blue).

We define *difficulty* of a datapoint based on the smallest model size that consistently predicts the label on this datapoint correctly, when trained on ground truth. For example, suppose we have 4 ground truth models W_1, W_2, W_3, W_4 that use compute $C_1 < C_2 < C_3 < C_4$ respectively. Suppose models W_1, W_3, W_4 predict the example correctly when it is in a held-out set, while W_2 predicts it incorrectly. Then we will assign a difficulty of C_3 to the example.

Then given a difficulty cutoff D, we filter the training set to examples with difficulty $\leq D$. We subsample the filtered set so that the number of training examples is equal to the number of examples at the lowest difficulty level. We train a model on the subsampled training set using ground truth labels, and measure its accuracy on a held out test set (with no subsampling).

The subsampling ensures that we use the same training set size for each difficulty cutoff. Using ground truth labels ensures that the label accuracy is the same (100%) for each cutoff. We also use the same test set for each cutoff. This setup lets us vary only training data difficulty, and measure its impact on the trained model's accuracy.

We plot results in Figure 21. The y-axis is accuracy on the test set, while the x-axis is the difficulty cutoff. Increasing the difficulty cutoff generally leads to an increase in accuracy. This result suggests that solving easy-to-hard generalization is non-trivial even if there are no weak label errors.

For smaller models (darker lines), the accuracy initially increases, but starts to decrease beyond a point. The drop generally happens when the difficulty cutoff exceeds the capacity of the model itself, i.e. when the examples are too difficult for the model to fit. However, large models trained on easy examples often perform well.

C.3 GPT-4 PREDICTED DIFFICULTY

Ultimately, we care about strong models generalizing from human supervision. From this perspective, it is important to understand whether we can achieve easy-to-hard generalization, where the difficulty is measured according to humans, rather than capacity-constrained models. In Appendix C.1, we explored this question in chess, but we would want to extend this analysis to the NLP tasks.

Most natural datasets do not come with information about problem difficulty. As a rough estimate, we automatically generated difficulty labels using GPT-4. More concretely, we used GPT-4 to rank pairs of examples in each dataset, asking "which question is easier, Question A or Question B?" We then calculated the Elo scores for each example via a finite number of random comparisons.

Table 3: **Weak-to-strong generalization on ImageNet**. We train linear probes on the representations extracted by DINO models with weak supervision from an AlexNet model. The strong students substantially outperform their weak supervisor.

Model	Top-1 Accuracy (%)	PGR (%)
AlexNet (weak supervisor)	56.6	-
Dino ResNet50	63.7	-
Dino ViT-B/8	74.9	-
AlexNet → DINO ResNet50	60.7	57.8
AlexNet → DINO ViT-B/8	64.2	41.5

To evaluate the quality of GPT-4 Elo score as a measure of difficulty, we performed correlation analysis against human annotations for datasets with human difficulty levels such as MATH (Hendrycks et al., 2021) and chess, as well as against weak model confidence. We found that the three measures align better for reasoning tasks such as MATH, as we show in Figure 22(a), but not much for some natural language tasks. When looking at the samples, we found that GPT-4 Elo scores tend to be higher for longer questions, but those questions may actually be easy for smaller models since they provide more context.

Using GPT-4 Elo score as a proxy for human difficulty, we used different cutoffs on scores to separate easy and hard examples, trained the strong models on the easy examples only (with ground truth labels), and evaluated on the hard examples. Preliminary results are shown in Figure 22(b).

In general, we found that using GPT-4 Elo as measure of hardness makes generalization slopes steeper than our main setup of weak-to-strong generalization. One possible confounder for interpretation is that our Elo measurements could be noisy, causing generalization to be better.

Note that this setup is a classic covariate shift problem, whereas our main setup focuses more on concept shift and noisy labels. It is unclear which setup would be more relevant, and we think it is important to study easy-to-hard generalization more thoroughly in future work.

D Other weak-to-strong settings

D.1 Self-supervised vision models

We additionally demonstrate weak-to-strong generalization in a simple image classification experiment. We use a pretrained AlexNet model (Krizhevsky et al., 2012) as a weak supervisor, and use it to generate weak labels on the ImageNet (Russakovsky et al., 2015) validation set. As a strong student, we use linear probing on frozen representations extracted by DINO models (Caron et al., 2021) based on ResNet-50 (He et al., 2016) and ViT-B/8 (Dosovitskiy et al., 2020) architectures. The DINO models are pretrained in an unsupervised way and did not observe direct supervision for ImageNet classification or any other classification task during pretraining, so this experiment does not have the pretraining leakage disanalogy discussed in Section 6.1.

We use $40k$ datapoints from the validation set to train the linear probes, and evaluate performance on the remaining $10k$ datapoints. For training the linear probes, we use a batch size of 128, Adam optimizer (Kingma & Ba, 2014) and a learning rate of 10^{-3}. We run 20 epochs of training for ResNet-50 and 5 epochs for ViT-B/8.

We report the results in Table 3. Similarly to our main experiments in Section 4, the student can substantially outperform the supervisor, achieving PGR on the order of 50%. This experiment shows that our results are not limited to the natural language setting, and generalize to other domains. It also shows that strong students can generalize from weak supervision on tasks where they only had indirect pretraining, i.e. where the knowledge of the task is latent.

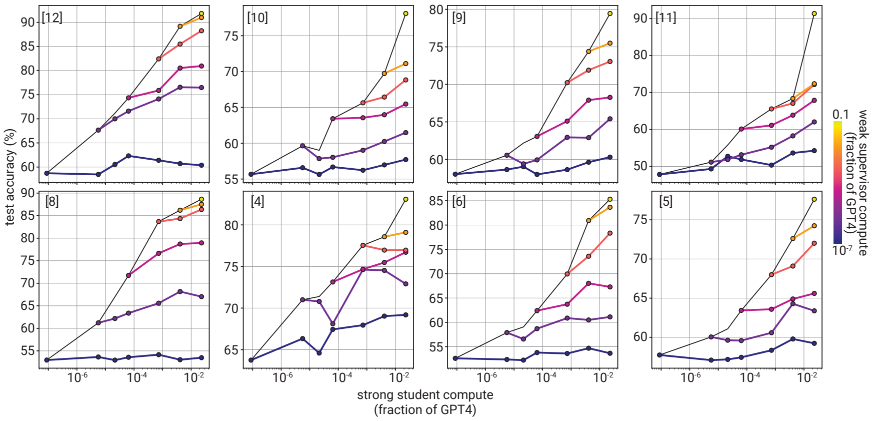


Figure 23: **Linear probing qualitatively matches finetuning weak-to-strong generalization.** Test accuracy as a function of strong student compute on a subset of our NLP tasks. Inset numbers indicate dataset id (compare Figure 12). Accuracy of a linear probe on student model trained with ground truth in black, accuracy of linear probe on students trained directly with weak linear probe supervision shown in solid lines with circles (hue indicates compute of weak supervision).

D.2 Linear probing

In addition to our main finetuning experiments, we also perform weak-to-strong generalization experiments in the linear probing setting. We freeze all weak and strong model parameters, and train new linear classification heads both using ground truth labels and using weak labels. We train linear probes with Adam optimizer (Kingma & Ba, 2014), 10^{-3} learning rate, batch size 128, and no weight decay for 200 epochs, for both weak and strong model training. We do early stopping based on agreement to the weak labels on the validation set and report test accuracy. Results are shown in Figure 23. We observe qualitatively similar generalization compared to the full finetuning case.

Generally, we found the linear probing setting to be very useful to quickly iterate on methods, datasets and ideas. While finetuning provides better results, the qualitative trends in linear probing are similar, and the experiments are much faster and easier to run. For example, we initially found positive results with confidence loss (Section 4.3) and bootstrapping (Section 4.3.1) in the linear probing setting.

E The effects of weak label structure

One challenge in weak-to-strong generalization is the presence of errors in the weak labels. Throughout most of this paper, we consider a particular type of weak error structure: the kinds of errors smaller, capacity-constrained language models make. However, this is not the only type of errors possible.

In this section, we analyze synthetic examples of other kinds of weak label structures, and the implications they have on generalization. Weak model error structure must be considered in relation to the particular strong model at hand. For example, we conjecture that the extent to which the strong model can imitate the weak supervisor may be very important. If we have two strong models of the same performance on the actual task but one is very good at imitating the labels, then we expect that model will generalize less desirably, at least with the naive finetuning method.

In Section 5.1.3 we found that surprisingly the strongest students are imitating the weak supervisor mistakes less than smaller student models in our setting. Since we expect superhuman models to be very good at imitating human supervisor, this may be a major disanalogy. In this section we test cases where the weak supervisor can be imitated easily.

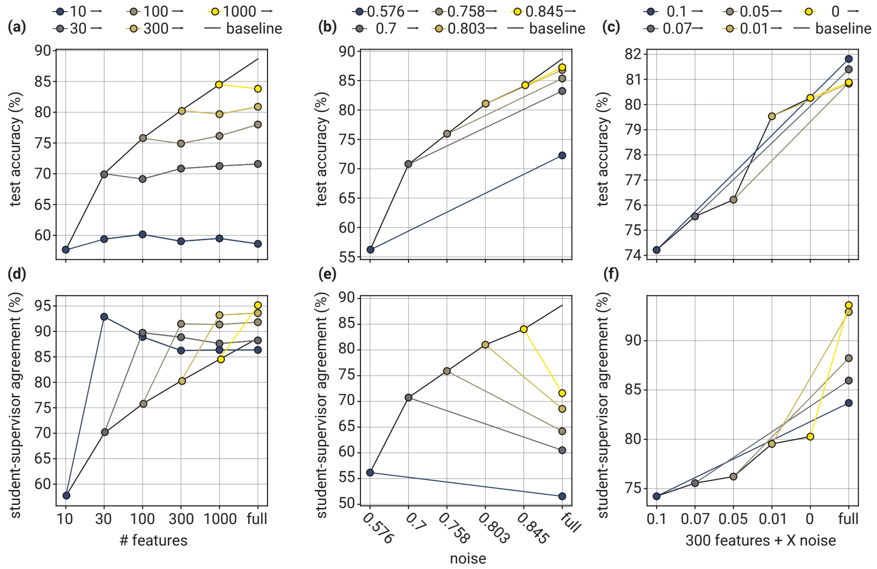


Figure 24: **Synthetic experiment on simulation difficulty.** We consider three types of weak errors in a linear probing setting: **(a,d)** perfectly simulatable, where weak models use a subset of strong model features; **(b,e)** completely unsimulatable, where the weak labels are obtained by applying random noise to the ground truth; **(c,f)** a mixture of the two settings, where label noise is applied to perfectly simulatable weak labels. Top row of panels shows test accuracy and bottom row shows agreement to the weak labels. In addition to weak label accuracy, the structure of mistakes plays a major role in weak-to-strong generalization.

E.1 Synthetic experiments on simulation difficulty

First, we consider a simplified linear probing setting, where we can ensure that the student can perfectly simulate the supervisor predictions by construction. Specifically, we extract a representation $X \in \mathbb{R}^{n \times d}$ of the SciQ dataset using a model of an intermediate size in the GPT-4 family, where n is the number of datapints, and d is the dimensionality of the residual stream (Elhage et al., 2021). We can then consider the family of linear models[10] $\mathcal{M}_k$ where $k \leq d$ by training a linear probe only on the first k features extracted by the model. In particular, for $k = d$ we recover the standard linear probe. By construction for $k_1 \geq k_2$, the model $\mathcal{M}_{k_1}$ can perfectly simulate $\mathcal{M}_{k_2}$.

Next, we can run our standard weak-to-strong generalization experiment, following the setup described in Section 3, using the family of models $\mathcal{M}_k$. We train the weak supervisor models on $10k$ datapoints, and produce hard weak labels on the remaining $13k$ datapoints. We report the results in Figure 24(a,d). In this setting, the simulation is very easy, and we do not observe substantial improvements in the strong student model compared to the supervisor performance. The test agreement values are substantially higher than the weak model accuracy, indicating that the students are overfitting to the supervisor errors. Interestingly, even in this simple setting the agreements are not 100%, likely due to the fact that the student models are trained on finite data, and with light l_2-regularization.

We can also consider the opposite setting: what if the student model cannot simulate the mistakes of the weak teacher at all? Specifically, we generate weak labels by randomly flipping the labels to match the accuracy of the weak models from the previous experiment. As a result, we get weak labels with the same accuracy, but which are completely unpredictable. In Figure 24(b,e), when we train the student model on the these weak labels, we can get substantially higher accuracy than the accuracy of the weak labels. In other words, if the errors of the weak supervisor are completely unpredictable (random) for the student, with enough data we should be able to recover good generalization, substantially exceeding the performance of the supervisor.

[10] We train logistic regression using the default parameters in the `sklearn.linear_model.LogisticRegression` class (Pedregosa et al., 2011) for this experiment.

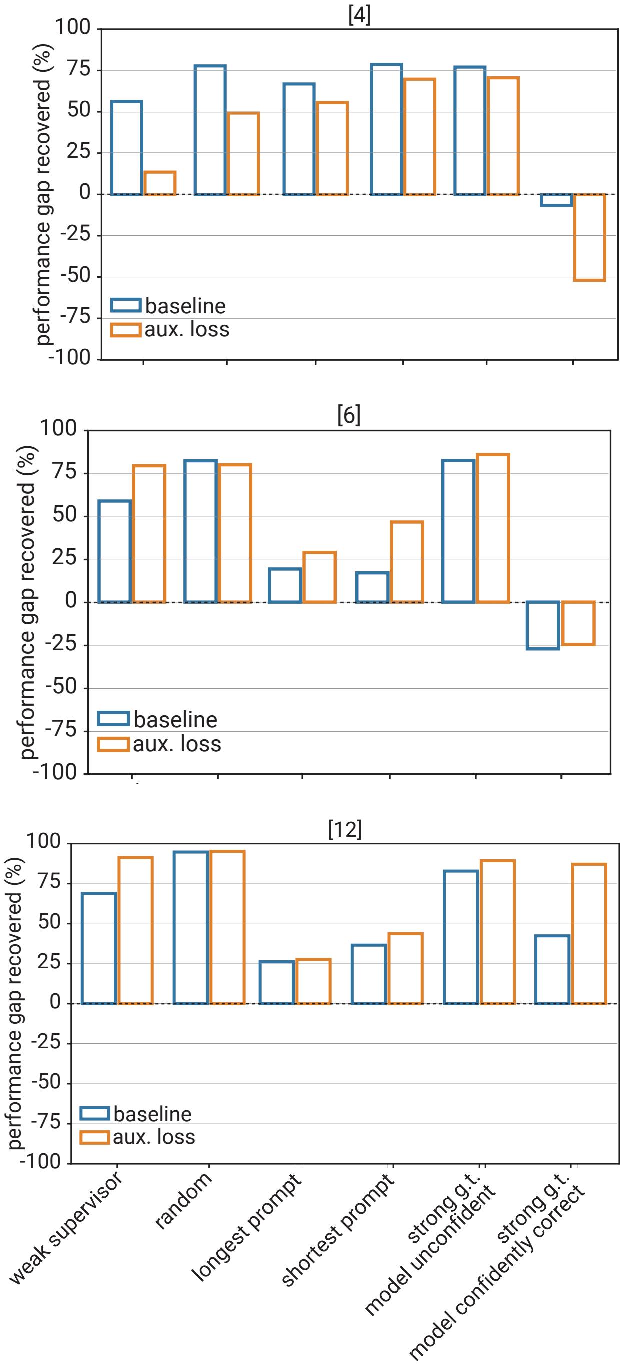


Figure 25: **PGR for weak labels with same accuracy but different error structures.** The inset number in each panel indicates the dataset (compare Figure 12). Weak-to-strong generalization and methods both depend critically on the structure of the weak supervisor errors. While it is trivial to pick error structures that generalize well (for instance, random noise), these error structures are also very disanalogous to the ultimate superalignment setting, where we want to study the structures of human errors.

Finally, in Figure 24(c,f) we consider a mixture of these two settings: we start with a perfectly simulatable weak model $\mathcal{M}_{300}$, and then add various amounts of label noise to the resulting weak labels. By training a strong student model (using all features) on the resulting weak labels, we recover the performance close to the performance of $\mathcal{M}_{300}$.

Discussion of results. The simple experiment in this section suggests that in addition to the weak label accuracy, it is important to consider the *structure of weak errors*. In particular, if the weak errors are extremely easy for the strong model to simulate, the student may not generalize much better than the weak supervisor with naive finetuning on the weak labels. On the other hand, if the mistakes of the weak supervisor are completely unpredictable, the student can denoise the predictions of the supervisor and generalize better. In future work, we believe it is important to consider various types of weak supervision with different structures of mistakes, and build a better understanding of how they affect weak-to-strong generalization.

E.2 Different weak error structure means different generalization

To further explore the impact of different weak error structures, we created several synthetic sets of weak labels for each dataset, all with error rate identical to the weak model's error rate. To construct these labels, we start from ground truth, and then flip a subset of labels to match the accuracy of a particular weak model. We target a few types of error structures, such as pure noise, easy-to-model bias, hard-to-model bias, and adversarial bias.

In particular, we looked at:

1. `weak supervisor`: the baseline — labels are generated in the same way as in the rest of the paper
2. `random`: flip the label of random datapoints
3. `longest prompt`: flip the label of longest datapoints by characters
4. `shortest prompt`: flip the label of shortest datapoints by characters
5. `strong g.t. model unconfident`: flip the label of the datapoints that the strong ceiling model is most unconfident on
6. `strong g.t. model confidently correct`: flips the label of the datapoints that the strong ceiling model is most confidently correct on

Despite all of these weak labelers having the same weak accuracy, we find that the generalization can vary wildly depending on the structure of the weak errors. We report the results in Figure 25.

Furthermore, the dynamics of supervisor-student agreement through training can have qualitatively different behavior (Figure 26). For errors coming from a weak model, we see that there is often initially a period of generalization, followed by a period of overfitting where it learns the weak model's errors. The confidence auxiliary loss mitigates this overfitting. For easy-to-fit error structures such as `longest prompt`, the overfitting happens much faster. For other kinds of errors, such as random noise, we often see that generalization improves throughout: weak errors are not modeled, but the signal from the weak model is.

E.3 Making imitation trivial

One possible major disanalogy in our setup, as discussed in Section 6.1, is the fact that our models are not very good at imitating the weak model[11] (Section 5.1.3), but superhuman models may be very good at imitating humans. It is possible that if the strong model were good at imitating the weak model, then it would generalize substantially less desirably by default.

To test an extreme version of this hypothesis, we create a synthetic setting where the strong model can trivially imitate the weak model very well. In particular, we modify the task by appending "I think this is {`weak_label`}. What do you think?" to every prompt, where weak_label is "correct" or "incorrect" based on the weak model prediction. In this case, the hardened weak label is present in-context, and the simulation is trivial.

[11] Also known as learning the "human simulator" in the terminology of Christiano et al. (2022).

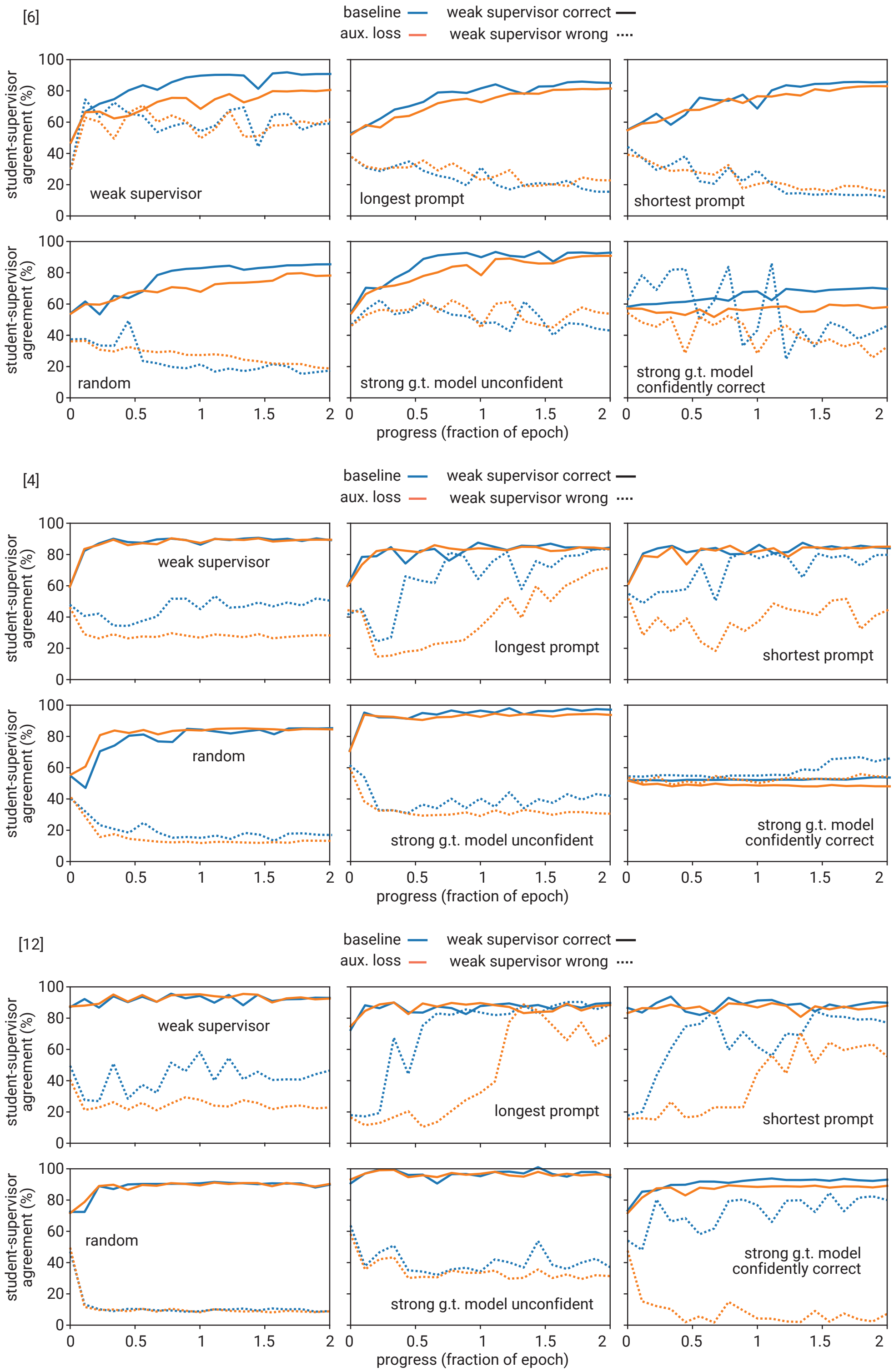


Figure 26: **Training dynamics change for different weak errors.** We show teacher-student agreement for different weak error structures on three datasets. We see that the training dynamics have qualitatively different behavior for different error structures, despite all weak labelers having the same accuracy.

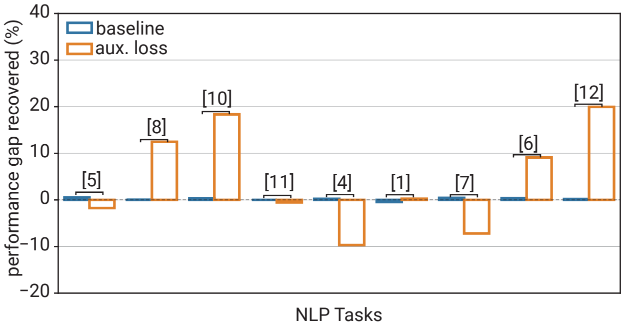


Figure 27: **Generalization when emulating weak labels is trivial.** Very little weak-to-strong generalization occurs if emulating the weak labels is trivial: average PGR across tasks is 0.002 ± 0.003 for baseline, and 0.046 ± 0.108 for aux loss, compared to around 0.2 and 0.8 respectively for the original tasks.

As expected, we find that both the baseline and the confidence loss introduced in Section 4.3 show poor weak-to-strong generalization (Figure 27) in most cases. Interestingly, the confidence loss still improves upon the baseline achieving non-trivial generalization in several tasks.

F How should we empirically study superalignment, methodologically?

What makes a setup good for studying superalignment in the first place, all things considered? Tractability and ease of study are clearly important criteria, but also certainly not the only ones. This question is non-obvious because superalignment is qualitatively different from other machine learning problems: it is a problem we will face in the future, not a problem that we face today. Nevertheless, it is crucial that we solve this problem *before* it becomes serious, as even a single failure of superintelligence misalignment in practice could be catastrophic.

This presents a major methodological challenge: how do we even approach studying a problem that is not yet a problem? How do we make progress on the core difficulties of superalignment? How do we make progress with today's systems, knowing that our efforts will not be wasted by surprising new model capabilities that will inevitably arise in the future (Wei et al., 2022)? We do not claim to have a complete answer to these questions, but we outline some best practices for maximizing our chances of making real progress on superalignment.

Analogous setups. We should construct increasingly analogous empirical setups, and we should enumerate any remaining disanalogies. A setup is analogous if our results on that setup do not rely on assumptions that will break down in the future, making results today likely qualitatively similar to results in the future. Our main evaluation setup, introduced in Section 3, is intended to be more analogous to the superalignment problem. We enumerate some remaining disanalogies with our setup in Section 6.1.

Enumerating assumptions. We should enumerate the key assumptions that our results (either implicitly or explicitly) rely on. Clarifying what assumptions we are making makes it much easier to know when our results might break down. We enumerate our main disanalogies and assumptions in Section 6.1 and Appendix G.3.

Sensitivity analysis. We should evaluate the sensitivity of our results to changes in our assumptions and empirical setup. While we can make informed guesses about the future, we do not know exactly what future models will be like, so it is difficult to entirely trust any particular experimental setup. Validating that our results are robust to many different sets of assumptions can make us substantially more confident our results will transfer to the future superalignment problem. We do some initial sensitivity analysis in Appendix E, and intend to do much more in future work.

Scalable techniques. We should avoid techniques that rely on assumptions that will likely break down for future (superhuman) models. For example, when we do few-shot prompting we are intuitively incentivizing models to predict some useful distribution of human text, whereas when we do finetuning we are intuitively incentivizing a model to output what it knows regardless of how it knows it. This is one of the reasons we focus on finetuning methods in this paper: they are more likely to scale to superhuman models compared to prompting.

Incidental usefulness today. One possible validation that progress on our setup is real would be to show that it is incidentally useful in practice today; while we advocate focusing on the core challenges of superalignment, if our findings are never useful with today's models that would be evidence that we are not on the right track. One example of a near-term practical milestone would be to align GPT-4 on instruction-following tasks using only GPT-3-level supervision; if we could get strong alignment without any humans involved at all, that would make alignment much simpler and cheaper today. However, usefulness today is certainly not sufficient for aligning superintelligence, and in general a common failure mode of empirical alignment research is it prioritizes usefulness today at the expense of analogousness and scalability.

Updating over time. We should update our evaluations and validate past findings as we learn more about what future models will look like. While we focus on the pretrained language model paradigm today, we plan on updating our setup if or when this stops being the dominant paradigm.

G How weak-to-strong generalization fits into alignment

Superintelligent AI systems will be extraordinarily powerful; humans could face catastrophic risks including even extinction (CAIS, 2022) if those systems are misaligned or misused. It is important for AI developers to have a plan for aligning superhuman models ahead of time—before they have the potential to cause irreparable harm.

Our plan for aligning superintelligence is a work in progress, but we believe that weak-to-strong techniques could serve as a key ingredient. In this section we sketch several illustrative possiblities for how we could use weak-to-strong generalization to help align superintelligent systems.

G.1 High-level plan

Leike & Sutskever (2023) propose the following high level plan, which we adopt:

1. Once we have a model that is capable enough that it can automate machine learning—and in particular alignment—research, our goal will be to align that model well enough that it can safely and productively automate alignment research.

2. We will align this model using our most scalable techniques available, e.g. RLHF (Christiano et al., 2017; Ouyang et al., 2022), constitutional AI (Bai et al., 2022b), scalable oversight (Saunders et al., 2022; Bowman et al., 2022), adversarial training, or—the focus of this paper—-weak-to-strong generalization techniques.

3. We will validate that the resulting model is aligned using our best evaluation tools available, e.g. red-teaming (Perez et al., 2022a;b) and interpretability (Ribeiro et al., 2016; Olah et al., 2018; Bills et al., 2023; Li et al., 2023).

4. Using a large amount of compute, we will have the resulting model conduct research to align vastly smarter superhuman systems. We will bootstrap from here to align arbitrarily more capable systems.

The goal of weak-to-strong generalization is to ensure step (2) is solved: align the first model capable of automating machine learning and alignment research. Importantly, this first model will likely be qualitatively superhuman along important dimensions, so RLHF is unlikely to be sufficient (Section 4). If we had a superhuman model, how would we apply weak-to-strong generalization to align it?

G.2 Eliciting key alignment-relevant capabilities with weak-to-strong generalization

There are many different alignment-relevant capabilities we could try to elicit from a superhuman model that could significantly help with alignment, including:[12]

- **Safety:** does a given behavior produced by an AI system risk the safety of human lives or well-being in important ways?
- **Honesty:** is a given natural language statement true or false?
- **Instruction following:** does a given behavior produced by an AI system follow a user's instruction faithfully?
- **Code security:** does some given code have important security vulnerabilities or backdoors? Is it safe to execute it?

In the ideal case, the capability we elicit from the model would be robust enough that we can turn it into a reward model and safely optimize it; future work should assess the feasibility of this approach. At the opposite extreme, we could potentially use the elicited capability as an "oracle" that we can manually query; intuitively, if we had a superhuman oracle model, we may be able to leverage it to help us bootstrap to a more robust alignment solution, even if that oracle is not itself entirely robust.

G.3 Alignment plan assumptions

Many alignment plans which appear different on the surface actually depend on heavily correlated assumptions. For a given alignment plan, it is also often unclear which subproblems the plan attempts to solve, and which subproblems the plan assumes are unlikely to be an obstacle. As a result, we think enumerating assumptions is an important part of making progress on alignment.

In addition to the major disanalogies discussed in Section 6.1, the assumptions we make for an alignment plan based on weak-to-strong generalization include:

- **No deceptive alignment in base models.** We assume that pretrained base models (or the equivalent in future paradigms) will be highly intelligent but not highly agentic (e.g. will not have long-term goals)—and consequently will not be deceptively aligned (Hubinger et al., 2019; Ngo et al., 2022; Carlsmith, 2023) out-of-the-box. Our goal is to elicit the superhuman capabilities of this capable but safe base model, and use those capabilities to create an aligned (possibly agentic) superhuman model.
- **Elicited concepts are sufficiently robust, or do not need to be.** We assume it is either possible to solve alignment using only a small amount of optimization applied to the capabilities we elicit, or that it is possible to make weak-to-strong elicited capabilities sufficiently robust against overoptimization.
- **The concepts we care about are natural to future AGI.** The superhuman base model we apply weak-to-strong generalization to has some "alignment-complete" concept, such as honesty, that is extrapolated in the way we would endorse if we could understand everything the superhuman model understands, and which is natural enough to the model that it is feasible to elicit.
- **Sufficiently gradual takeoff.** Before we have superintelligence, we will have somewhat superhuman models long enough that we can use them to finish solving the full superintelligence alignment problem. We can use it to solve superalignment before it causes recursive self improvement or catastrophic damage.
- **Moderately superhuman models are sufficient to solve alignment.** We assume the first models capable of automating alignment research in practice are moderately superhuman, i.e. in a regime similar to what we study empirically in this work. For example, we may assume that we only need to bridge a weak-strong gap of at most (say) 4 OOMs of effective compute.

[12] Ideally we elicit several related concepts and verify that we get consistent answers between them.

- **No need to solve human values.** We assume we do not need to solve hard philosophical questions of human values and value aggregation before we can align a superhuman researcher model well enough that it avoids egregiously catastrophic outcomes.

This list represents a non-exhaustive set of notable assumptions we often operate under, and we will constantly reassess and update these assumptions over time as we learn more. We *do not* think these are necessarily valid assumptions by default, and believe it is important to validate them, work towards making them true, or mitigate failure modes from them being invalid.

Furthermore, there are a huge number of uncertainties about what future AI systems will look like and exactly how we should align them.