Anchor Points: Benchmarking Models with Much Fewer Examples

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, Douwe Kiela
Stanford University
{rvivek, kawin, diyiy, dkiela}@stanford.edu

Abstract

Modern language models often exhibit powerful but brittle behavior, leading to the development of larger and more diverse benchmarks to reliably assess their behavior. Here, we suggest that model performance can be benchmarked and elucidated with much smaller evaluation sets. We first show that in six popular language classification benchmarks, model confidence in the correct class on many pairs of points is strongly correlated across models. We build upon this phenomenon to propose *Anchor Point Selection*, a technique to select small subsets of datasets that capture model behavior across the entire dataset. Anchor points reliably rank models: across 87 diverse language model-prompt pairs, evaluating models using 1-30 anchor points outperforms uniform sampling and other baselines at accurately ranking models. Moreover, just several anchor points can be used to estimate model per-class predictions on all other points in a dataset with low mean absolute error, sufficient for gauging where the model is likely to fail. Lastly, we present *Anchor Point Maps* for visualizing these insights and facilitating comparisons of the performance of different models on various regions within the dataset distribution. Code will be released at https://github.com/rvivek3/AnchorPoints/.

Figure 1: SST-2 Validation Set Anchor Point Map. The locations of all 872 points are learned using the predictions of 60 randomly-selected source models on SST-2. We then evaluate a held-out model, Falcon-7B, on 30 anchor points (green triangles). The model's predictions on only these 30 points are used to estimate the Falcon-7B predictions on the remaining 842 points with a mean absolute error of 0.09, achieving 92% agreement with the model's true predictions. The anchor points identify regions where the model is weak (red regions). We show the same Anchor Point Map colored by the true Falcon-7B predictions in Figure 8, demonstrating that the model is indeed weak in these areas.

1 Introduction

Language models have unlocked incredible generalization through scaling up parameters and pretraining data. Yet these same systems prove to be brittle, spurring the development of larger, more diverse, and more shrewd benchmarks to reliably assess their behavior (Rajpurkar et al., 2016; Hendrycks et al., 2021; Kiela et al., 2021; Wu et al., 2023). Modern benchmarks typically have on the order of 10^5 validation examples, with $10^3 - 10^4$ per task. Such numbers ensure that validation performance strongly correlates with test performance and that all in-domain regions are captured. But these sizes are often unwieldy for rapid experimentation and do not easily afford interpretability. To compare model configurations, design the most robust prompt, or analyze failure cases, researchers and practitioners must forward-pass (and potentially manually inspect) the development set many times. How small can benchmark development sets be while still capturing model behavior over the full breadth of the benchmark? Surprisingly, very small—just several to a few dozen examples might suffice.

In this work, we investigate the problem of benchmarking model performance and revealing model weaknesses on large datasets with as few

Figure 2: Anchor Points are Micro-Benchmarks, tiny representative subsets of large benchmarks. Correlative structure in the predictions of source models on the large benchmark can be used to extract these points. Each anchor point has a weight corresponding to the fraction of the benchmark it represents. Evaluating models on the anchor points produces a score that rank correlates with performance on the entire benchmark. Anchor Point Maps visualize a given model's likely instance-level performance on all points in the benchmark using only its performance on the anchor points.

evaluation examples as possible, an objective we call *micro-benchmarking*. We propose *Anchor Point Selection*, a technique that finds small evaluation sets that are maximally representative of model behavior over the entirety of a large dataset. We find that anchor points are effective development sets: across 87 diverse language models and prompts, using 1-30 anchor points shows superior performance at ranking model performance over random and embedding-based selection baselines. Moreover, evaluating just a few dozen anchor points can be used to predict the model's instance-level predictions on all other points in the dataset with low average error, sufficient to estimate where models are likely to fail.

Our approach builds upon a simple insight: for many pairs of points from a given dataset, the predicted probability of the correct class strongly correlates across models. Thus, evaluating every model on the entirety of a dataset is redundant. The predictions of existing *source* models on a given dataset are telling about predictions of a new *target* model, even across considerable performance gaps. This mirrors phenomena like *Accuracy on the Line* (Miller et al., 2021) and *Agreement on the Line* (Baek et al., 2022), but at a much more granular scale. Anchor Points can be selected by solving a facility location problem in a space where distance is a positive, monotonically decreasing function of cross-model correlations between points. We further propose to visualize this space using multi-dimensional scaling and U-MAP (McInnes et al., 2018), which we call *Anchor Point Maps*. Anchor Point Maps show what region each anchor point captures and highlight that different models and prompts may struggle on different dataset regions, allowing fine-grained comparisons.

Emphatically, we do not aim to replace large benchmarks, but rather provide cheaper signal about model performance on these benchmarks to help accelerate the development of models and prompts. We also strive to be blunt about the limitations of our technique. Anchor point selection relies on the predictive correlations of source models being consistent with that of target models. We share simple rules of thumb to promote this consistency, but lack rigorous theory to guide source model selection. We lay foundations for further research into model predictive correlations and efficient, interpretable model evaluation.

2 Related Work

Sample-Efficient Model Evaluation We deviate from prior sample-efficient model evaluation literature along key axes. Many works (Kossen et al., 2021a,b; Deng and Zheng, 2021; Corneanu et al., 2020) minimize evaluation *annotation* costs by actively selecting points to annotate for evaluating a given model. In the era of large benchmarks and large models, labeled evaluation examples are widely-available but evaluating all of them is cumbersome. Our technique instead minimizes the number of *forward-passed examples* necessary for reliable model evaluation. Furthermore, our selection strategy is agnostic to the target model(s) being evaluated. The resulting evaluation set is transferable to other language models.

Other works (Rodriguez et al., 2021; Ethayarajh

et al., 2022; Bowman and Dahl, 2020) point out that it is often a minority of points that differentiate the performance of various models. Rodriguez et al. (2021) show that evaluating models on points identified as the most *discriminative* can effectively rank model performance. However, optimizing for discriminability does not result in representative subsets: very easy or very hard points will tend to be excluded, leading to a different distribution. Instead, we optimize for *representativeness* which we show naturally leads to discriminability.

Instance-Level Model Performance Despite being noisy (Zhong et al., 2021), instance-level model predictions are a rich source of information about model behavior. Swayamdipta et al. (2020) present Data Maps, a powerful technique that leverages instance-level predictions to reveal underlying structure in the interplay of models and data points. Various training example regions play distinct roles in guiding a classifier to its solution. Unlike our technique, Data Maps are not used for comparing model performance or isolating distinct regions of the dataset distribution where models are weak.

Ethayarajh et al. (2022) further show that instance-level predictions can be used to quantify how much information a given model can extract from a dataset, providing a formal metric of dataset difficulty that exposes model behavior at the dataset and data point level.

Predictive Correlations across Models Miller et al. (2021) present the *Accuracy on the Line* phenomenon: out-of-distribution (OOD) performance is strongly positively correlated with in-distribution (ID) performance for a wide range of models and distribution shifts. This is unexplained by classical theory which provides only weak bounds relating the two metrics (Ben-David et al., 2018). Baek et al. (2022) further show that the *agreement* of two classifiers of ID data strongly linearly correlates with their agreement on OOD data whenever *Accuracy on the Line* holds. Both phenomena can be used to cheaply estimate model OOD performance. Our findings presented in this work mirror these phenomena at a much more granular scale, i.e., data instances rather than data sets.

Coresets Anchor Points can be interpreted as a coreset, broadly defined as a small "summary" of a large set of data (Feldman, 2020; Guo et al., 2022). Selecting dataset coresets for efficient model training is widely-explored in prior work, performed through various techniques including clustering (Kaufman and Rousseeuw, 2009; Reza Zanjirani Farahani, 2009), gradient-matching (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a), bi-level optimization (Borsos et al., 2020; Killamsetty et al., 2021b), and submodularity-based methods (Kai Wei, 2015). We use a clustering-based approach, but focus on summarizing data for model *evaluation* rather than training.

3 Problem Set-Up

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{D}$ be a language classification benchmark. $\mathcal{D}$ is partitioned into a training split $\mathcal{D}_{train}$ and one or more evaluation splits $\mathcal{D}_{eval}$, each drawn i.i.d from $\mathcal{D}$. We are given M models to evaluate, denoted as the target set $\mathcal{T} = \{\phi_m\}_{m=1}^{M}$. Each unique ϕ_m corresponds to a model fine-tuned on $\mathcal{D}_{train}$ or paired with a specific prompt template that directs the model at solving $\mathcal{D}$. The dataset has an evaluation metric (accuracy, F1-score, etc.) to measure the aggregate performance P_m of each ϕ_m on an evaluation split $\mathcal{D}_{eval}$.

We additionally have access to the instance-level predictions of a source set of N model and prompt template pairs $\mathcal{S} = \{\phi_n\}_{n=1}^{N}$ over the entirety of $\mathcal{D}_{eval}$. $\mathcal{S}$ and $\mathcal{T}$ are disjoint. The predictions occupy an $N \times |\mathcal{D}_{eval}| \times Y$ tensor $P_{\mathcal{S}}$, where Y is the cardinality of our classification task. In practice, source models could be open-source models that can be run locally for free while target models might be closed-source and/or more expensive.

Towards Micro-Benchmarking We aim to extract a development set from a large benchmark that captures the benchmark's broad coverage and reliable ranking power while being as small as possible, improving performance interpretability and evaluation efficiency. We refer to this objective as **micro-benchmarking**. Our technique must 1) acquire a small representative subset of evaluation points from the large benchmark ($X_{acq} \subseteq \mathcal{D}_{eval}, |X_{acq}| \ll |\mathcal{D}_{eval}|$), 2) leverage the predictions of target models $\{\phi_m\}_{m=1}^{M}$ on the subset to produce scores $S_{1...M}$ that correlate as much as possible with model performances $P_{1...M}$ on the entire dataset $\mathcal{D}_{eval}$, and 3) estimate instance-level performance on untested points from $\mathcal{D}_{eval}$.

4 Micro-Benchmarking Approach

We measure how well an example (x_1, y_1) represents another example using the Pearson correlation of correct class confidence in the two examples

measured across the predictions of the source models in $\mathcal{S} = \{\phi_n\}_{n=1}^{N}$. However, model confidences are bounded by [0, 1] and thus can only follow a linear trend for a bounded range. Akin to Miller et al. (2021), we take the logit transform of the confidences to scale the axes from [0,1] to $[-\inf, +\inf]$ prior to computing correlations. We denote this composite operation as $CORR_{\mathcal{S}}$. The task of selecting a X_{acq} that is maximally representative of X_{eval} for all models in $\mathcal{S}$ reduces to solving a facility location problem that maximizes the correlation between the selected and the remaining points:

$$\min_{X_{acq}} \sum_{i=1}^{B} \sum_{x_{eval}^{j} \in C_i} 1 - CORR_{\mathcal{S}}(x_{acq}^{i}, x_{eval}^{j}) \quad (1)$$

where C_i is the set containing all x_{eval} that are more strongly correlated with x_{acq}^{i} than any other x_{acq}^{j}. This objective makes the assumption that our estimate for $\phi(x_{eval})$ leverages only the most strongly correlated $\phi(x_{acq}^{i})$. This is appropriate as we do not find that estimating $\phi(x_{eval})$ using multiple $\phi(x_{acq}^{j})$ is helpful. We efficiently solve our objective using an efficient implementation of K-Medoids (Kaufman and Rousseeuw, 2009).

4.1 Anchor Point Techniques

We present two techniques for leveraging anchor points to benchmark model performance with minimal evaluation examples.

Technique 1: Anchor Point Predictor We propose the Anchor Points Predictor, an ensemble of univariate linear regression models that use model predictions on each anchor point to estimate predictions on all other points in X_{eval}. Specifically, anchor point $\phi(x_{acq}^{i})$ is used to estimate all $\phi(x_{eval}) \in C_i$. These instance-level prediction estimates can then be used to detect regions of model weaknesses (see Section 4.1) and compute an estimate of any performance metric for the target model. This technique requires that the trend lines fitted to source model predictions explains the variance in target model predictions well. We explore when this is the case in Appendix A. Algorithms 1 and 2 show pseudocode for fitting and making predictions with the Anchor Point Predictor Model.

Technique 2: Anchor Point Weighted Score Rather than attempting to estimate exact model performance at the instance-level, we can aim to produce a score for each target model that highly correlates with the model's performance on $\mathcal{D}_{eval}$. Using the predictions of our source model set $\mathcal{S}$, we select $N \in 1...B$ anchor points according to Equation 1. Each of these points $x_{acq,i}$ strongly correlates with a subset of the untested points in $\mathcal{D}_{eval}$, namely all $x_{eval}^{j} \in C_i$. We propose the Anchor Point Weighted (APW) score, a weighted average of the model's correct class probability predictions on the anchor points with weights proportional to cluster size.

$$APW(\phi_i) = \frac{1}{|D|} \sum_{i=1}^{B} |C_i| * \phi_i(x_{acq}^{i})[y_i] \quad (2)$$

Anchor Point Maps To highlight the insights provided by anchor points, namely model strengths and weaknesses in distinct regions of a dataset, we propose to visualize the cross-model correlative space from which anchor points are drawn. We compute pairwise correlations between all points using $CORR_{\mathcal{S}}$, creating a $|\mathcal{D}_{eval}| \times |\mathcal{D}_{eval}|$ correlation matrix C. We then represent our distance matrix D as a positive, monotonically decreasing function of correlation: $D = 1 - C$. Finally, we cast the points to a high-dimensional continuous space using Multi-Dimensional Scaling on D and project the space to two components with U-MAP (McInnes et al., 2018). This technique visualizes our dataset in a space where a model's performance on each point generalizes to the local neighborhood of that point.

5 Predictive Correlations

To motivate our approach, we first show that language models make predictions with consistent structure. For many pairs of points from a given dataset, the predicted probability of the correct class strongly correlates across models.

Experimental Set-Up We obtain a diverse set of language models from HuggingFace and the OpenAI API: 27 BERT-family models, 11 non-instruction-tuned GPT-family models, 5 instruction-tuned GPT-family models, and 5 GPT-3/3.5 variants (all listed in Appendix B). We denote these model sets as the BERT-family, GPT-family, InstructGPT-family, and OpenAI-family respectively. For six datasets from the GLUE Benchmark (Wang et al., 2019), we finetune each BERT-family model on the training split $\mathcal{D}_{train}$ and obtain its predicted probability of the correct class on

(a) Pairwise Correlation Matrix of ALL-Family Correct Class Confidences on 50 QQP Examples. Rich structure indicates the relatedness of various examples. Many are strongly correlated.

(b) Correct Class Confidences of 87 Language Model Predictions from all model families on Two Selected QQP examples. A model correctly classifying one example is predictive of the model correctly classifying the other. This phenomenon is very common.

Figure 3: Predictive Correlations at the Instance-Level Across Language Models

each instance in the validation split $\mathcal{D}_{eval}$. For the other model families we zero-shot prompt for the same datasets using three multiple-choice prompts per dataset (Appendix B). We obtain the predicted probability of each class by normalizing over the logits corresponding to each class for the first generated token. Thus, we obtain a prediction matrix P for each model family and dataset with size $N_{subjects} \times |\mathcal{D}_{eval}|$ matrix where $N_{subjects} = 27$ for P_{BERT}, 30 for P_{GPT}, 15 for P_{IGPT}, 15 for P_{OAI}, and 87 for their union P_{ALL}.

Model predictions on pairs of examples are linearly correlated across models. The prediction matrices are approximately low-rank for all tasks (Table 1). This is due to a simple phenomenon: let $\phi_n(x_1)[y]$ denote the probability mass outputted by ϕ_n on $x_1 \sim \mathcal{D}_{eval}$ for a given class y. We see that $\phi_n(x_1)[y]$ often linearly correlates with the model's prediction on a different instance $\phi_n(x_2)[y]$ across models, i.e. $\phi_n(x_2)[y] \approx w\phi_n(x_1)[y] + b$, where w and b can be found by fitting a trend line through the correct class confidences of all source models.

Figure 3 visualizes this phenomenon. We first observe that the correlation matrices of model predictions on the same evaluation set show rich structure. Figure 3(b) shows an QQP example pair that is strongly positively correlated across models from all families. This suggests that evaluating a model on both examples is redundant. This phenomenon is wide-spread: it holds for many pairs of examples across the six GLUE tasks and 87 widely-used language models and prompts.

Task	BERT	GPT	IGPT	OAI	ALL
MNLI	$2_{0.1}$	$1_{0.07}$	$3_{0.07}$	$4_{0.1}$	$3_{0.09}$
SST-2	$1_{0.08}$	$2_{0.05}$	$2_{0.09}$	$3_{0.1}$	$2_{0.09}$
QQP	$2_{0.06}$	$2_{0.06}$	$2_{0.1}$	$4_{0.09}$	$3_{0.09}$
RTE	$2_{0.07}$	$2_{0.06}$	$2_{0.08}$	$3_{0.1}$	$2_{0.08}$
MRPC	$2_{0.08}$	$4_{0.1}$	$3_{0.09}$	$7_{0.1}$	$7_{0.1}$
QNLI	$1_{0.08}$	$2_{0.08}$	$3_{0.09}$	$7_{0.09}$	$3_{0.09}$

Table 1: Approximate matrix rank of the correct class probability predictions of four model families on six GLUE task validation sets. We compute low-rank matrix approximations for 27 BERT-family predictions, 30 GPT-family predictions, 15 Instruction-tuned GPT-family predictions (IGPT), 15 OpenAI-family predictions (OAI), and all 87 model predictions (ALL) together. The approximations achieve very low mean absolute error (indicated by the subscripts), despite their ranks being considerably smaller than the number of models in each family.

Predictive Correlations Tend to Generalize Across Diverse Models Figure 5 shows how well the strong trends from each family transfer to other families. Errors are lowest in the bottom row, suggesting that using source models that span all model families result in the most generalizable trend lines. We also observe that the predictive correlations of the BERT-family do not generalize well to other model families. Overall most errors are reasonably low, suggesting that predictive correlations tend to generalize across diverse models.

	Random	Random Mean	Pretrained	Pretrained Weighted	Fine-Tuned	Fine-Tuned Weighted	AP Weighted	AP Predictor
Datasets	Exact	Corr	Exact	Corr	Exact	Corr	Corr	Exact
SST-2	0.685	0.705	0.734	0.725	0.730	**0.787**	0.757	0.727
QQP	0.669	0.678	0.189	0.233	0.766	**0.770**	0.756	0.701
RTE	0.366	0.308	0.143	-0.052	0.354	0.275	**0.483**	0.462
QNLI	0.321	0.331	0.192	0.294	0.127	0.144	**0.439**	0.303
MRPC	0.687	0.679	0.528	0.604	0.641	0.681	**0.726**	0.716
MNLI	0.438	0.433	0.177	0.166	0.523	0.453	**0.544**	0.517
Average.	0.528	0.522	0.327	0.328	0.523	0.518	**0.612**	0.571

Table 2: Area Under the (Kendall's τ) Correlation Curve from 1 to 30 points for ranking 77 language models at 6 GLUE tasks. We randomly select 10 models to be source models for the AP methods and rank the remaining 77 models, averaging over 100 randomized runs. AP Weighted and AP Predictor show significant gains, which are most dramatic at smaller evaluation sets (Figures 12 and 13). "Exact" indicates the method generates a score that is intended to approximate the true aggregate performance directly, while "Corr" indicates the method generates a score intended only to rank correlate with true performance. The best score is **bolded** and second best score is underlined. Table 4 shows these results broken down by model family.

6 Sample-Efficient Model Evaluation

We now evaluate the anchor point techniques and compare against baselines for representative subset selection. Uniform random sampling is a natural baseline. Since correct class model confidence is a rich signal, we use the mean correct class confidence of randomly selected examples as another baseline. We also evaluate K-Medoids sampling over the embedding spaces of a generic sentence encoder– SentenceBERT (Reimers and Gurevych., 2019) – and an encoder fine-tuned on the dataset– the *CLS* token of bert-base-uncased (Devlin et al., 2018). We use aggregate model performance on the selected point as the estimate of aggregate model performance on the entire dataset.

We further consider weighted variants of the embedding baselines. Akin to Anchor Points Weighted, we produce a score estimate for each model by computing a weighted average of the correct class probabilities assigned to each selected point. The weights are proportional to the size of each selected point's cluster and sum to 1.

We perform model performance ranking experiments within each model family as well as across the union of all families. We use $|\mathcal{S}| = 5$ source model predictions to which we fit anchor points for the InstructGPT-family and OpenAI-family experiments. For the BERT-family, GPT-family, and ALL-family, we use $|\mathcal{S}| = 10$ model predictions. The partitioning of source and target models is randomized over 100 runs for each validation set size and model family setting.

To assess the ranking performance of each technique, we compute the Kendall rank correlation coefficient (Kendall's τ) for a range of evaluation set sizes and report the resulting **Area Under the Correlation Curve** (AUCC) from 1 to a maximum budget B. We then normalize AUCC by dividing by the best possible area. In our experiments we use $B = 30$, the small data regime for which it is reasonable for practitioners to manually inspect predictions.

Anchor Points Show Competitive Performance at Ranking Models with Small Evaluation Sets Table 2 shows the Kendal's τ AUCC of our methods for ranking 77 models belonging to all model families. These curves are shown in Figures 12 and 13. Table 4 shows performance within each model family. Overall, we observe that Anchor Points Weighted outperforms random selection in 29 of 30 settings and Anchor Points Predictor outperforms random selection in 27 of 30 settings. Anchor Points Weighted proves to be the strongest among all techniques, followed by Anchor Points Predictor.

Anchor Point Weighted Correlations Are Reliable, Unlike Strong Contenders We observe that the Anchor Point Weighted (APW) correlations tend to be more reliable across evaluation set sizes and test settings than APP as well as our baselines. We suspect APW is more reliable than APP because it makes a weaker assumption that source model predictions simply follow similar correlations as target model predictions rather than closely matching the exact regression line fitted to the source model predictions.

(a) QQP Anchor Point Map colored by the estimated predictions of deberta-v3-base. Thirty anchor points are used to estimate model predictions on all other points with low error (MAE = 0.11).

(b) QQP Anchor Point Map colored by the estimated predictions of text-davinci-003 with prompt 1. Thirty anchor points are used to estimate model predictions on all other points with low error (MAE = 0.16).

Figure 4: Anchor Point Map for 1000 QQP examples projected with UMAP. The map is computed using the predictions of 60 randomly-selected source models that do not include deberta-v3-base or text-davinci-003 (prompted with QQP prompt 1). We then evaluate the two models on 30 anchor points (green triangles) and color all other points according to each models' estimated predictions using the anchor points. This analysis highlights that anchor points achieve efficient detection of model weaknesses and do not rely on these weaknesses being consistent across models. Comparisons to the true predictions are shown in Figure 10.

We observe inconsistent and sometimes erratic behavior in our embedding-based baselines. Despite Fine-tuned and Fine-tuned Weighted baselines collectively achieving the best performance in 10 settings, these baselines prove to be inconsistent: Fine-tuned and Fine-tuned Weighted perform worse than random in 12 and 10 settings, respectively. We suspect this is because language model embedding spaces are not necessarily smooth and are likely to be inconsistent across models.

Anchor Points Efficiently Estimate Model Predictions The Anchor Point Predictor can estimate model performance at the instance-level over the entire dataset. Table 3 shows the average agreement between Anchor Point estimated instance-level model predictions and true model predictions using various numbers of anchor points and randomly selected source models. The estimates of a small number of anchor points achieve high average agreement with true model predictions. However, the source model correlations do not generalize well to all target models, resulting in lower agreement on these outlier models and thus larger standard deviations in Table 3. Strategies for intelligently selecting source models are a promising direction to resolve poor anchor point generalization, which we discuss further in Appendix A.

Anchor Point Maps Visualize Where Models Generalize Anchor Point Maps reveal the extent to which models learn distinct regions of dataset distributions. Model performance on each sample correlates with performance in the sample's neighborhood, allowing the Anchor Point Predictor to estimate where models will fail without evaluating the entire dataset. Figure 1 shows the SST-2 validation set mapped by the predictions of 60 source models and then colored by estimated Falcon-7B predictions using thirty anchor points. These estimated predictions are quite faithful to the model's true predictions (achieving 0.09 MAE and 92% agreement), revealing regions where the model is weak. This localization of model behavior starkly contrasts with typical language embedding spaces, where model performance tends to be non-localized (Figure 9). However, like embedding spaces, Anchor Point Maps are semantically meaningful: similar examples are close together and different classes naturally cluster apart.

Anchor Points Accurately Predict Diverse Model Behavior Figure 4 shows two Anchor Point Maps of 1000 QQP points colored by the (faithful) estimated predictions of deberta-v3-base and text-davinci-003 using 30 anchor points. The two models are weak in distinct regions of the dataset. Despite deberta-v3-base achieving

		Number of Anchor Points				
Dataset	Validation Set Size	1	5	10	30	50
SST-2	872	$0.74_{0.18}$	$0.85_{0.12}$	$0.85_{0.11}$	$0.86_{0.10}$	$0.86_{0.10}$
QQP	40430	$0.73_{0.17}$	$0.83_{0.14}$	$0.84_{0.13}$	$0.85_{0.13}$	$0.85_{0.12}$
RTE	277	$0.81_{0.20}$	$0.83_{0.18}$	$0.82_{0.18}$	$0.84_{0.16}$	$0.84_{0.16}$
QNLI	5463	$0.73_{0.19}$	$0.74_{0.17}$	$0.74_{.16}$	$0.74_{0.15}$	$0.75_{0.15}$
MRPC	408	$0.77_{.16}$	$0.80_{.15}$	$0.80_{.14}$	$0.81_{.14}$	$0.82_{.13}$
MNLI	9815	$0.70_{.24}$	$0.79_{.20}$	$0.79_{.19}$	$0.80_{.18}$	$0.81_{.18}$
Average.	—	0.75	0.81	0.81	0.81	0.82

Table 3: Agreement of true target model predictions and estimated target model predictions using various numbers of anchor points (higher is better). We define agreement as the percentage of points for which the anchor point predictor estimates the model will predict the class that the model actually does predict. We randomly select 60 models to be source models and use the Anchor Point Predictor to estimate the predictions of the remaining 27 models, averaging over 10 randomized runs. All average agreements are quite high and trend upward with more anchor points, indicating stronger generalization to new models. Variance in agreements decreases moderately but remains large, indicating the presence of outlier models.

stronger performance overall, it fails in a region of the negative class cluster where text-davinci-003 is mostly correct. This highlights that anchor point generalization does not rely on models making similar predictions, but rather only that models follow similar predictive correlations. We explore anchor point generalization further in Appendix A.

7 Limitations

Anchor Points Show Diminishing Returns For many settings, APW correlation begins to plateau as the evaluation set size grows to 100 points. For 19 of the 30 settings, the random selection curve eventually reaches the APW curve at some evaluation set size smaller than 100 points. This is natural: a randomly-selected evaluation set is sufficiently representative when sufficiently large. These diminishing returns occur more quickly for simpler datasets (e.g. SST-2) and less so for more complex datasets (e.g. MNLI), reflecting that more complex distributions require larger evaluation sets to be well-represented. However, it is surprising that random selection eventually *surpasses* the APW curve in 5 of these 19 settings. This is a result of the APW curve having a tighter upper bound than random selection: in the limit where $|\mathcal{X}_{acq}| = |\mathcal{X}_{eval}|$, instance-level confidence in the correct class averaged over the entire dataset does not perfectly correlate with accuracy over the entire dataset. This upper bound does not exist for APP.

Anchor Points Do Not Always Generalize In order for anchor points and anchor point maps to generalize, the predictive trends of source models must be consistent with that of target models. While Table 4 shows that anchor points often transfer both within and across model families, the correlations of randomly selected source models do not necessarily generalize to all target models (e.g. Figure 11). Moreover, while Figure 5 suggests using diverse source models results in the best generalization, we currently lack rigorous theory to guide source model selection for evaluation of a given set of target models.

Anchor Points Can be Further Optimized In its current form, Anchor Point Maps require pairwise correlations between all points in the dataset. This is computationally expensive for huge datasets. Multi-dimensional scaling with missing values or correlation matrix completion would dramatically decrease these costs.

8 Conclusion and Future Work

We present Anchor Point Selection, a technique that finds maximally-representative subsets from large datasets which can be used to efficiently rank models and estimate model behavior over the entire dataset. We also present Anchor Point Maps, a tool to visualize how well models generalize across various regions of datasets.

Important future work includes developing better theory underlying when models share predictive correlations and designing intelligent source model selection strategies to ensure anchor point generalization to desired sets of target models.

The fact that diverse language models make highly correlated predictions on many pairs of examples suggests that modern benchmarks contain a large amount of redundant examples. An interest-

ing future direction would be using predictive correlations to guide benchmark development by selecting fewer redundant examples, leading to more diverse and difficult benchmarks.

The potential to extend Anchor Point Maps and Anchor Point Selection to tasks beyond language classification is exciting. Perhaps other continuous performance metrics are strongly correlated between examples, such as BLEU scores, log probabilities of sequences, or the rewards given by a reward model. Exploiting such phenomena could allow a dramatic reduction in the number of examples that modern language models must be evaluated on to gauge their performance and discover their weaknesses.

References

Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. 2021. Exploring the limits of large scale pre-training. arXiv.

Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-Jian Jiang, and Alexander M. Rush. 2022. Promptsource: An integrated development environment and repository for natural language prompts. Association for Computational Linguistics.

Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. 2022. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift.

Elisa Bassignana, Max Müller-Eberstein, Mike Zhang, and Barbara Plank. 2022. Evidence > intuition: Transferability estimation for encoder selection.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2018. A theory of learning from different domains. Springerlink.

Zalán Borsos, Mojmír Mutný, and Andreas Krause. 2020. Coresets via bilevel optimization for continual learning and streaming.

Samuel R. Bowman and George E. Dahl. 2020. What will it take to fix benchmarking in natural language understanding? NAACL.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. EMNLP.

Ciprian Corneanu, Meysam Madadi, Sergio Escalera, and Aleix Martinez. 2020. Computing the testing error without a testing set.

Weijian Deng and Liang Zheng. 2021. Are labels always necessary for classifier accuracy evaluation?

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

William B Dolan and Chris Brockett. 2004. Automatically constructing a corpus of sentential paraphrases. International Workshop on Paraphrasing.

Zeyad Ali Sami Emam, Hong-Min Chu, Ping-Yeh Chiang, Wojciech Czaja, Richard Leapman, Micah Goldblum, and Tom Goldstein. 2019. Active learning at the imagenet scale.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with v-usable information. ICML.

Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021. On statistical bias in active learning: How and when to fix it.

Dan Feldman. 2020. Introduction to core-sets: an updated survey.

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

Chengcheng Guo, Bo Zhao, and Yanbing Bai. 2022. Deepcore: A comprehensive library for coreset selection in deep learning.

Kelvin Guu, Albert Webson, Ellie Pavlick, Lucas Dixon, Ian Tenney, and Tolga Bolukbasi. 2023. Simfluence: Modeling the influence of individual training examples by simulating training runs. arXiv.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *NAACL*.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning.

Jeff Bilmes Kai Wei, Rishabh Iyer. 2015. Submodularity in data subset selection and active learning.

Leonard Kaufman and Peter J Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in nlp. *NAACL*.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. 2021a. Grad-match: Gradient matching-based data subset selection for efficient deep model training.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. 2021b. Glister: Generalization based data subset selection for efficient and robust learning.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021a. Active surrogate estimators: An active learning approach to label-efficient model evaluation.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021b. Active testing: Sample-efficient model evaluation.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles.

David J. C. MacKay. 1992. Information-based objective functions for active data selection. *Neural Computation*.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv.

Barbara Plank Mike Zhang. 2020. Cartography active learning. *EMNLP*.

John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. 2021. Accuracy on the line: On the strong correlation between out-of-distribution and in-distribution generalization.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. arXiv.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, , and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. EMNLP.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning.

Masoud Hekmatfar Reza Zanjirani Farahani. 2009. *Facility location: concepts, models, algorithms and case studies*. Springer-Verlag Berlin Heidelberg.

Pedro Rodriguez, Joe Barrow, Alexander Miserlis Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change nlp leaderboards? *ACL*.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach.

Ryan Smith, Jason Fries, Braden Hancock, and Stephen Bach. 2022. Language models in the loop: Incorporating prompting into weak supervision.

Richard Socher, Alex Perelygin, Jean Wu, Christopher D Manning Jason Chuang, Andrew Ng, , and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment tree-bank. Conference on Empirical Methods in Natural Language Processing.

Aarohi Srivastava and Abhinav Rastogi. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. In *arxiv*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2021. Beyond marginal uncertainty: How accurately can bayesian regression models estimate posterior predictive correlations?

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Eric Wallace, Mitchell Stern, and Dawn Song. 2020. Imitation attacks and defenses for black-box machine translation systems. *EMNLP*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. ICLR.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. Association for Computational Linguistics: Human Language Technologies.

Zhengxuan Wu, Christopher Manning, and Christopher Potts. 2023. Recogs: How incidental details of a logical form overshadow an evaluation of semantic interpretation. In *arxiv*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023. Data selection for language models via importance resampling. arXiv.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022a. Automatic chain of thought prompting in large language models. arXiv.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models.

Ruiqi Zhong, Dhruba Ghosh, Dan Klein, and Jacob Steinhardt. 2021. Are larger pretrained language models uniformly better? comparing performance at the instance level.

A Anchor Points Generalization Analysis

We perform further investigations to assess under what conditions the predictive correlations of source models generalize well to target models. Figure 6 shows how well strong positive and strong negative trends in source model predictions generalize to target model predictions. We observe that positive trends generalize more reliably than negative trends. Negative trends suggest that models struggle to correctly label both of two examples correctly: greater confidence in the correct class in one example correlates with lesser confidence in the correct class of the other example. We hypothesize that this characteristic is a weakness of some models that does not generalize to stronger models, leading to poor generalization. Note that Equation 1 selects points with strong positive trends, avoiding this issue.

Figure 7 shows the correlation matrix of 50 randomly selected SST-2 points, computed across the predictions of all 87 models. We observe that nearly all correlations are positive, with a notable exception. This exception is a highly contentious example: "we root for (clara and paul) , even like them , though perhaps it 's an emotion closer to pity, " labeled as positive sentiment. We observe in Figure 7(b) that prompted language models tend to follow a negative trend between this example and others, while fine-tuned BERT models do not. This suggests a distinction in how fine-tuned and prompted models fit the task distribution. Notably, it highlights that different model families may have predictive correlations that do not generalize to other families. Further analysis of these properties could lead to the design of intelligent source selection schemes for evaluating a given set of target models.

B Models

Open-source models can be found at https://huggingface.co/models. Closed-source models can be found at https://platform.openai.com/docs/models/overview. We use three randomly selected prompts per dataset from PromptSource: https://github.com/bigscience-workshop/promptsource. The specific prompts will be shared at https://github.com/rvivek3/AnchorPoints/.

Figure 5: Transfer table showing the mean absolute error of trend lines fit to the instance-level predictions of one model family and used to estimate the instance-level predictions of another family. Along the diagonal, we partition the family into source and target halves randomly. Results are averaged over 1000 points pairs from each of our 6 GLUE tasks. Point pairs are randomly selected from all pairs having a Pearson correlation greater than +0.8 within the source model predictions.

Figure 6: Mean Absolute Error Distribution for regression lines fit to source model predictions with moderate-to-strong fit ($R^2 > 0.64$) and used to predict target model predictions. Generalization is worse for trends with negative slopes.

B.1 (Open Source) BERT Family Models

The following set of models is included in the BERT family of models:

1. 'Bio_ClinicalBERT'
2. 'albert-base-v2'
3. 'bart-base'
4. 'bart-large'
5. 'bert-base-cased'
6. 'bert-base-multilingual-cased'
7. 'bert-base-uncased'
8. 'bert-large-cased'
9. 'bert-large-uncased'
10. 'bert-mini'
11. 'bert-tiny'
12. 'biobert-v1.1'
13. 'deberta-base'
14. 'deberta-large'
15. 'deberta-v3-base'
16. 'deberta-v3-xsmall'
17. 'distilbert-base-cased'
18. 'distilbert-base-uncased'
19. 'electra-base-discriminator'
20. 'legal-bert-small-uncased'
21. 'roberta-base'
22. 'scibert_scivocab_uncased'
23. 'sentence_bert'
24. 'sentiment-roberta-large-english'
25. 'twitter-roberta-base'
26. 'xlm-roberta-base'
27. 'xlm-roberta-large'

(a) Pairwise Correlation Matrix of 87 Model-Prompt Pairs Correct Class Confidences on 50 SST-2 Examples. Rich structure indicates the relatedness of various examples. Notably, a dark bar at index 13 indicates an example that tends to negatively correlate with all other examples.

(b) Gold Label Predictions of 87 Models on Two Selected SST-2 sentence pair examples. Example 1 (x-axis) corresponds to the dark bar in 7(a), a highly contentious example. Prompted models follow a negative trend while BERT models follow no trend, highlighting the distinct behavior of fine-tuned vs. prompted models.

Figure 7: Predictive Correlations at the Instance-Level Across Language Models: SST-2 Case Study

B.2 (Open Source) GPT Family Models

The following set of models is included in the open source GPT family of models:

1. 'Cerebras-GPT-1.3B'
2. 'Cerebras-GPT-111M'
3. 'Cerebras-GPT-256M'
4. 'bloom-1b7'
5. 'gpt-neo-1.3B'
6. 'gpt-neo-125m'
7. 'gpt2-large'
8. 'gpt2-medium'
9. 'gpt2'
10. 'openai-gpt'

B.3 (Open Source) Instruct-GPT Family Models

The following set of models is included in the open source InstructGPT family of models:

1. 'RedPajama-INCITE-Instruct-7B-v0.1'
2. 'bloomz-3b'
3. 'falcon-7b'
4. 'mpt-7b-instruct'
5. 'mt0-small'

(Closed-Source) OpenAI Models The following set of models is included in the closed source OPENAI (GPT3/3.5) family of models:

1. 'text-ada-001'
2. 'text-babbage-001'
3. 'text-curie-001'
4. 'text-davinci-002'
5. 'text-davinci-003'

C Anchor Point Algorithms

The algorithms for fitting the Anchor Points Predictor to a set of source model predictions and estimating the predictions of a target model are shown below. Note that in the case of multi-class classification with K classes, the gold label predictions can be substituted for predictions of all points on a given class y_k. The algorithm can then be run K times to estimate model predictions on each class y_k on each point.

(a) SST-2 Anchor Point Map Colored with Estimated Falcon-7b (Prompt 3) Predictions. The estimates achieve an MAE of 0.09 and agreement of 92%.

(b) SST-2 Anchor Point Map Colored with True Falcon-7b (Prompt 3) Predictions.

Figure 8: SST-2 Validation Set Anchor Point Map. The locations of all 872 points are learned using the predictions of 60 randomly-selected source models on SST-2. We then evaluate Falcon-7b (held-out) on 30 anchor points, shown by green triangles in 8(a). Next, the model's predictions on only these anchor points is used to estimate the models' predictions on the remaining 842 points, with a mean absolute error of 0.09 and 92% agreement with the true predictions. The estimates are also shown in 8(a). achieving 0.09 MAE and 92% agreement with the models' true predictions. Finally, we color the Anchor Point Map with Falcon-7B's true predictions in 8(b). The estimated predictions successfully identify regions where the model is weak (red regions).

Algorithm 1 Anchor Points Predictor Fit

$k \leftarrow$ number of anchor points
$P \leftarrow NxD$ array of N source models' normalized gold label predictions on D points
$P \leftarrow logit(P)$ ▷ take logit transform of predictions
$C \leftarrow$ corrcoef(P) ▷ $D \times D$ correlation matrix
$AP \leftarrow$ K-MEDOIDS(1 - C, K) ▷ length K array of anchor point indices
$N \leftarrow$ argmax(C[:, AP], axis = 1) ▷ length D array of nearest anchor point to each point in D
$T \leftarrow \{0...D-1\} \setminus AP$ ▷ length D - K array of test point indices
$M \leftarrow$ empty $(D-K)$ array ▷ To store slopes
$B \leftarrow$ empty $(D-K)$ array ▷ To store biases
for i, test $\in$ enumerated T **do**
  anchor $\leftarrow AP[N[i]]$ ▷ Get index of nearest anchor point to this test point
  m, b $\leftarrow$ LinearRegression(P[:,anchor], P[:,test]) ▷ Slope and bias of trend line
  M[i] $\leftarrow$ m
  B[i] $\leftarrow$ b
end for
  return AP, N, T, M, B

(a) 1000 QQP Points Embedded with Sentence-BERT and colored with the correct class confidence of bert-large-cased.

(b) 1000 QQP Points Embedded with the CLS token of bert-base-uncased (fine-tuned on QQP). Points are colored with the correct class confidence of bert-large-cased.

(c) 1000 QQP Points Visualized with the Anchor Point Map using 60 source models and colored with the correct class confidence of bert-large-cased.

Figure 9: Comparison of model performance trends within various embedding spaces. Only the Anchor Points Map results in clean localization of model performance where a held-out model's performance on each point correlates with its performance on neighboring points. This allows fine-grained comparison of how different models perform on the same dataset distribution.

Algorithm 2 Anchor Points Predictor Predict

Require: AP, N, T, M, B ▷ Returned by Anchor Points Fit
  P ← length K array of a target model's gold label predictions on the K anchor points
  $P \leftarrow logit(P)$ ▷ take logit transform of predictions
  Y ← empty $(D - K)$ array ▷ To store estimated target model predictions
  for i, test ∈ enumerated T **do**
    anchor ← $AP[N[i]]$ ▷ Get index of nearest anchor point to this test point
    y ← M[i] * P[anchor] + B[i] ▷ Prediction step
    Y[i] ← $expit(y)$ ▷ Inverse of the logit transform
  end for
    return Y

(a) QQP Anchor Point Map Colored with Estimated deberta-v3-base predictions. The estimates achieve an MAE of 0.11 and agreement of 89%.

(b) QQP Anchor Point Map Colored with Estimated text-davinci-003 (Prompt 1) predictions. The estimates achieve an MAE of 0.16 and agreement of 76%.

(c) QQP Anchor Point Map Colored with True deberta-v3-base predictions.

(d) QQP Anchor Point Map Colored with True text-davinci-003 (Prompt 1) predictions.

Figure 10: Anchor Point Map for 1000 QQP examples. The map is computed using the predictions of 60 randomly-selected source models. We then estimate the predictions of the two held-out target models, deberta-v3-base and text-davinci-003, by evaluating each on 30 anchor points. We color the remaining 970 test points in 10(a) and 10(b) with these estimates. Finally, we color maps 10(c) and 10(d) with the true target model predictions. We observe that the estimated predictions achieve low MAE and high agreement with the true predictions.

(a) QQP Anchor Point Map Colored with Estimated text-curie-001 (Prompt 1) predictions. The estimates achieve a large MAE of 0.30 and poor agreement of 54%.

(b) QQP Anchor Point Map Colored with True text-curie-001 (Prompt 1) predictions.

Figure 11: Anchor Point Failure Case: The map is computed using the predictions of 60 randomly-selected source models on QQP and used to estimate the predictions of text-curie-001 with Prompt 1 (11(a)). The estimates achieve a large MAE and near random agreement. Upon inspecting the true predictions (11(b)), we observe that the behavior of text-curie-001 is not localized in the Anchor Point Map. Model performance appears sporadic. This suggests that text-curie-001 does not follow the same predictive correlations as the source models, preventing effective estimation of the model's predictions.

	Mean	Random	Random Weighted	Pretrained	Pretrained Weighted	Fine-Tuned	Fine-Tuned Weighted	AP Weighted	AP Predictor
Datasets	Family	Exact	Corr	Exact	Corr	Exact	Corr	Corr	Exact
SST-2	BERT	0.485	0.607	0.512	0.637	0.594	**0.738**	0.696	0.593
QQP	BERT	0.459	0.631	0.670	0.613	0.572	**0.756**	0.691	0.555
RTE	BERT	0.478	0.406	0.316	-0.001	0.486	0.335	0.591	**0.597**
QNLI	BERT	0.346	0.523	-0.252	0.613	-0.430	0.510	**0.665**	0.571
MRPC	BERT	0.730	0.741	0.737	0.749	0.692	**0.786**	0.778	0.721
MNLI	BERT	0.684	0.748	0.646	0.653	0.676	**0.735**	0.733	0.720
BERT Avg.		0.530	0.609	0.438	0.543	0.432	0.643	**0.699**	0.626
SST-2	GPT	0.440	0.278	0.548	0.215	**0.651**	0.531	0.441	0.632
QQP	GPT	0.561	0.591	-0.557	-0.537	0.720	0.752	0.777	**0.825**
RTE	GPT	0.188	0.138	-0.119	-0.251	0.133	0.082	0.378	**0.452**
QNLI	GPT	0.101	0.076	0.209	0.197	0.185	0.130	**0.158**	0.126
MRPC	GPT	0.475	0.460	0.155	0.321	0.360	0.360	0.516	**0.681**
MNLI	GPT	0.033	0.059	-0.517	-0.382	0.181	0.099	0.201	**0.494**
GPT Avg.	GPT	0.300	0.267	-0.470	-0.730	0.371	0.325	0.410	**0.535**
SST-2	IGPT	0.655	0.722	0.726	0.728	0.606	0.760	**0.814**	0.719
QQP	IGPT	0.734	0.801	0.264	0.276	0.824	**0.893**	0.884	0.845
RTE	IGPT	0.389	0.322	0.220	0.03	0.391	0.323	**0.671**	0.629
QNLI	IGPT	0.404	0.400	0.512	0.469	0.449	0.296	0.597	**0.690**
MRPC	IGPT	0.727	0.680	0.520	0.651	0.651	0.712	**0.776**	0.738
MNLI	IGPT	0.265	0.305	-0.074	-0.088	**0.460**	0.420	0.434	0.317
IGPT Avg.		0.529	0.538	0.361	0.344	0.563	0.566	**0.696**	0.656
SST-2	OAI	0.666	0.697	0.774	**0.871**	0.761	0.833	0.734	0.661
QQP	OAI	0.610	0.651	-0.090	-0.080	0.676	0.722	**0.747**	0.637
RTE	OAI	0.362	0.303	0.206	0.066	0.381	0.294	-0.378	**0.411**
QNLI	OAI	0.451	0.453	0.381	0.378	0.286	-0.03	**0.497**	0.309
MRPC	OAI	0.633	0.661	0.533	0.563	0.550	0.673	**0.714**	0.705
MNLI	OAI	0.300	0.278	-0.015	-0.181	0.559	**0.490**	0.350	0.486
OAI Avg.	OAI	0.503	0.507	0.298	0.270	0.536	0.497	**0.570**	0.535
SST-2	ALL	0.685	0.705	0.734	0.725	0.730	**0.787**	0.757	0.727
QQP	ALL	0.669	0.678	0.189	0.233	0.766	**0.770**	0.756	0.701
RTE	ALL	0.366	0.308	0.143	-0.052	0.354	0.275	**0.483**	0.462
QNLI	ALL	0.321	0.331	0.192	0.294	0.127	0.144	**0.439**	0.303
MRPC	ALL	0.687	0.679	0.528	0.604	0.641	0.681	**0.726**	0.716
MNLI	ALL	0.438	0.433	0.177	0.166	0.523	0.453	**0.544**	0.517
General Avg.	ALL	0.528	0.522	0.327	0.328	0.523	0.518	**0.612**	0.571

Table 4: Area Under the (Kendall's τ) Correlation Curve from 1 to 30 points for ranking language models at 6 GLUE tasks from each model family. We randomly select 5 source models each for the InstructGPT and OpenAI Families and 10 each for the BERT, GPT, and ALL families. We then rank the remaining models within each family and average over 100 randomized runs. AP Weighted and AP Predictor prove to be the most effective at accurately ranking models in this small-data regime. In each row, the best score is **bolded** and second best score is underlined.

(a) QNLI

(b) MRPC

(c) MNLI

Figure 12: Kendall Tau Rank Correlation between Model Rankings on Small Evaluation Sets and Full Validation Sets for QNLI, MRPC, and MNLI. We rank 77 language models belonging to all model families using various evaluation selection techniques. Anchor Points are fit to 10 source model predictions. Each point is the mean of 100 runs with randomized source and target models. Shading indicated standard error. Anchor Points Weighted achieves the most reliable performance overall at low evaluation set sizes.

(a) QQP

(b) RTE

(c) SST-2

Figure 13: Kendall Tau Rank Correlation between Model Rankings on Small Evaluation Sets and Full Validation Sets for QQP, RTE, and SST-2. We rank 77 language models belonging to all model families using various evaluation selection techniques. Anchor Points are fit to 10 source model predictions. Each point is the mean of 100 runs with randomized source and target models. Shading indicated standard error. Anchor Points Weighted achieves the most reliable performance overall at low evaluation set sizes.

(a) MNLI (b) MRPC

(c) QNLI (d) QQP

(e) RTE (f) SST-2

Figure 14: Correlation Matrices of Correct Class Confidence Predictions from 87 models across all model families. Each plot shows the correlations between 50 examples sampled from the validation Sets of GLUE Tasks.