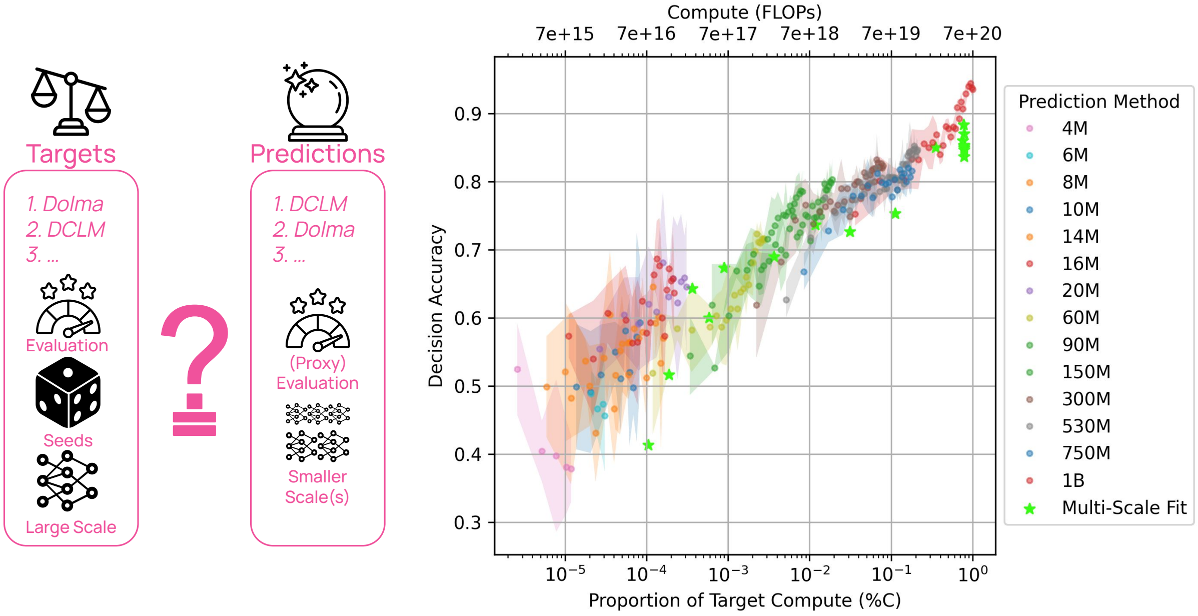

Figure 1: Which pretraining data to use? Ideally, large experiments with fixed models over random seeds would be used to compare options (left). In practice, cheaper, smaller scale experiments are used (center). DataDos is a method to validate lower-cost predictions to make pretraining data decisions (right). This plot shows the relationship between compute used to predict a ranking of datasets and how accurately that ranking reflects performance (quantified here by OLMES) at the target (1B) scale of models pretrained from scratch on those datasets. Each point represents the average decision accuracy of 3 prediction attempts over different random seeds using a given method and model size up to some compute budget and shading shows standard deviation. More compute makes better decisions. Decisions from intermediate checkpoints are as good as compute equivalent final checkpoints.

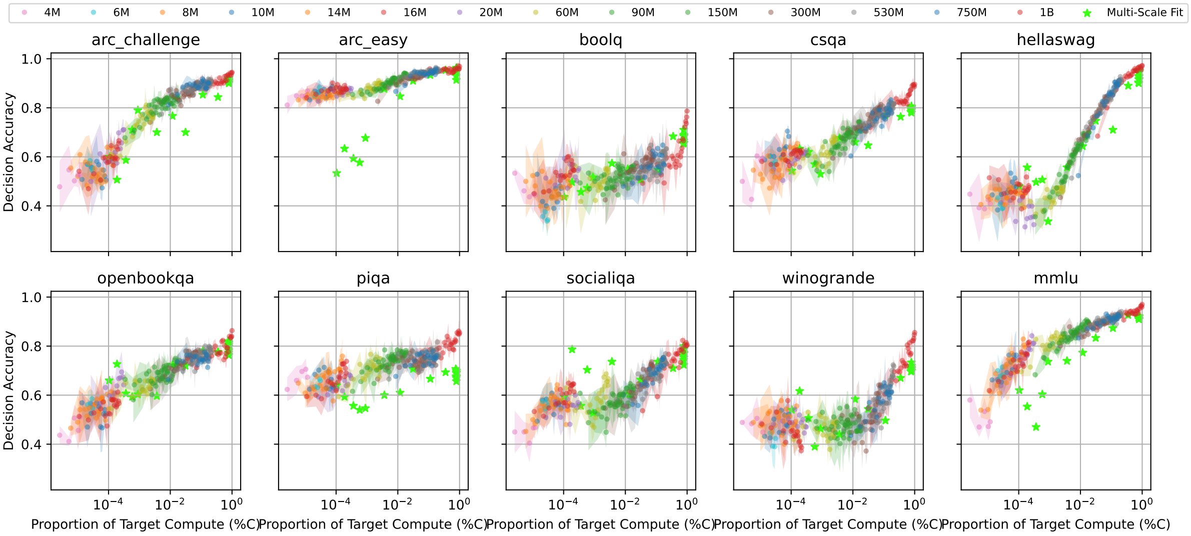

Figure 2: Accuracy in pairwise decisions on best data when evaluating on the 10 OLMES tasks with primary_metric (shown aggregated in Figure 1). The amount of compute needed to make good predictions varies between tasks. ARC and MMLU make good predictions with less than 2 orders of magnitude of target compute. HellaSwag predicts well with more compute. The rest of OLMES tasks give markedly less reliable predictions across the scales we examine.

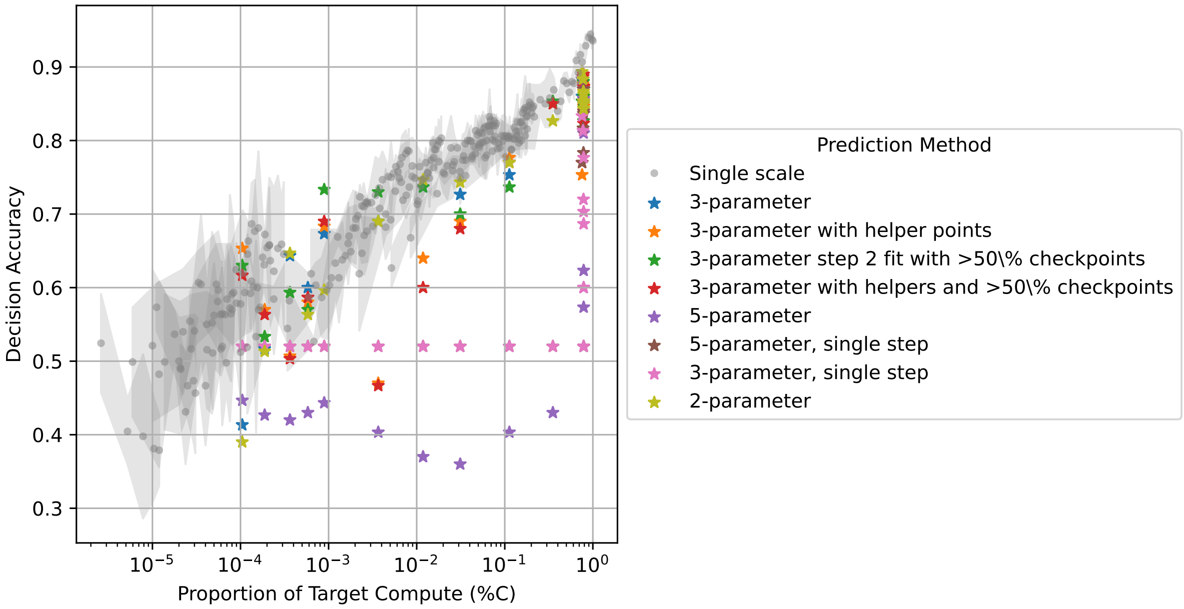

Figure 3: Decision accuracy over 8 baseline scaling law variants. At best, these approaches reach only the same compute to decision accuracy frontier as ranking single scale experiments. DataDos can be used to iterate on future scaling law prediction methods.

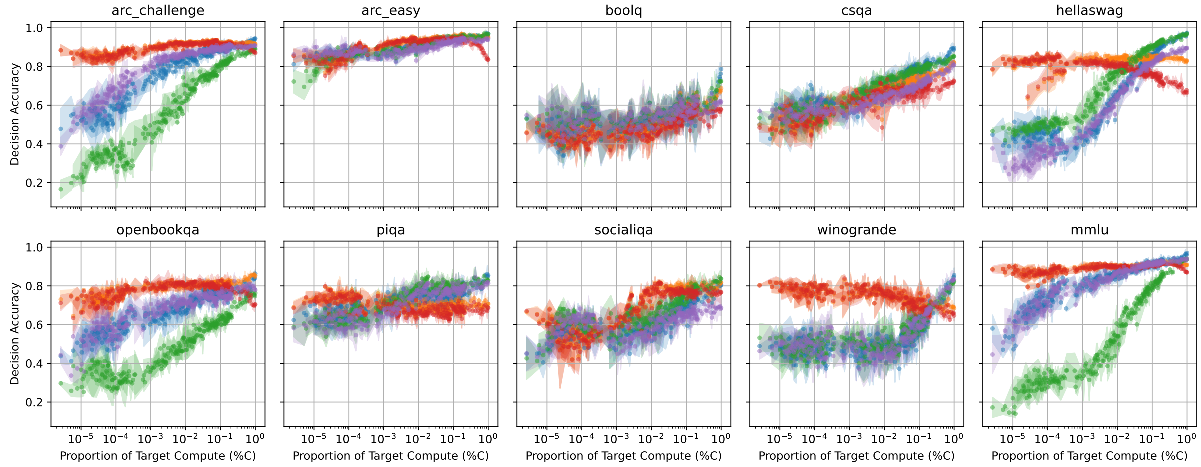

Figure 4: Per-task decision accuracy using character normalized proxy metrics for primary_metric targets. 5 tasks benefit at smaller scales from using raw likelihood of answers (correct_prob and total_prob), as opposed to discrete primary_metric or continuous metrics that penalize probability on incorrect answers (norm_correct_prob, margin).

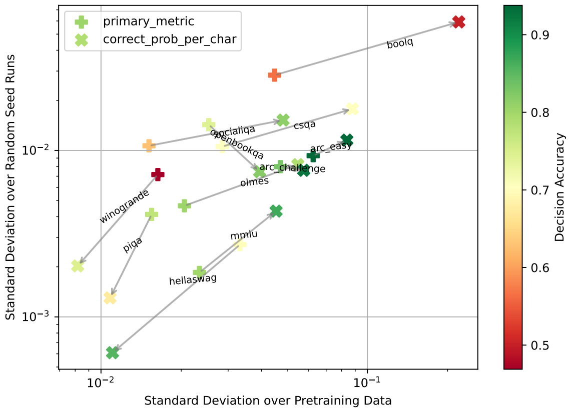

Figure 5: Why do some tasks or metrics get better or worse decision accuracy? High decision accuracy at 150M when using correct_prob for tasks like HellaSwag comes with low run-to-run variance and for tasks like arc_easy come with widely spread the performance assigned to different pretraining data.

	Discrete		Continuous	
size	4M	60M	4M	60M
task				
gsm8k	0.39	0.53	0.45	0.52
minerva	0.36	0.56	0.52	0.53
mbpp	0.41	0.56	**0.84**	**0.86**
humaneval	0.38	0.52	**0.81**	**0.75**

Figure 6: Code tasks such as humaneval and MBPP go from trivial decision accuracy to largely predictable when using using continuous correct_prob_per_char instead of discrete primary_metric. Meanwhile common math tasks remain near trivial decision accuracy regardless of metric. Bolded values more than 10% above random