RESIDUAL-AUGMENTED SCALING LAWS FOR LLM PRETRAINING PERFORMANCE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The computational cost of pretraining Large Language Models (LLMs) at modern scales makes it essential to predict pretraining outcomes (validation loss on hold-out data) before committing to expensive full-scale pretraining. The scaling laws of parameters and token sizes enable extrapolation from small-scale pilots but miss other pretraining feature-dependent variations (i.e., hyperparameters and data states), while purely data-driven methods require substantial training data and extrapolate unreliably beyond observed regimes. To address this dual limitation, we propose `ReScaL` (Residual-augmented Scaling Laws), a hybrid method that diversely fits an ensemble of scaling laws on minimal pilot runs; learns ML-predicted residual corrections to capture pretraining feature-dependent variations that scaling laws miss; and dynamically combines scaling law extrapolation with residual corrections to leverage the strengths of both. Through experiments on 52 pairs of pretraining datasets and budget sizes, we show that `ReScaL` is ranked 1st in 41/52 cases (79%) with the best average rank of 1.21, and attains up to 5.7× lower MAPE than the strongest counterpart. Ablations show that removing each design can degrade accuracy by up to 4.3×, while sensitivity analysis confirms our hyperparameter choices as robust operating points.

1 INTRODUCTION

Pretraining Large Language Models (LLMs) at modern scales is computationally expensive, making it impractical to perform full-scale experiments for every hyperparameter setting/pretraining data state. As a result, critical decisions about scaling must often be made **prior** to full-scale pretraining. Empirical studies have shown that pretraining performance (the loss on a hold-out validation set at a fixed token budget during pretraining) often exhibits smooth, predictable trends across varying model sizes, dataset sizes, and compute budgets (Baker et al., 2018; Hestness et al., 2017). These trends, captured in scaling laws, can be inferred from small-scale low-cost pilot runs and used to forecast outcomes of larger training runs and avoid unnecessary full-scale trials (Ghorbani et al., 2022; Muennighoff et al., 2023; Chen et al., 2025b). However, existing methods encounter substantial limitations that may restrict their effectiveness.

Classical scaling laws model LLM validation loss as a power-law function of scale variables, typically including model size (number of parameters N) and pretraining data volume (number of tokens D) (Kaplan et al., 2020; Gadre et al., 2025; Sengupta et al., 2025). However, pretraining outcomes may also be sensitive to other pretraining features, i.e., hyperparameters (e.g., batch size) and relevant pretraining data states (e.g., ratios of domains in the data), not captured by scale variables alone (Subramanian et al., 2024). Moreover, different LLM families and architectures may follow distinct trends, and even within the same architecture, scaling behavior can shift with data characteristics and hyperparameters, requiring careful calibration for new settings (Li et al., 2025c; Tay et al., 2023; Chen et al., 2025c). Alternatively, purely data-driven methods, such as Random Forests (`RF`) or Neural Networks (`NN`), can incorporate a wider set of pretraining features of LLMs (Domhan et al., 2015; Klein et al., 2017). While expressive, these models often demand large datasets for training and tend to extrapolate poorly outside the regimes they have observed.

A central challenge thus emerges: scaling laws excel at extrapolating to larger, expanded scales when scaling trends are well-established, but miss other pretraining feature-dependent variations; conversely, data-driven predictors can capture rich feature effects when sufficient data is available in

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the same LLM scale, but extrapolate unreliably beyond observed regimes, especially when pretraining data is limited. This scenario-dependent trade-off motivates combining both approaches, as illustrated in Figure 1.

To overcome these constraints, we propose ReScaL (Residual-augmented Scaling Laws) for LLM pretraining performance prediction, addressing both the feature-representation gaps of scaling laws and the extrapolation limitations of machine learning models. In a nutshell, there are contributions within ReScaL:

- We propose and show that the diverse fitting and parametric softmax ensemble of scaling laws improves robustness under limited pilot data by stabilizing adaptation and reducing single-law mis-specification.

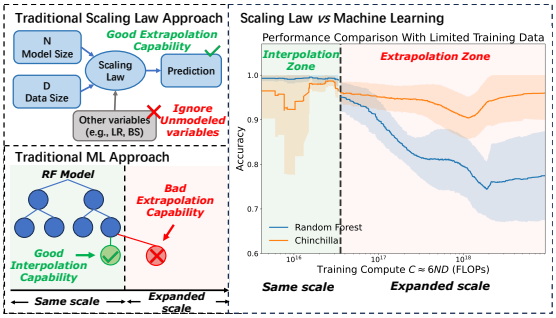


Figure 1: Limitations of scaling law Chinchilla and machine learning predictor RF.

- We leverage residual learning to correct systematic deviations that the scaling laws miss in *prior* due to pretraining features beyond (N, D); ablations show it yields the largest accuracy degradations when removed (e.g., $\approx$2.7$\times$ on DENSE4 dataset).
- We design a pilot-set-level, regime-dependent weight α that blends residual corrections with the scaling-law ensemble without per-dataset validation tuning; removing it gives the worst average rank (2.25), evidencing its importance.

Overall, across 52 dataset-budget cases, ReScaL achieves rank 1 in 41/52 cases (79%), the best average rank (1.23), and up to 5.7$\times$ lower MAPE than the strongest counterpart in an individual case. ReScaL also obtains the best mean ranks on configuration-ranking and selection metrics. All data and code can be accessed at our anonymous repository: https://anonymous.4open.science/r/ReScaL-8D70.

2 Preliminaries and Related Work

2.1 Problem Formulation

We study **pretraining validation-loss prediction** for a limited pilot-compute budgets, where the goal is to forecast the validation loss of expensive large-scale pretraining runs from a small number of low-compute pilot runs. Each LLM pretraining run i under a scale setting[1] yields $(N_i, D_i, \mathbf{x}_i, y_i)$: model size N_i, training tokens D_i, a feature vector $\mathbf{x}_i \in \mathbb{R}^d$ describing other pretraining features of LLM hyperparameters (e.g., batch size) and/or data states (e.g., data mixing ratios), and the target validation loss y_i computed on a held-out validation corpus.

Concretely, for an autoregressive LLM with parameters $\boldsymbol{\theta}$ and a validation token sequence $w_{1:T}$, its loss function is the (token-average) negative log-likelihood (cross-entropy)

$$y \;=\; \ell_{\text{val}}(\boldsymbol{\theta}) \;=\; -\frac{1}{T}\sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(w_t \mid w_{<t}) \tag{1}$$

which is also the validation loss often modeled in prior scaling law studies (Kaplan et al., 2020; Hoffmann et al., 2022).

Given all samples in a pilot set $\mathcal{S}_{\text{pilot}} = \{(N_i, D_i, \mathbf{x}_i, y_i)\}_{i=1}^{n}$ (typically small n), which might involve multiple scale settings and their runs, together with a target scale setting $(N^\star, D^\star)$ of much larger scales ($N^\star \gg \forall N_i; D^\star \gg \forall D_i$) and its extra pretraining feature values $\mathbf{x}^\star$, the task is to learn a predictor f that outputs $\hat{y}^\star = f(N^\star, D^\star, \mathbf{x}^\star)$. The prediction can be evaluated by any error metric as in curve extrapolation (Chen et al., 2025a).

[1]Note that a scale setting (N, D) might have different pretraining runs due to the different hyperparameters values and/or data states.

2.2 Related Work

Scaling laws for validation loss. It has been shown that cross-entropy loss follows predictable scaling trends over wide ranges and that fitted laws can guide compute-allocation decisions (Kaplan et al., 2020). New scaling laws have been proposed to extend that analysis across autoregressive generative settings (Henighan et al., 2020); and Hoffmann et al. (2022) refine compute-optimal scaling by training hundreds of models and validating improved token-to-parameter trade-offs. Recently, Bahri et al. (2024) investigate theoretical explanations for why neural scaling laws emerge. There exist surveys on common scaling law forms and fitting practices (Sengupta et al., 2025; Li et al., 2025c), while practical estimation issues reveal that using intermediate checkpoints and carefully matched reference models can improve prediction reliability has also been studied (Choshen et al., 2025). Despite these advances, scaling laws suffer from three fundamental weaknesses: (1) they primarily model a small set of fixed scaling variables (e.g., N and D), while holding the remaining training configurations fixed or assuming them to be optimally configured (Subramanian et al., 2024), creating a feature gap; (2) changes in data characteristics and hyperparameter values can alter scaling trends, requiring recalibration for new scenarios (Li et al., 2025c; Subramanyam et al., 2025); and (3) they still require a sufficient number of well-chosen training points to obtain reliable estimates, leading to non-trivial empirical cost in practice.

Performance modeling and learning-curve prediction. Learning-curve extrapolation is a long-standing strategy to reduce training cost by forecasting outcomes from early progress signals, e.g., current work studies extrapolation and uncertainty-aware prediction (Domhan et al., 2015; Klein et al., 2017). In software/system performance modeling, machine learning surrogates (e.g., `RF`, `NN`) can predict validation loss from richer feature vectors (Gong & Chen, 2025). Nevertheless, purely black-box predictors exhibit two critical shortcomings: (1) they often require substantial pretraining data to generalize (Domhan et al., 2015; Klein et al., 2017), undermining cost savings when it requires many pretraining runs under diverse scale settings to learn; and (2) they extrapolate unreliably beyond observed regimes (White et al., 2021), especially if the pretraining data is limited.

Hybrid models. Hybrid predictors that combine an analytical forms with a learned models exist in performance engineering: Didona et al. (2015) explicitly propose combining analytical modeling with machine learning to improve robustness, and Ataie et al. (2022) show that fusing analytical and machine learning predictors can reduce the number of measurements needed to reach useful accuracy. Zhang et al. (2026) propose NCPL, which maps full training configurations to performance and learns configuration-dependent residuals relative to a Chinchilla baseline. However, the existing hybrids fundamentally differ from `ReScaL`: (1) they often rely on a **single** chosen functional form, making them sensitive to mis-specification when scaling behavior deviates from that form (Choshen et al., 2025) (`ReScaL` uses **ensemble**); (2) they typically use **static** combination strategies between the analytical and learned parts, which cannot adapt across scale regimes or feature settings (Ataie et al., 2022) (`ReScaL` is **dynamic**); (3) they often do not focus on direct "correction" relationship as in `ReScaL` but linearly combined the independent predictions of each together; and (4) they incorporate domain-specific analytical theory, e.g., queuing theory for systems, which is incompatible for LLM.

3 The ReScaL Method

To mitigate these limitations, we introduce `ReScaL`, a three-stage framework illustrated in Figure 2 and detailed below.

3.1 Scaling Law Ensemble

We consider $K = 9$ candidate laws where the kth law $f_k(N, D; \boldsymbol{\theta}_k)$ maps scale variables (N, D) to LLM pretraining validation loss; each law differs on the structure and/or the focused parameters in fitting. We instantiate f_k using the formulations of three structurally different fundamental laws to reduce sensitivity to mis-specification when scaling behavior deviates from any single form:

- **`Kaplan`** (coupled N–D) (Kaplan et al., 2020):

$$f_{\text{Kap}}(N, D) = \left[\left(\frac{N_c}{N} \right)^{\alpha_N/\alpha_D} + \left(\frac{D_c}{D} \right) \right]^{\alpha_D} \tag{2}$$

- **`Chinchilla`** (additive surface; E is a constant) (Hoffmann et al., 2022):

$$f_{\text{Chin}}(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} \tag{3}$$

- **`Farseer`** (interaction via N-dependent data exponent) (Li et al., 2025b):

$$f_{\text{Far}}(N, D) = \exp(a_3 N^{\gamma} + b_3) + \exp(a_2 N^{\beta} + b_2) \cdot D^{-\exp(a_1 N^{\alpha} + b_1)} \tag{4}$$

3.1.1 Diverse Fittings on Pilot Runs

Scaling laws come with some default, pre-fitted parameter values. Yet, their scaling behaviors can shift with other pretraining features, requiring re-fitting these laws to new settings. Leveraging solely on full parameters re-fitting from a small pilot set can be unstable; recent analysis recommends leveraging prior estimates and using constrained estimation when data are limited (Choshen et al., 2025; Li et al., 2025c). We therefore adopt **diverse fitting**: a variant law, instantiated from the three templates, can be either fully or partially fitted. Those variant laws that follow partial fitting can *warm-start* from previously fitted parameters $\boldsymbol{\theta}_k^0$ and update only a subset $\boldsymbol{\theta}_{k,\text{adapt}}$, keeping the rest $\boldsymbol{\theta}_{k,\text{fixed}}^0$ fixed:

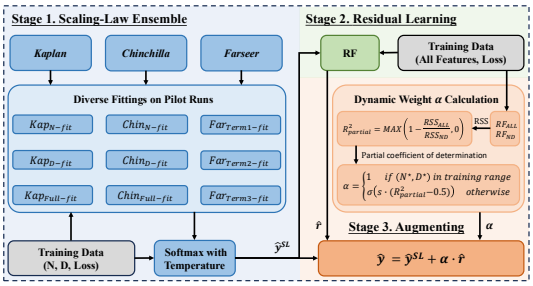


Figure 2: Overview of `ReScaL` framework.

$$\boldsymbol{\theta}_{k,\text{adapt}}^{\star} = \arg\min_{\boldsymbol{\theta}_{k,\text{adapt}}} \sum_{(N_i, D_i, y_i) \in \mathcal{S}_{\text{pilot}}} \ell\left(y_i, \hat{y}_i^{(k)}\right), \hat{y}_i^{(k)} := f_k\left(N_i, D_i; \boldsymbol{\theta}_{k,\text{adapt}}, \boldsymbol{\theta}_{k,\text{fixed}}^0\right). \tag{5}$$

The subset $\boldsymbol{\theta}_{k,\text{adapt}}$ is selected based on the scaling law structure: for `Kaplan` and `Chinchilla`, we create variants of laws that update either N-related parameters (e.g., N_c, α_N for `Kaplan`; A, α for `Chinchilla`) or D-related parameters (e.g., D_c, α_D for `Kaplan`; B, β for `Chinchilla`), allowing adaptation to different scale regimes. For `Farseer`, we update parameter groups corresponding to each exponential term: (a_3, γ, b_3), (a_2, β, b_2), or (a_1, α, b_1), leading to three variant laws since each captures distinct scaling behaviors. Here, the fixed parameters are constrained to vary within $\pm 1\%$ of their past values, while adapted parameters can vary within $\pm 35\%$ using L-BFGS-B optimization (Byrd et al., 1995). When multiple pretraining runs share the same (N, D) due to differing $\mathbf{x}$, we only use the one with the best LLM validation loss to approximate the optimal case (Hoffmann et al., 2022). We also include full fittings for `Kaplan` and `Chinchilla`, but not for `Farseer`, whose more complex structure and larger parameter set make reliable fitting difficult on limited pilot data. In total, `ReScaL` retains $K = 9$ laws (variants) in the ensemble. More details are in Appendices A.1–A.2.

3.1.2 Softmax Scaling Law Ensemble with Temperature

After the fitting, the kth law yields a prediction $\hat{y}^{(k)}$ for a target scale setting and its extra pretraining feature values. Since no single law is consistently the best, we combine them using error-weighted averaging for a final prediction $\hat{y}^{\text{SL}}$:

$$\hat{y}^{\text{SL}} = \sum_{k=1}^{K} w_k \, \hat{y}^{(k)}, \qquad w_k = \frac{\exp(s_k/t)}{\sum_{j=1}^{K} \exp(s_j/t)}, \qquad \hat{y}^{(k)} := f_k(N, D; \boldsymbol{\theta}_k^{\star}) \tag{6}$$

where $\boldsymbol{\theta}_k^{\star}$ denotes all parameters (fixed or adapted) after fitting for the kth law. Here, each fitted law is evaluated using 5-fold cross-validation on the pilot data, and its error[2], denoted as e, is converted

[2]In this work, we primarily use MAPE for accuracy/error; see Section 4.2.

into a scalar score, i.e., $s_k = -e_k$ (larger s_k indicates better fit) for the kth law. These scores are then mapped to a function of **parametric softmax** w_k for each law, controlled by the temperature t: smaller t emphasizes the top-performing laws, while bigger t yields smoother, closer-to-uniform weighting (Guo et al., 2017). This is common for combining mixture-style models (Shazeer et al., 2017).

3.2 Residual Learning

Scaling laws provide a strong extrapolative prior in (N, D), but systematic deviations can arise from other LLM pretraining features not represented by scale variables. To solve this, we aim to combine a structured prior with data-driven flexibility. Thereby, a common solution is to learn a **residual** (discrepancy) model that corrects the prior rather than replacing it (Kennedy & O'Hagan, 2001). For training data where we observe the true LLM validation loss y, we can compute the actual residual:

$$r := y - \hat{y}^{\text{SL}} \tag{7}$$

where $\hat{y}^{\text{SL}}$ is predicted by the scaling laws ensemble for a scale setting; y is the corresponding ground truth. We learn a regressor $g(\cdot)$ that maps a feature vector $\mathbf{z}_i$ to the residual r_i for a pretraining run i:

$$r_i = g(\mathbf{z}_i), \qquad \mathbf{z}_i := \left[N_i,\ D_i,\ \mathbf{x}_i\right] \tag{8}$$

$\mathbf{z}_i$ includes the extra pretraining features $\mathbf{x}_i$ together with the scale variables (N_i, D_i) for i, enabling the regressor to learn interactions within full pretraining features. The regressor g is learned by minimizing the regression loss on the pilot set:

$$\mathcal{L} := \sum_{i \in \mathcal{S}_{\text{pilot}}} \ell(r_i,\ g(\mathbf{z}_i)) \tag{9}$$

Model Choice. `ReScaL` is agnostic to the choice of regressor g. In this work, we use `RF` by default because it is a simple, yet strong non-parametric regressor with good out-of-the-box robustness in small-data settings and can capture non-linear feature interactions (Zhou & Feng, 2017).

3.3 Augmenting Scaling Laws with Residual Learning via Dynamic Weight α

A static combination of scaling-law predictions and residual corrections is inherently suboptimal, since the residual learner can only model systematic errors supported by the pilot data; its contribution degrades under noisy features or when extrapolating far beyond the observed scale regime. We therefore use a bounded, regime-dependent correction weight that controls how much the residual model is trusted. For a target scale setting $(N^\star, D^\star)$ and its corresponding extra pretraining feature values $\mathbf{x}^\star$, the final prediction is:

$$\hat{y}^\star := f(N^\star, D^\star, \mathbf{x}^\star) := \hat{y}^{\text{SL}} + \alpha \cdot \hat{r}, \qquad \alpha \in [0, 1] \tag{10}$$

where $\hat{y}^{\text{SL}}$ is the prediction of scaling laws ensemble using $(N^\star, D^\star)$ and $\hat{r}$ is the predicted residual using the learned regressor g under $(N^\star, D^\star)$ and $\mathbf{x}^\star$.

Intuitively, the weight α should be **small** when the scaling law ensemble is reliable, and **large** when systematic feature-driven deviations remain, and residual correction is expected to help. We operationalize this by training two `RF` models on the pilot set: (1) $\texttt{RF}_{\texttt{ND}}$ using only (N, D) features, (2) $\texttt{RF}_{\texttt{ALL}}$ using all features including $\mathbf{x}$; and compute the partial coefficient of determination (R^2_{partial}) (Lipsitz et al., 2001) to quantify how much other pretraining features beyond (N, D) improve the residual model:

$$R^2_{\text{partial}} = \max\left(1 - \frac{\text{RSS}_{\text{All}}}{\text{RSS}_{\text{ND}}}, 0\right) \tag{11}$$

where RSS_{All} and RSS_{ND} are the residual sums of squares (RSS) of $\texttt{RF}_{\texttt{ALL}}$ and $\texttt{RF}_{\texttt{ND}}$, respectively, using all samples in the pilot set. Thus, each pilot set produces one global extrapolation coefficient $\alpha_{\text{global}} = \sigma(s(R^2_{\text{partial}} - 0.5))$. The weight used for a target scale setting is:

$$\alpha = \begin{cases} 1 & \text{if } (N^\star, D^\star) \text{ in training range} \\ \sigma(s \cdot (R^2_{\text{partial}} - 0.5)) & \text{otherwise} \end{cases} \tag{12}$$

When the target scale setting $(N^\star, D^\star)$ falls within the training-data coverage (i.e., $N^\star D^\star$ lies within the observed compute range), we are in an interpolation regime and set $\alpha = 1$. Otherwise, every extrapolation target associated with the same pilot set uses the same α_{global}. Hence, the gate is regime-dependent rather than continuously sample-specific. Larger R^2_{partial} values yield larger residual weights; σ bounds the coefficient and s controls the transition sharpness. The midpoint 0.5 conservatively amplifies the correction only when extra features explain a substantial portion of the residual variation. More discussion and direct fixed-α comparisons are in Appendix C.1.

4 EXPERIMENTS

We evaluate `ReScaL` through three research questions: **RQ1:** How accurately does `ReScaL` predict and rank candidate configurations under different pilot budgets? **RQ2:** How much does each component contribute (i.e., diverse fitting, ensembling, residual learning, and dynamic α)? **RQ3:** How sensitive is `ReScaL` to t (ensemble temperature), s (gate sharpness), and the residual weight?

4.1 DATASETS

We evaluate `ReScaL` on eight LLM pretraining datasets from prior studies. Table 1 summarizes the dataset sizes, scale ranges, and extra pretraining feature dimensions. Each dataset contains multiple scale settings for the same LLM architecture. Each scale setting can contain one or more pretraining run i that provides $(N_i, D_i, \mathbf{x}_i, y_i)$, where the scale variables (N_i, D_i) is a unique combination and $\mathbf{x}_i$ contains other available LLM pretraining features from the source study; y_i is the validation loss. Detailed pretraining feature definitions and dataset information are given in Appendix B.1.

Table 1: Datasets studied. $|\boldsymbol{\mathcal{S}}|$: # scale settings (distinct scale variables (N, D)); $|\boldsymbol{\mathcal{X}}|$: feature dimension excluding N and D.

Dataset	$\lvert\boldsymbol{\mathcal{S}}\rvert$	N value/range	D value/range	$\lvert\boldsymbol{\mathcal{X}}\rvert$	Reference
DENSE1	407	2.0×10^8–6.4×10^9	1.0×10^9–5.1×10^{11}	10	Li et al. (2025a)
DENSE2	17	2.1×10^8–1.1×10^9	4.0×10^9–1.0×10^{11}	7	Li et al. (2025b)
DENSE3	137	7.1×10^6–8.7×10^9	1.0×10^8–9.0×10^{11}	1	Muennighoff et al. (2023)
DENSE4	3	7.0×10^7–4.1×10^8	3.0×10^{10}	7	Ye et al. (2025)
DENSE5	5	7.0×10^7–1.0×10^9	4.1×10^{10}	11	Ye et al. (2025)
DENSE6	907	5.7×10^7–1.2×10^9	1.0×10^8–2.1×10^{10}	12	Besiroglu et al. (2024)
MOE1	12	2.2×10^9	2.0×10^9–2.0×10^{10}	15	Li et al. (2025a)
MOE2	131	1.7×10^7–2.1×10^{11}	1.3×10^{11}	16	Clark et al. (2022)

4.2 EXPERIMENTAL SETUP

Evaluation Protocol. We evaluate each dataset independently, adopting a non-random train/test split that follows the standard practice of using a small number of low-cost, small-scale pilot runs to forecast validation loss for high-cost, large-scale trials. We predefine a training-cost proxy using the FLOP estimate $C \approx 6ND$ (Kaplan et al., 2020).

For a dataset with $|\mathcal{S}|$ scale settings (distinct (N, D) groups), we sample $\lfloor \rho \cdot |\mathcal{S}| \rfloor$ **cheapest** scale settings on C and use all of their pretraining runs as training (pilot) dataset $\mathcal{S}_{\text{pilot}}$. We vary $\rho \in (0, 1)$ to simulate different pilot budget sizes; datasets with more settings use finer grids (e.g., 0.1 steps), while smaller datasets use coarser grids (e.g., $\{0.1, 0.3, 0.5, 0.7, 0.9\}$). In this way, the number of samples of LLM pretraining runs for training `ReScaL` ranges from 14 to $30,844$. The above pilot budgets/training scale settings, together with eight datasets, lead to 52 cases of comparison.

For testing data, we select the pretraining runs from the **most expensive** scale settings on the **Pareto front** of all scale settings, i.e., the non-dominated points where no other setting has both larger N and larger D. This allows us to focus evaluation on extrapolating from various small-scale pilots to some unseen large-scale targets (kindly refer to Appendix B.2 for more details).

Method Hyperparameters. For `ReScaL`, we use $K = 9$ scaling laws instantiated from `Kaplan`, `Chinchilla`, and `Farseer`; the `RF` for residual learning is set with default hyperparameters. Unless otherwise stated (e.g., see **RQ3**), the ensemble temperature is fixed to $t = 0.1$, and the sigmoid scaling parameter is $s = 20$. All other counterparts use their default hyperparameters.

Metric and Statistical Test. We focus on MAPE (Mean Absolute Percentage Error) as the main **accuracy metric** for each test scale setting (MAE/RMSE results can be found in Appendix), defined as $\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$, where n is the number of pretraining runs under a test scale setting, y_i is the actual LLM validation loss, and $\hat{y}_i$ is the predicted LLM validation loss. For all test scale

Table 2: Full RQ1 comparison across all 52 dataset–budget cases; HUMAN is reported where available. For each case, blue cells mark the best mean MAPE when achieved by ReScaL, while red cells mark the best mean MAPE when achieved by another method. Methods in the best Scott–Knott ESD group ($r = 1$) are highlighted in **bold**.

Dataset	ReScaL		Farseer		$\text{Farseer}_{\text{d}}$		Chinchilla		$\text{Chinchilla}_{\text{d}}$		Kaplan		Kaplan_{d}		RF		RF_{ND}		NCPL		HUMAN	
	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)
DENSE1																						
$\rho = 0.1$	**1**	**0.0203 (0.02)**	5	0.1554 (0.14)	10	0.7612 (0.00)	3	0.1234 (0.04)	6	0.1769 (0.03)	2	0.0893 (0.03)	4	0.1336 (0.04)	9	0.4449 (0.08)	8	0.4119 (0.07)	7	0.2323 (0.04)		N/A
$\rho = 0.2$	**1**	**0.0135 (0.00)**	2	0.0764 (0.10)	8	0.7612 (0.00)	3	0.1108 (0.04)	5	0.1769 (0.03)	2	0.0810 (0.03)	4	0.1336 (0.04)	7	0.3673 (0.07)	7	0.3616 (0.07)	6	0.2206 (0.04)		N/A
$\rho = 0.3$	**1**	**0.0152 (0.01)**	2	0.0645 (0.10)	8	0.7612 (0.00)	3	0.0950 (0.03)	5	0.1769 (0.03)	2	0.0738 (0.03)	4	0.1336 (0.04)	7	0.3237 (0.07)	7	0.3122 (0.07)	6	0.2048 (0.04)		N/A
$\rho = 0.4$	**1**	**0.0162 (0.01)**	2	0.0622 (0.09)	9	0.7612 (0.00)	3	0.0889 (0.03)	5	0.1769 (0.03)	2	0.0652 (0.03)	4	0.1336 (0.04)	8	0.2770 (0.07)	7	0.2491 (0.07)	6	0.1980 (0.04)		N/A
$\rho = 0.5$	**1**	**0.0193 (0.02)**	2	0.0591 (0.09)	9	0.7612 (0.00)	3	0.0832 (0.03)	5	0.1769 (0.03)	2	0.0600 (0.03)	4	0.1336 (0.04)	8	0.2471 (0.07)	7	0.2306 (0.06)	6	0.1930 (0.04)		N/A
$\rho = 0.6$	**1**	**0.0233 (0.02)**	2	0.0603 (0.09)	8	0.7612 (0.00)	3	0.0790 (0.03)	5	0.1769 (0.03)	2	0.0543 (0.03)	4	0.1336 (0.04)	7	0.2152 (0.07)	6	0.1940 (0.06)	6	0.1887 (0.04)		N/A
$\rho = 0.7$	**1**	**0.0253 (0.02)**	2	0.0605 (0.08)	7	0.7612 (0.00)	3	0.0722 (0.03)	6	0.1769 (0.03)	2	0.0501 (0.02)	4	0.1336 (0.04)	6	0.1762 (0.06)	5	0.1627 (0.06)	6	0.1819 (0.04)		N/A
$\rho = 0.8$	**1**	**0.0247 (0.02)**	3	0.0606 (0.08)	8	0.7612 (0.00)	3	0.0613 (0.03)	7	0.1769 (0.03)	2	0.0455 (0.02)	5	0.1336 (0.04)	5	0.1303 (0.05)	4	0.1194 (0.05)	6	0.1704 (0.03)		N/A
$\rho = 0.9$	**1**	**0.0227 (0.02)**	4	0.0602 (0.09)	9	0.7612 (0.00)	3	0.0470 (0.02)	8	0.1769 (0.03)	2	0.0409 (0.02)	6	0.1336 (0.04)	5	0.0930 (0.04)	5	0.0867 (0.04)	7	0.1568 (0.03)		N/A
DENSE2																						
$\rho = 0.1$	2	0.1007 (0.16)	4	0.2614 (0.14)	7	0.7664 (0.00)	3	0.2016 (0.03)	2	0.1104 (0.03)	5	0.4090 (0.04)	**1**	**0.0909 (0.02)**	3	0.2059 (0.17)	6	0.4956 (0.08)	4	0.2434 (0.02)		N/A
$\rho = 0.3$	**1**	**0.0460 (0.03)**	2	0.0744 (0.02)	9	0.7664 (0.00)	5	0.1463 (0.02)	4	0.1104 (0.03)	7	0.2131 (0.02)	3	0.0909 (0.02)	2	0.0753 (0.05)	8	0.3323 (0.06)	6	0.1880 (0.02)		N/A
$\rho = 0.5$	**1**	**0.0268 (0.02)**	4	0.1130 (0.07)	9	0.7664 (0.00)	5	0.1668 (0.07)	4	0.1104 (0.03)	6	0.1900 (0.03)	3	0.0909 (0.02)	2	0.0661 (0.04)	8	0.2616 (0.08)	7	0.2056 (0.06)		N/A
$\rho = 0.7$	**1**	**0.0356 (0.03)**	2	0.0543 (0.05)	8	0.7664 (0.00)	4	0.0988 (0.03)	5	0.1104 (0.03)	7	0.1433 (0.03)	3	0.0909 (0.02)	2	0.0490 (0.07)	6	0.1309 (0.04)	7	0.1407 (0.03)		N/A
$\rho = 0.9$	**1**	**0.0242 (0.01)**	4	0.0662 (0.06)	8	0.7664 (0.00)	3	0.0505 (0.05)	7	0.1104 (0.03)	5	0.0770 (0.05)	6	0.0909 (0.02)	**1**	**0.0263 (0.02)**	2	0.0381 (0.03)	6	0.0869 (0.05)		N/A
DENSE3																						
$\rho = 0.1$	2	0.0917 (0.02)	5	0.3388 (0.12)	7	0.8235 (0.01)	2	0.0872 (0.06)	3	0.1234 (0.06)	3	0.1168 (0.03)	4	0.1717 (0.04)	8	1.0975 (0.32)	8	1.0917 (0.32)	**1**	**0.0442 (0.04)**	6	0.4180 (0.06)
$\rho = 0.2$	**1**	**0.0476 (0.06)**	6	0.3063 (0.16)	8	0.8235 (0.01)	4	0.1413 (0.08)	3	0.1234 (0.06)	3	0.1168 (0.03)	5	0.1717 (0.04)	7	0.5681 (0.22)	7	0.5521 (0.22)	2	0.0721 (0.08)	6	0.3076 (0.04)
$\rho = 0.3$	**1**	**0.0410 (0.06)**	5	0.1478 (0.18)	9	0.8235 (0.01)	5	0.1457 (0.07)	4	0.1234 (0.06)	3	0.1096 (0.03)	6	0.1717 (0.04)	8	0.5043 (0.20)	8	0.4768 (0.19)	2	0.0766 (0.07)	7	0.2401 (0.04)
$\rho = 0.4$	**1**	**0.0378 (0.05)**	4	0.1507 (0.16)	8	0.8235 (0.01)	4	0.1445 (0.05)	3	0.1234 (0.06)	2	0.0782 (0.03)	5	0.1717 (0.04)	7	0.3735 (0.16)	7	0.4011 (0.17)	2	0.0754 (0.06)	6	0.2039 (0.04)
$\rho = 0.5$	**1**	**0.0505 (0.03)**	5	0.1707 (0.15)	8	0.8235 (0.01)	**1**	**0.0532 (0.07)**	4	0.1234 (0.06)	2	0.0599 (0.03)	5	0.1717 (0.04)	6	0.3804 (0.17)	7	0.6713 (0.26)	3	0.0677 (0.03)	5	0.1863 (0.05)
$\rho = 0.6$	2	0.0922 (0.04)	4	0.1447 (0.16)	8	0.8235 (0.01)	3	0.1235 (0.04)	3	0.1234 (0.06)	**1**	**0.0782 (0.03)**	5	0.1717 (0.04)	7	0.5439 (0.25)	9	1.3066 (0.53)	5	0.1684 (0.08)	6	0.1844 (0.05)
$\rho = 0.7$	2	0.1050 (0.06)	**1**	**0.0769 (0.18)**	9	0.8235 (0.01)	5	0.1702 (0.08)	3	0.1234 (0.06)	**1**	**0.0860 (0.03)**	5	0.1717 (0.04)	6	0.1938 (0.11)	8	0.5025 (0.19)	7	0.2248 (0.10)	4	0.1522 (0.04)
$\rho = 0.8$	**1**	**0.1085 (0.06)**	**1**	**0.0892 (0.18)**	6	0.8235 (0.01)	4	0.1790 (0.09)	2	0.1234 (0.06)	**1**	**0.0938 (0.03)**	4	0.1717 (0.04)	**1**	**0.0979 (0.05)**	5	0.2219 (0.09)	5	0.2388 (0.11)	3	0.1407 (0.03)
$\rho = 0.9$	3	0.1129 (0.06)	3	0.1140 (0.18)	6	0.8235 (0.01)	4	0.1800 (0.09)	3	0.1234 (0.06)	2	0.0983 (0.03)	4	0.1717 (0.04)	**1**	**0.0879 (0.05)**	5	0.2463 (0.14)	5	0.2394 (0.10)	2	0.0950 (0.02)
DENSE4																						
$\rho = 0.1$	2	0.2769 (0.13)	5	0.9620 (1.03)	4	0.7228 (0.15)	3	0.5310 (0.62)	3	0.5223 (0.58)	3	0.5310 (0.62)	3	0.5074 (0.50)	**1**	**0.2217 (0.11)**	3	0.6085 (0.83)	3	0.5351 (0.64)	3	0.5995 (0.73)
$\rho = 0.9$	**1**	**0.0688 (0.06)**	3	0.5592 (0.76)	4	0.7228 (0.15)	3	0.5449 (0.70)	3	0.5223 (0.58)	3	0.5431 (0.69)	3	0.5074 (0.50)	2	0.1049 (0.08)	3	0.5611 (0.77)	3	0.5491 (0.71)	3	0.5830 (0.72)
DENSE5																						
$\rho = 0.2$	2	0.2789 (0.20)	5	0.9832 (0.64)	4	0.7735 (0.07)	2	0.2705 (0.36)	**1**	**0.2335 (0.28)**	2	0.2687 (0.36)	**1**	**0.2282 (0.24)**	2	0.3106 (0.19)	3	0.5482 (0.50)	2	0.2736 (0.36)	3	0.5182 (0.49)
$\rho = 0.4$	**1**	**0.1289 (0.07)**	3	0.2752 (0.36)	5	0.7735 (0.07)	3	0.2509 (0.33)	3	0.2335 (0.28)	3	0.2611 (0.34)	3	0.2282 (0.24)	2	0.1865 (0.16)	4	0.4053 (0.45)	3	0.2531 (0.33)	4	0.4567 (0.47)
$\rho = 0.6$	**1**	**0.1379 (0.07)**	3	0.2803 (0.37)	5	0.7735 (0.07)	3	0.2716 (0.36)	3	0.2335 (0.28)	3	0.2666 (0.35)	3	0.2282 (0.24)	2	0.1681 (0.15)	4	0.3659 (0.43)	3	0.2747 (0.37)	4	0.4211 (0.45)
$\rho = 0.8$	**1**	**0.1231 (0.09)**	3	0.2336 (0.28)	6	0.7735 (0.07)	4	0.2592 (0.34)	3	0.2335 (0.28)	4	0.2609 (0.34)	3	0.2282 (0.24)	2	0.1550 (0.14)	4	0.2978 (0.39)	4	0.2617 (0.35)	5	0.3902 (0.44)
DENSE6																						
$\rho = 0.1$	2	0.0468 (0.05)	5	0.1093 (0.06)	9	0.8125 (0.01)	5	0.1052 (0.05)	3	0.0817 (0.05)	**1**	**0.0335 (0.05)**	6	0.1556 (0.05)	8	0.4694 (0.15)	7	0.4030 (0.08)	4	0.0966 (0.05)		N/A
$\rho = 0.2$	2	0.0444 (0.05)	3	0.0673 (0.06)	9	0.8125 (0.01)	5	0.1111 (0.05)	4	0.0817 (0.05)	**1**	**0.0338 (0.05)**	6	0.1556 (0.05)	8	0.3685 (0.11)	7	0.3130 (0.07)	5	0.1024 (0.05)		N/A
$\rho = 0.3$	**1**	**0.0397 (0.05)**	2	0.0581 (0.03)	8	0.8125 (0.01)	4	0.1112 (0.05)	3	0.0817 (0.05)	**1**	**0.0373 (0.03)**	5	0.1556 (0.05)	7	0.2987 (0.11)	6	0.2441 (0.07)	4	0.1025 (0.05)		N/A
$\rho = 0.4$	**1**	**0.0390 (0.05)**	2	0.0525 (0.01)	8	0.8125 (0.01)	4	0.1097 (0.05)	3	0.0817 (0.05)	**1**	**0.0431 (0.02)**	5	0.1556 (0.05)	7	0.2448 (0.11)	6	0.1996 (0.07)	4	0.1010 (0.05)		N/A
$\rho = 0.5$	**1**	**0.0338 (0.05)**	3	0.0497 (0.01)	9	0.8125 (0.01)	5	0.1082 (0.05)	4	0.0817 (0.05)	2	0.0447 (0.01)	6	0.1556 (0.05)	8	0.3111 (0.01)	7	0.1891 (0.07)	5	0.0995 (0.05)		N/A
$\rho = 0.6$	**1**	**0.0328 (0.05)**	2	0.0468 (0.01)	7	0.8125 (0.01)	4	0.1067 (0.05)	3	0.0817 (0.05)	2	0.0457 (0.01)	5	0.1556 (0.05)	6	0.2830 (0.04)	5	0.1627 (0.06)	4	0.0980 (0.05)		N/A
$\rho = 0.7$	**1**	**0.0327 (0.05)**	2	0.0446 (0.01)	7	0.8125 (0.01)	4	0.0965 (0.05)	3	0.0817 (0.05)	2	0.0461 (0.01)	5	0.1556 (0.05)	6	0.3081 (0.03)	5	0.1449 (0.06)	4	0.0896 (0.05)		N/A
$\rho = 0.8$	**1**	**0.0327 (0.05)**	2	0.0413 (0.02)	8	0.8125 (0.01)	4	0.0893 (0.05)	4	0.0817 (0.05)	3	0.0457 (0.01)	6	0.1556 (0.05)	7	0.2225 (0.01)	5	0.1324 (0.06)	4	0.0806 (0.05)		N/A
$\rho = 0.9$	**1**	**0.0325 (0.05)**	8	0.3409 (0.04)	9	0.8125 (0.01)	4	0.0796 (0.05)	4	0.0817 (0.05)	2	0.0450 (0.01)	6	0.1556 (0.05)	7	0.2937 (0.20)	5	0.0927 (0.06)	3	0.0694 (0.05)		N/A
MOE1																						
$\rho = 0.1$	**1**	**0.0195 (0.01)**	7	0.1557 (0.01)	10	0.7689 (0.00)	4	0.0565 (0.02)	6	0.1392 (0.02)	3	0.0527 (0.02)	2	0.0362 (0.02)	8	0.2206 (0.02)	9	0.2317 (0.02)	5	0.0713 (0.02)		N/A
$\rho = 0.2$	**1**	**0.0213 (0.01)**	6	0.1606 (0.01)	9	0.7689 (0.00)	3	0.0518 (0.02)	5	0.1392 (0.02)	8	0.2214 (0.02)	2	0.0362 (0.02)	7	0.2061 (0.02)	7	0.2085 (0.02)	4	0.0666 (0.02)		N/A
$\rho = 0.3$	**1**	**0.0206 (0.01)**	4	0.1644 (0.01)	9	0.7689 (0.00)	5	0.1895 (0.02)	3	0.1392 (0.02)	8	0.2161 (0.02)	2	0.0362 (0.02)	6	0.1970 (0.02)	6	0.2006 (0.02)	7	0.2076 (0.02)		N/A
$\rho = 0.4$	**1**	**0.0385 (0.02)**	2	0.0574 (0.02)	9	0.7689 (0.00)	3	0.0856 (0.02)	6	0.1392 (0.02)	5	0.1078 (0.02)	**1**	**0.0362 (0.02)**	7	0.1512 (0.02)	8	0.1677 (0.02)	4	0.1014 (0.02)		N/A
$\rho = 0.5$	**1**	**0.0192 (0.01)**	**1**	**0.0219 (0.01)**	8	0.7689 (0.00)	3	0.0476 (0.02)	7	0.1392 (0.02)	4	0.0626 (0.02)	2	0.0362 (0.02)	5	0.1228 (0.02)	6	0.1323 (0.02)	4	0.0610 (0.02)		N/A
$\rho = 0.6$	**1**	**0.0131 (0.01)**	3	0.0540 (0.02)	9	0.7689 (0.00)	4	0.0708 (0.02)	8	0.1392 (0.02)	5	0.0864 (0.02)	2	0.0362 (0.02)	6	0.1143 (0.02)	7	0.1231 (0.02)	5	0.0856 (0.02)		N/A
$\rho = 0.7$	**1**	**0.0145 (0.01)**	3	0.0445 (0.02)	8	0.7689 (0.00)	4	0.0633 (0.02)	7	0.1392 (0.02)	5	0.0775 (0.02)	2	0.0362 (0.02)	6	0.0962 (0.02)	6	0.0957 (0.02)	5	0.0781 (0.02)		N/A
$\rho = 0.8$	**1**	**0.0138 (0.01)**	2	0.0350 (0.02)	7	0.7689 (0.00)	3	0.0538 (0.02)	6	0.1392 (0.02)	4	0.0666 (0.02)	2	0.0362 (0.02)	5	0.0815 (0.02)	5	0.0815 (0.02)	4	0.0686 (0.02)		N/A
$\rho = 0.9$	**1**	**0.0130 (0.01)**	3	0.0408 (0.02)	9	0.7689 (0.00)	5	0.0482 (0.02)	8	0.1392 (0.02)	6	0.0525 (0.02)	2	0.0362 (0.02)	2	0.0342 (0.01)	4	0.0454 (0.02)	7	0.0630 (0.02)		N/A
MOE2																						
$\rho = 0.1$	**1**	**0.0385 (0.02)**	7	0.7119 (0.05)	8	0.8405 (0.03)	4	0.3419 (0.11)	2	0.0903 (0.16)	2	0.0796 (0.15)	3	0.2340 (0.13)	4	0.3285 (0.08)	6	0.4785 (0.10)	5	0.4244 (0.10)		N/A
$\rho = 0.3$	**1**	**0.0264 (0.02)**	8	0.6200 (0.07)	9	0.8405 (0.03)	5	0.3269 (0.11)	2	0.0903 (0.16)	2	0.0791 (0.15)	4	0.2340 (0.13)	3	0.2093 (0.05)	7	0.4547 (0.10)	6	0.4094 (0.10)		N/A
$\rho = 0.5$	**1**	**0.0944 (0.01)**	7	0.4908 (0.09)	8	0.8405 (0.03)	4	0.2968 (0.12)	**1**	**0.0903 (0.16)**	**1**	**0.0788 (0.15)**	3	0.2340 (0.13)	2	0.1338 (0.04)	6	0.4154 (0.10)	5	0.3793 (0.11)		N/A
$\rho = 0.7$	2	0.0994 (0.01)	6	0.5827 (0.07)	7	0.8405 (0.03)	4	0.2749 (0.12)	2	0.0903 (0.16)	2	0.0784 (0.15)	3	0.2340 (0.13)	**1**	**0.0580 (0.02)**	4	0.2633 (0.12)	5	0.3573 (0.11)		N/A
$\rho = 0.9$	2	0.0377 (0.01)	4	0.2493 (0.13)	6	0.8405 (0.03)	4	0.2508 (0.13)	3	0.0903 (0.16)	3	0.0782 (0.15)	4	0.2340 (0.13)	**1**	**0.0247 (0.01)**	4	0.2412 (0.13)	5	0.3332 (0.11)		N/A
Average r	**1.23**		3.62		7.77		3.67		4.13		3.02		3.81		5.02		5.94		4.69		4.47	

settings under a dataset-budget pair, we report on average MAPE with standard deviation. To further validate the **practical utility of ReScaL for hyperparameter configuration selection**, reporting the gap between the predicted-best and true best configurations in the testing dataset using Spearman correlation (Sp), pairwise ranking accuracy (PairAcc), top-1 selection accuracy (Top1), regret (Reg), and normalized regret (NReg). Here, regret is the excess observed performance loss incurred by the predicted-best configuration against the true best configuration; NReg is its normalized version.

To verify the statistical significance on error differences over all test scale settings under a case, we employ the Scott-Knott Effect Size Difference (ESD) test (Scott & Knott, 1974)—a hierarchical clustering algorithm that gives ranks r to methods by putting them into statistically different groups without overlap, e.g., three methods with $r = 1$ for $\{A, B\}$ and $r = 2$ for $\{C\}$ means that A and B are statistically similar but they both significantly outperform C.

4.3 RQ1: ReScaL vs. State-of-the-art Methods

Method. We compare ReScaL against the following baselines and state-of-the-art methods: (1) individual scaling laws Farseer, Chinchilla, and Kaplan **fully fitted** on our pilot set data; (2) their **default-parameter variants** $\text{Farseer}_{\text{d}}$, $\text{Chinchilla}_{\text{d}}$, and Kaplan_{d}; (3) **RF** using all features $(N, D, \mathbf{x})$ and $\mathbf{RF}_{ND}$ using only (N, D); (4) **NCPL**, a neural configuration-to-performance scaling law (Zhang et al., 2026), adapted to the same training splits and with its Chinchilla component refitted on each pilot set; and (5) where available, a **HUMAN** baseline of expert-designed functions. More details are in Appendix B.3.

Results. Table 2 compares ReScaL with all baselines across 52 dataset-budget cases, with HUMAN reported on its available cases. Overall, ReScaL attains rank 1 in 41/52 cases (79%) and achieves the best average rank (1.23), ahead of the strongest fully covered baselines (Kaplan: 3.02; Chinchilla: 3.67; NCPL: 4.69). The partially covered HUMAN baseline has an average rank of 4.47 over its 15 available cases. ReScaL also obtains the lowest mean MAPE in 37/52 cases (71%), showing consistently strong performance across datasets and pilot budgets. Additional results using MAE and RMSE, which confirm the same conclusions, are provided in Appendix C.2.

The gains are most visible when pilot data are scarce. On DENSE1, ReScaL reduces MAPE from 0.0893 (best baseline Kaplan) to 0.0203 at $\rho = 0.1$, and from 0.0764 (best baseline Farseer) to 0.0135 at $\rho = 0.2$. On MOE2, it also improves early-budget accuracy, e.g., 0.0264 vs. 0.0791 for the best baseline Kaplan at $\rho = 0.3$. For the HUMAN baseline, available on DENSE3–5, ReScaL outperforms it in 14/15 cases, suggesting that it can match or exceed expert-designed functions while remaining automatic. Further analysis on ReScaL's robustness can be found in Appendix A.4.

Table 3: Decision quality over all 52 dataset–budget cases; Best mean ranks are **bold**. The format is the same as Table 2. More detailed results can be found in Appendix C.3.

Method	Sp ↑		PairAcc ↑		Top1 ↑		Reg ↓		NReg ↓	
	Mean r	Value	Mean r	Value	Mean r	Value	Mean r	Value	Mean r	Value
ReScaL	**1.12**	**0.646**	**1.25**	**0.773**	**1.25**	0.462	**1.38**	**0.043**	**1.25**	**0.143**
RF	1.75	0.368	1.75	0.662	1.38	0.327	**1.38**	0.118	1.38	0.237
RF$_{ND}$	2.75	0.135	2.75	0.534	2.12	0.000	2.25	1.194	2.12	0.488
Farseer$_{d}$	2.00	0.382	2.00	0.645	1.75	0.173	1.75	1.067	1.75	0.347
Kaplan$_{d}$	2.00	0.365	2.12	0.628	1.75	0.173	1.88	1.067	1.75	0.347
Chinchilla$_{d}$	2.25	0.330	2.25	0.611	2.12	0.000	1.88	1.068	1.75	0.352
Kaplan	2.12	0.365	2.12	0.628	1.88	0.173	1.88	1.067	1.75	0.347
Chinchilla	2.12	0.316	2.12	0.627	1.75	0.135	1.88	1.079	1.75	0.373
Farseer	2.38	0.170	2.50	0.575	1.75	0.173	2.38	1.383	2.25	0.537
NCPL	2.62	0.087	2.75	0.510	1.88	0.173	2.25	0.364	2.12	0.352
HUMAN	2.33	0.575	2.33	0.773	1.33	0.600	2.00	0.688	2.00	0.284

Furthermore, Table 3 shows that ReScaL achieves the best mean Scott–Knott rank on all five decision-oriented metrics. It correctly orders 77.3% of candidate pairs, selects the true best configuration in 46.2% of cases, and obtains a mean normalized regret of 0.143. These results show that ReScaL performs well not only in prediction accuracy but also in configuration ranking and selection, supporting its practical utility.

4.4 RQ2: ABLATION STUDY

Method. We remove exactly one component from diverse fitting, scaling law ensemble, residual learning, and dynamic α in ReScaL, while keeping the others unchanged.

Table 4: Ablation study showing contribution of each ReScaL component across all budget sizes. The format is the same as Table 2. More detailed results can be found in Appendix C.4.

Dataset	ReScaL		w/o dynamic α		w/o ensemble		w/o residual		w/o diverse fitting	
	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)
DENSE1	**1**	**0.0215 (0.01)**	4	0.0822 (0.04)	2	0.0291 (0.02)	**1**	**0.0217 (0.01)**	3	0.0465 (0.02)
DENSE2	**1**	**0.0467 (0.08)**	**1**	**0.0544 (0.10)**	**1**	**0.0452 (0.07)**	2	0.0657 (0.04)	**1**	**0.052 (0.08)**
DENSE3	**1**	**0.0767 (0.06)**	2	0.1306 (0.07)	2	0.1473 (0.17)	**1**	**0.0787 (0.06)**	**1**	**0.0868 (0.06)**
DENSE4	**1**	**0.1728 (0.14)**	**1**	**0.1586 (0.14)**	**1**	**0.1731 (0.15)**	2	0.4661 (0.21)	**1**	**0.1676 (0.15)**
DENSE5	**1**	**0.1674 (0.14)**	**1**	**0.1467 (0.11)**	2	0.1965 (0.18)	3	0.4529 (0.16)	**1**	**0.1647 (0.13)**
DENSE6	**1**	**0.0365 (0.04)**	3	0.0764 (0.04)	2	0.0607 (0.09)	**1**	**0.0366 (0.04)**	**1**	**0.0424 (0.05)**
MOE1	**1**	**0.0191 (0.01)**	4	0.0823 (0.07)	2	0.0265 (0.02)	2	0.0258 (0.02)	3	0.0564 (0.04)
MOE2	**1**	**0.059 (0.03)**	2	0.0851 (0.07)	**1**	**0.059 (0.03)**	3	0.1389 (0.15)	3	0.1398 (0.10)
Average r	**1.00**		2.25		1.63		1.88		1.75	

Results. Table 4 reports Scott-Knott ranks and mean MAPE for ReScaL and four ablated variants across all budget sizes for each dataset. The full ReScaL achieves rank 1 on all eight datasets, while removing dynamic α gives the worst average rank (2.25), followed by removing residual learning (1.88). This shows that adaptive weighting is the most important component overall, and that residual learning is also central to correcting non-scaling effects.

The component effects are dataset-dependent but consistent with the design. Removing dynamic α causes the largest degradations on DENSE1 (0.0215→0.0822; ∼3.8×) and MOE1 (0.0191→0.0823; ∼4.3×), showing its role in controlling residual influence. Removing residual learning is most harmful on feature-rich datasets such as DENSE4 and DENSE5 (∼2.7× each), while ensembling is most useful on DENSE3 and diverse fitting is especially important on MoE datasets. Although a few ablations match or slightly outperform ReScaL on isolated datasets, none improves overall robustness. Additional results, including MAE and RMSE, more detailed evaluations, and results under diverse fixed α values, are provided in Appendices C.4 and C.1.

4.5 RQ3: SENSITIVITY ANALYSIS

Method. We analyze sensitivity to the combination of temperature t (controlling the softmax ensemble) and the sigmoid scaling s (controlling the steepness of the sigmoid function in α for residual-augmented prediction), which are key hyperparameters of ReScaL. We examine how different values of s interact with ensemble temperature t through full combinatorial settings of $t \in \{0.01, 0.1, 0.5, 1.0\}$ and $s \in \{1, 10, 20, 50, 100\}$. Other setups remain unchanged.

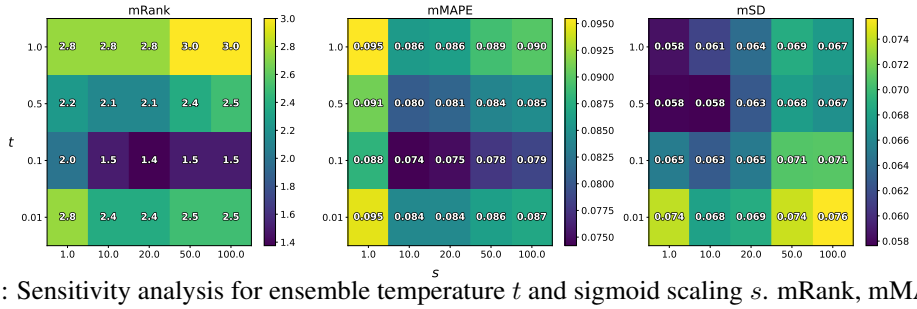


Figure 3: Sensitivity analysis for ensemble temperature t and sigmoid scaling s. mRank, mMAPE, and mSD denote the mean Scott-Knott rank, mean MAPE, and mean standard deviation across all cases, respectively. Detailed results can be found in Appendix C.5.

Results. Figure 3 shows a clear sweet spot around temperature $t = 0.1$. The best mRank is achieved at $(t = 0.1, s = 20)$ (mRank$= 1.4$), while the lowest mMAPE occurs at $(t = 0.1, s = 10)$ (mMAPE$= 0.074$). This supports the intuition that a moderate temperature avoids over-committing to a single scaling law while still emphasizing stronger laws, and that moderate s values provide useful but not overly brittle residual gating.

The sensitivity results also show a trade-off between accuracy and stability. Larger t with smaller s gives the most stable predictions (e.g., mSD$= 0.058$ at $(t = 0.5, s = 10)$), but with worse accuracy; very low temperature $(t = 0.01)$ is consistently suboptimal because it behaves like hard law selection and amplifies pilot-set noise. Overall, the default $(t = 0.1, s = 20)$ is a robust compromise: it gives the best mRank, nearly the best mMAPE (0.075 vs. 0.074), and avoids the most unstable settings.

4.6 Discussion

Why does `ReScaL` Work? The results suggest that augmenting scaling law priors with residual learning is practical for LLM pretraining performance prediction. Scaling laws provide a strong extrapolation backbone on (N, D), while the residual model corrects deviations therein. The ensemble stabilizes predictions by weighting laws based on their fit to the pilot regime, and the pilot-set-level gate controls feature-dependent corrections when extrapolating beyond limited pilot data.

Computational Overhead. The fitting overhead of `ReScaL` itself is small on a standard server: $1.6s$–$9.5s$ for training per dataset-budget case and $0.007s$–$0.09s$ per target prediction (Appendix C.6). Separately, we evaluate pilot-sample efficiency by break-even analysis. `ReScaL` reaches or exceeds that accuracy in 98.1% of cases, does so with no greater pilot compute in 84.6%, and yields a median compute saving of 82.1% among matched cases (Appendix C.7).

Limitations. `ReScaL` has several limitations: First, the cost proxy $C \approx 6ND$ is a simplified estimate that ignores, e.g., hardware and system-level overheads. While suitable for ordering runs by approximate resource cost, it may not match wall-clock time in all training setups. Second, the benefit of residual learning depends on the richness of features: with a small feature set, the residual model would be unlikely to improve over strong scaling laws.

5 Conclusion

This paper introduces `ReScaL`, a residual-augmented scaling-law method for LLM performance prediction. It combines fully and partially fitted analytical laws with feature-aware residual learning and a pilot-set-level gate, enabling small-scale runs to forecast expensive large-scale validation loss. Across 52 cases, the gains extend beyond point-wise error to configuration ranking and selection, including lower normalized regret than the compared predictors. These results support using `ReScaL` to prioritize promising configurations before full-scale training, while the observed system dependence underscores the need to validate feature coverage in each deployment. Future work could explore cross-LLM generalization, uncertainty quantification, and automatic feature discovery.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks with required disclosure. We used generative AI tools to edit software code and to improve the readability of the paper, which are tasks with recommended disclosure. We have reviewed all AI-assisted work. AI-assisted code edits were verified and tested for correctness by the authors, and AI-assisted edits to the paper were reviewed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies prediction methods using previously published aggregate results from LLM pretraining experiments. It does not involve human subjects or sensitive data.

REPRODUCIBILITY STATEMENT

Section 4.2 describes the datasets, evaluation protocol, baselines, hyperparameters, metrics, and statistical testing used in the experiments. Additional details on features, data selection, complete results, computational overhead, and reproducibility are provided in Appendices B–D. The datasets, scripts, and full experimental results are also available through our anonymous repository at `https://anonymous.4open.science/r/ReScaL-8D70`.

REFERENCES

Ehsan Ataie, Athanasia Evangelinou, Eugenio Gianniti, and Danilo Ardagna. A hybrid machine learning approach for performance modeling of cloud-based big data applications. volume 65, pp. 3123–3140, 2022. doi: 10.1093/COMJNL/BXAB131. URL https://doi.org/10.1093/comjnl/bxab131.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. volume 121, pp. e2311878121. National Academy of Sciences, 2024.

Bowen Baker, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Accelerating neural architecture search using performance prediction. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018.

Tamay Besiroglu, Ege Erdil, Matthew Barnett, and Josh You. Chinchilla scaling: A replication attempt. volume abs/2404.10102, 2024. doi: 10.48550/ARXIV.2404.10102. URL https://doi.org/10.48550/arXiv.2404.10102.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. volume 16, pp. 1190–1208, 1995. doi: 10.1137/0916069. URL https://doi.org/10.1137/0916069.

Pengzhou Chen, Jingzhi Gong, and Tao Chen. Accuracy can lie: On the impact of surrogate model in configuration tuning. volume 51, pp. 548–580, 2025a. doi: 10.1109/TSE.2025.3525955. URL https://doi.org/10.1109/TSE.2025.3525955.

Yangyi Chen, Binxuan Huang, Yifan Gao, Zhengyang Wang, Jingfeng Yang, and Heng Ji. Scaling laws for predicting downstream performance in llms. volume 2025, 2025b.

Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and Jingang Wang. Revisiting scaling laws for language models: The role of data quality and training strategies. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 23881–23899. Association for Computational Linguistics, 2025c. URL https://aclanthology.org/2025.acl-long.1163/.

Leshem Choshen, Yang Zhang, and Jacob Andreas. A hitchhiker's guide to scaling law estimation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/choshen25a.html.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake A. Hechtman, Trevor Cai, Sebastian Borgeaud, George van den Driessche, Eliza Rutherford, Tom Hennigan, Matthew J. Johnson, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Marc'Aurelio Ranzato, Jack W. Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pp. 4057–4086. PMLR, 2022. URL https://proceedings.mlr.press/v162/clark22a.html.

Diego Didona, Francesco Quaglia, Paolo Romano, and Ennio Torre. Enhancing performance prediction robustness by combining analytical modeling and machine learning. In Lizy K. John, Connie U. Smith, Kai Sachs, and Catalina M. Lladó (eds.), *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering, Austin, TX, USA, January 31 - February 4, 2015*, pp. 145–156. ACM, 2015. doi: 10.1145/2668930.2688047. URL https://doi.org/10.1145/2668930.2688047.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In Qiang Yang and Michael J. Wooldridge (eds.), *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pp. 3460–3468. AAAI Press, 2015. URL http://ijcai.org/Abstract/15/487.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Luca Soldaini, Jenia Jitsev, Alex Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, and et al. Language models scale reliably with over-training and on downstream tasks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Jingzhi Gong and Tao Chen. Deep configuration performance learning: A systematic survey and taxonomy. volume 34, pp. 25:1–25:62, 2025. doi: 10.1145/3702986. URL https://doi.org/10.1145/3702986.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research, pp. 1321–1330. PMLR, 2017. URL http://proceedings.mlr.press/v70/guo17a.html.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. volume abs/2010.14701, 2020. URL https://arxiv.org/abs/2010.14701.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. volume abs/1712.00409, 2017. URL http://arxiv.org/abs/1712.00409.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. volume abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. volume 63, pp. 425–464. Wiley Online Library, 2001.

Aaron Klein, Stefan Falkner, Jost Tobias Springenberg, and Frank Hutter. Learning curve prediction with bayesian neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Houyi Li, Wenzhen Zheng, Jingcheng Hu, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. Predictable scale: Part I - optimal hyperparameter scaling law in large language model pretraining. volume abs/2503.04715, 2025a. doi: 10.48550/ARXIV.2503.04715. URL https://doi.org/10.48550/arXiv.2503.04715.

Houyi Li, Wenzhen Zheng, Qiufeng Wang, Zhenyu Ding, Haoying Wang, Zili Wang, Shijie Xuyang, Ning Ding, Shuigeng Zhou, Xiangyu Zhang, et al. Predictable scale (part ii)—farseer: A refined scaling law in llms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025b.

Margaret Li, Sneha Kudugunta, and Luke Zettlemoyer. (mis)fitting scaling laws: A survey of scaling law fitting techniques in deep learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025c.

Stuart R Lipsitz, Traci Leong, Joseph Ibrahim, and Steven Lipshultz. A partial correlation coefficient and coefficient of determination for multivariate normal repeated measures data. volume 50, pp. 87–95. Wiley Online Library, 2001.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A. Raffel. Scaling data-constrained language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/9d89448b63ce1e2e8dc7af72c984c196-Abstract-Conference.html.

Andrew Jhon Scott and Martin Knott. A cluster analysis method for grouping means in the analysis of variance. pp. 507–512. JSTOR, 1974.

Ayan Sengupta, Yash Goel, and Tanmoy Chakraborty. How to upscale neural networks with scaling law? A survey and practical guidelines. volume abs/2502.12051, 2025. doi: 10.48550/ARXIV.2502.12051. URL https://doi.org/10.48550/arXiv.2502.12051.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13969–13980, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html.

Symone G. Soares and Rui Araújo. A dynamic and on-line ensemble regression for changing environments. volume 42, pp. 2935–2948, 2015. doi: 10.1016/J.ESWA.2014.11.053. URL https://doi.org/10.1016/j.eswa.2014.11.053.

Shreyas Subramanian, Vignesh Ganapathiraman, and Corey Barrett. Hop, skip, jump to convergence: Dynamics of learning rate transitions for improved training of large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Findings of ACL, pp. 16349–16362. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.954. URL https://doi.org/10.18653/v1/2024.findings-emnlp.954.

Anirudh Subramanyam, Yuxin Chen, and Robert L. Grossman. Scaling laws revisited: Modeling the role of data quality in language model pretraining. volume abs/2510.03313, 2025. doi: 10.48550/ARXIV.2510.03313. URL https://doi.org/10.48550/arXiv.2510.03313.

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q. Tran, Dani Yogatama, and Donald Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pp. 12342–12364. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.825. URL https://doi.org/10.18653/v1/2023.findings-emnlp.825.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. volume abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, Proceedings of Machine Learning Research, pp. 10248–10259. PMLR, 2020. URL http://proceedings.mlr.press/v119/wenzel20a.html.

Colin White, Arber Zela, Robin Ru, Yang Liu, and Frank Hutter. How powerful are performance predictors in neural architecture search? In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 28454–28469, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ef575e8837d065a1683c022d2077d342-Abstract.html.

Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, JMLR Workshop and Conference Proceedings, pp. 1067–1075. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/wilson13.html.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Huaqing Zhang, Kaiyue Wen, and Tengyu Ma. Neural configuration to performance scaling law. *arXiv preprint arXiv:2602.10300*, 2026. URL https://arxiv.org/abs/2602.10300.

Zhi-Hua Zhou and Ji Feng. Deep forest: Towards an alternative to deep neural networks. In Carles Sierra (ed.), *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 3553–3559. ijcai.org,

2017. doi: 10.24963/IJCAI.2017/497. URL https://doi.org/10.24963/ijcai.2017/497.

APPENDIX

A Additional Discussion and Justification

A.1 The Source of Historical Parameters for Diverse Fitting

The historical parameters $\boldsymbol{\theta}_k^0$ used in diverse fitting are sourced from the default values reported by the original papers of each scaling law formulation (Kaplan et al., 2020; Hoffmann et al., 2022; Li et al., 2025b). These are domain-general estimates fitted on large-scale pretraining data from the original studies, not parameters learned from the current dataset's test set or other datasets in our evaluation. This ensures no data leakage: the warm-start values are public, pre-existing parameters that serve as informative priors when pilot data are limited, following the recommendation in (Choshen et al., 2025; Li et al., 2025c) to leverage prior estimates for constrained estimation.

For partially fitted variant laws, only a subset of parameters (e.g., N-related or D-related groups) are fitted to the pilot data while keeping others fixed, as detailed in Section 3.1. This warm-start strategy strikes a balance between the stability provided by historical priors and the adaptability to new pilot data, preventing the instability of full re-fitting on scarce pilot sets while still capturing meaningful dataset-specific scaling behavior.

A.2 Justification of the Parameter Constraints for Partially Fitted Laws

The parameter constraints ($\pm 1\%$ and $\pm 35\%$) in diverse fitting are motivated by the principle of constrained estimation for limited data (Choshen et al., 2025; Li et al., 2025c), which integrates strong historical priors for stability while allowing targeted adaptation to new pilot signals. The $\pm 1\%$ constraint on fixed parameters ensures that parameters retained from the original scaling law formulations remain close to their well-validated values, providing a stable prior when pilot data are scarce. The $\pm 35\%$ constraint on adapted parameters allows meaningful adaptation to dataset-specific scaling behaviors (e.g., changing an exponent from 0.34 to 0.46) while preventing overfitting to noise in limited data. The asymmetry reflects the design philosophy: fixed parameters offer stability by anchoring the model with strong priors, while adapted parameters introduce flexibility through controlled, data-driven adjustments.

Although these specific bounds are not derived from a formal sensitivity analysis, they are consistent with standard practices in constrained optimization under limited data (Byrd et al., 1995). Moreover, the effectiveness is supported empirically: our results show consistent improvements across diverse datasets and pilot budget sizes (Section 4.3), and the ablation study (Section 4.4) demonstrates that diverse fitting (which relies on these constraints) contributes meaningfully to `ReScaL`'s performance.

A.3 Justification of the Heuristic Threshold and Potential Over-Trusting of Residuals

We recognize the concern that the dynamic weight α uses a heuristic threshold (0.5) on partial R^2 and defaults to $\alpha = 1$ inside the training range, which could potentially over-trust residuals when the residual model overfits or is poorly supported. However, this design can be grounded in both theoretical and empirical justification. The 0.5 threshold is chosen as a balanced midpoint following established studies (Soares & Araújo, 2015): when $R^2_{\text{partial}} > 0.5$, the residual model explains more than half of the variance that the (N, D)-only model cannot capture, indicating substantial benefit from extra features. When $R^2_{\text{partial}} < 0.5$, the sigmoid naturally down-weights residual correction. Our sensitivity analysis in Section 4.5 further supports this by showing that moderate s values (10–20) lead to robust performance, validating the threshold choice empirically.

When the target $(N^\star, D^\star)$ falls within the training data coverage, we set $\alpha = 1$, thereby fully relying on the residual correction. This can be justified by several factors: (1) within the training range, the residual model operates in an interpolation regime, minimizing the risk of extrapolation errors; (2) the residual correction is applied additively to the scaling law ensemble prediction $\hat{y}^{\text{SL}}$, which itself is robust due to the temperature-controlled softmax ensemble; (3) potential overfitting of the residual model is mitigated by the strong prior imposed by the scaling law ensemble, and further safeguarded by evaluation through 5-fold cross-validation.

When the residual model is poorly supported (e.g., very small pilot sets, limited feature diversity), several mechanisms mitigate risk: automatic down-weighting outside training range via the sigmoid function, RF's inherent regularization (Zhou & Feng, 2017), and low R^2_{partial} naturally yielding small α values. The ablation study shows that removing residual learning degrades performance on feature-rich datasets but has less impact on feature-sparse datasets (e.g., DENSE3 with $|\mathcal{X}| = 3$), suggesting the method adapts appropriately to feature availability and does not over-trust residuals in poorly supported cases.

A.4 Effectiveness of ReScaL for Small Pilot Sets and Preventing Overfitting

It is worth noting that, to mitigate the risk of overfitting with very small pilot sets (e.g., $|\mathcal{S}| = 4$–6 for some datasets at low ρ), we leverage several complementary mechanisms: (1) 5-fold cross-validation to assess each scaling law's performance (Section 3.1), offering a more reliable estimate than training-set error alone; (2) the parameter constraints in diverse fitting ($\pm 1\%$ for fixed, $\pm 35\%$ for adapted parameters) to prevent overfitting to small samples through extreme parameter values; (3) softmax with temperature (Section 3.1) that ensembles predictions across multiple laws, avoiding overcommitment to a single best-fit model and promoting smoother, more stable predictions; and (4) the dynamic weight α (Section 3.3) automatically adapt the influence of residual correction when extrapolating beyond the support of the training data, further reducing the chance of overfitting in underrepresented regions.

While we acknowledge that very small pilot sets ($n < 10$) can have high variance, our RQ1 results in Section 4.3 show consistent advantages even at low ρ values. Further, the ablation study in Section 4.4 confirms that each component contributes meaningfully across varying budget sizes, reinforcing the overall effectiveness of ReScaL.

B Additional Experimental Setup Details

B.1 Pretraining Features and Preprocessing

Table 5 outlines the full set of pretraining features definitions (including the scale variables (N, D)) and per-dataset availability. Specifically, the feature vector $\mathbf{x}_i$ includes hyperparameters, e.g., batch size, and dataset states, e.g., data mixing ratio per domain of training data.

Majority of the features are classified as *numeric*, which can be either integer or float, including both truly continuous values (e.g., learning rate, model size) and discrete integer values with meaningful ordering (e.g., number of layers, number of experts, random seed). A small number of them are categorical, e.g., router type in MoE2, which can be Dense, Hash, RL-R, or S-Base. In this work, we translate categorical features using one-hot encoding, which represents *binary* feature type. For example, the `router_type_*` in MoE2 is represented as four binary features: `router_type_Dense`, `router_type_Hash`, `router_type_RL-R`, and `router_type_S-Base`

All numeric features are standardized using z-score normalization within each dataset. Missing features are handled by imputation using median values for numeric features and mode for categorical features, with a binary indicator flag for imputed values.

B.2 Training and Testing Data Selection

Training Data. Section 4.2 of the main paper details a data partitioning strategy based on the training-cost proxy $C \approx 6ND$ (Kaplan et al., 2020). For a given ratio ρ, the training set consists of the $\lfloor \rho \cdot |\mathcal{S}| \rfloor$ cheapest scale settings by C, sorted in ascending order of compute cost, ensuring that training data always comes from the most cost-efficient scale settings, simulating realistic scenarios where pilot runs are conducted at smaller scales before committing to expensive large-scale training.

Particularly, the training scale settings span from the minimum (N, D) values in each dataset up to progressively larger scales as ρ increases. For datasets with many scale settings (e.g., DENSE1 with $|\mathcal{S}| = 407$, DENSE6 with $|\mathcal{S}| = 907$), we use a fine-grained grid of $\rho \in \{0.1, 0.2, \ldots, 0.9\}$ with step size 0.1 to comprehensively evaluate performance across different pilot budget sizes. For smaller datasets (e.g., DENSE5 with $|\mathcal{S}| = 5$, DENSE4 with $|\mathcal{S}| = 3$), we use coarser grids (e.g., $\{0.2, 0.4, 0.6, 0.8\}$ or $\{0.1, 0.9\}$) to avoid having too few or too many training scale settings at

Table 5: Pretraining feature definitions and per-dataset availability. Features marked with ✓are available in the corresponding dataset.

Feature	Description	Type	DENSE1	DENSE2	DENSE3	DENSE4	DENSE5	DENSE6	MOE1	MOE2
N	Model size (parameters)	Numeric	✓	✓	✓	✓	✓	✓	✓	✓
D	Data size (tokens)	Numeric	✓	✓	✓	✓	✓	✓	✓	✓
h	Hidden dimension	Numeric	✓	✓	–	–	–	–	✓	–
$ffnh$	Feed-forward hidden dimension	Numeric	✓	✓	–	–	–	–	✓	–
$numh$	Number of attention heads	Numeric	✓	✓	–	–	–	–	✓	–
$numl$	Number of layers	Numeric	✓	✓	–	–	–	–	✓	–
lr	Learning rate	Numeric	✓	✓	–	–	–	–	✓	–
bs	Batch size	Numeric	✓	✓	–	–	–	–	✓	–
ti	Pretraining iterations	Numeric	✓	✓	–	–	–	–	✓	–
M	Memory estimate	Numeric	✓	–	–	–	–	–	✓	–
C	Compute estimate	Numeric	✓	–	–	–	–	✓	–	–
N_add_emb	Embedding parameters	Numeric	✓	–	–	–	–	–	–	–
uD	Unique data size	Numeric	–	–	✓	–	–	–	–	–
$step$ / $current_steps$	Training step / Current training steps	Numeric	–	–	–	✓	✓	✓	–	✓
val_github	Validation domain indicator (GitHub validation set)	Binary	–	–	–	✓	–	–	–	–
val_pilecc	Validation domain indicator (Pile-CC validation set)	Binary	–	–	–	✓	–	–	–	–
val_books3	Validation domain indicator (Books3 validation set)	Binary	–	–	–	✓	–	–	–	–
val_prose	Validation domain indicator (Prose validation set)	Binary	–	–	–	–	✓	–	–	–
$val_academic$	Validation domain indicator (Academic validation set)	Binary	–	–	–	–	✓	–	–	–
$val_dialogue$	Validation domain indicator (Dialogue validation set)	Binary	–	–	–	–	✓	–	–	–
$val_symbolic$	Validation domain indicator (Symbolic validation set)	Binary	–	–	–	–	✓	–	–	–
$val_internet$	Validation domain indicator (Internet validation set)	Binary	–	–	–	–	✓	–	–	–
r_github	Data mixing proportion (fraction of GitHub data)	Numeric	–	–	–	✓	–	–	–	–
r_pilecc	Data mixing proportion (fraction of Pile-CC data)	Numeric	–	–	–	✓	–	–	–	–
r_books3	Data mixing proportion (fraction of Books3 data)	Numeric	–	–	–	✓	–	–	–	–
r_prose	Data mixing proportion (fraction of Prose data)	Numeric	–	–	–	–	✓	–	–	–
$r_academic$	Data mixing proportion (fraction of Academic data)	Numeric	–	–	–	–	✓	–	–	–
$r_dialogue$	Data mixing proportion (fraction of Dialogue data)	Numeric	–	–	–	–	✓	–	–	–
$r_symbolic$	Data mixing proportion (fraction of Symbolic data)	Numeric	–	–	–	–	✓	–	–	–
$r_internet$	Data mixing proportion (fraction of Internet data)	Numeric	–	–	–	–	✓	–	–	–
$topk$	Top-k experts selected	Numeric	–	–	–	–	–	–	✓	–
$nume$	Number of experts	Numeric	–	–	–	–	–	–	✓	–
$moeh$	MoE hidden dimension	Numeric	–	–	–	–	–	–	✓	–
sed	Shared expert dimension	Numeric	–	–	–	–	–	–	✓	–
$numld$	Number of layers (dense)	Numeric	–	–	–	–	–	–	✓	–
Na	Additional parameters	Numeric	–	–	–	–	–	–	✓	–
$kvcache$	KV cache size	Numeric	–	–	–	–	–	–	✓	–
$num_experts$	Number of experts	Numeric	–	–	–	–	–	–	–	✓
d_model	Model dimension	Numeric	–	–	–	–	–	–	–	✓
num_blocks	Number of blocks	Numeric	–	–	–	–	–	–	–	✓
num_heads	Number of heads	Numeric	–	–	–	–	–	–	–	✓
kqv_size	Key-query-value size	Numeric	–	–	–	–	–	–	–	✓
$flop_increase$	FLOP increase factor	Numeric	–	–	–	–	–	–	–	✓
k	Top-k routing	Numeric	–	–	–	–	–	–	–	✓
$seed$	Random seed	Numeric	–	–	–	–	–	–	–	✓
$routing_frequency$	Routing frequency	Numeric	–	–	–	–	–	–	–	✓
$dense_parameter_count$	Dense parameter count	Numeric	–	–	–	–	–	–	–	✓
$flops_per_step$	FLOPs per step	Numeric	–	–	–	–	–	–	–	✓
$router_type_DENSE$	Routing algorithm indicator (Dense routing)	Binary	–	–	–	–	–	–	–	✓
$router_type_HASH$	Routing algorithm indicator (Hash routing)	Binary	–	–	–	–	–	–	–	✓
$router_type_RL-R$	Routing algorithm indicator (RL-R routing)	Binary	–	–	–	–	–	–	–	✓
$router_type_S-Base$	Routing algorithm indicator (S-Base routing)	Binary	–	–	–	–	–	–	–	✓
N_no_emb	Parameters without embeddings	Numeric	–	–	–	–	–	✓	–	–
C_no_emb	Compute without embeddings	Numeric	–	–	–	–	–	✓	–	–
C_6ND	Compute estimate ($6ND$)	Numeric	–	–	–	–	–	✓	–	–
$C_6ND_no_emb$	Compute estimate ($6ND$) without embeddings	Numeric	–	–	–	–	–	✓	–	–
$portion_steps_elapsed$	Training progress	Numeric	–	–	–	–	–	✓	–	–
$train_tokens$	Training tokens	Numeric	–	–	–	–	–	✓	–	–
$total_steps$	Total training steps	Numeric	–	–	–	–	–	✓	–	–
$total_epochs$	Total epochs	Numeric	–	–	–	–	–	✓	–	–
$current_epoch$	Current epoch	Numeric	–	–	–	–	–	✓	–	–
$peak_lr$	Peak learning rate	Numeric	–	–	–	–	–	✓	–	–
	Number of features (including N, D)		**12**	**9**	**3**	**9**	**13**	**14**	**17**	**18**

Table 6: Number of training scale settings (S) and actual training samples (N) for each dataset-budget (ρ) pair. Dashes (–) indicate ρ values not tested for that dataset. The training sample counts range from 14 to 30,844 across all dataset-budget (ρ) pairs.

Dataset	$\rho = 0.1$		$\rho = 0.2$		$\rho = 0.3$		$\rho = 0.4$		$\rho = 0.5$		$\rho = 0.6$		$\rho = 0.7$		$\rho = 0.8$		$\rho = 0.9$	
	S	**N**	**S**	**N**	**S**	**N**	**S**	**N**	**S**	**N**	**S**	**N**	**S**	**N**	**S**	**N**	**S**	**N**
DENSE1	40	40	81	81	122	122	162	162	203	203	244	244	284	285	325	326	366	369
DENSE2	1	119	–	–	5	596	–	–	8	939	–	–	11	1,274	–	–	14	1,625
DENSE3	13	14	27	29	41	43	54	62	68	91	82	106	94	141	108	165	122	234
DENSE4	1	1,096	–	–	–	–	–	–	–	–	–	–	–	–	2	3,343	–	–
DENSE5	–	–	1	8,932	–	–	2	16,544	–	–	3	24,552	–	–	4	30,844	–	–
DENSE6	90	1,033	181	1,743	272	2,363	362	2,846	453	3,220	544	3,555	634	3,873	725	4,214	816	4,529
MOE1	1	90	2	135	3	180	4	270	6	359	7	449	8	489	9	533	10	623
MOE2	13	546	–	–	39	1,742	–	–	65	2,886	–	–	91	4,056	–	–	117	5,198

extreme ρ values. After selecting the training scale settings, we exclude: (1) the point with maximum N in the N-max group, maximum D in that group, and (2) the point with maximum D in the D-max group, maximum N in that group from the training set, as these points are reserved for the testing set to ensure evaluation focuses on extrapolation to large-scale targets.

Table 6 summarizes the counts of training scale settings and actual training samples for each dataset-budget pair. The training sample counts range from 14 (for DENSE3 at $\rho = 0.1$) to 30,844 (for

DENSE5 at $\rho = 0.8$), reflecting the diversity in dataset sizes and the number of pretraining runs per scale setting across different datasets.

The training data covers the lower end of the scale variable ranges for each dataset. For example, at $\rho = 0.1$, training data for DENSE1 covers approximately the bottom 10% of the $N \times D$ range (from $N \approx 2.0{\times}10^8$, $D \approx 1.0{\times}10^9$ up to intermediate values), while at $\rho = 0.9$, it covers most of the available scale settings except the most expensive ones reserved for testing. This design allows us to evaluate how well `ReScaL` extrapolates from small-scale pilots (low ρ) to large-scale targets (testing set), which is the primary use case for validation loss prediction in LLM pretraining.

Testing Data. As outlined in Section 4.2 of the main paper, for testing data, we select the pretraining runs from the **most expensive** scale settings on the **Pareto front** of all scale settings, i.e., the non-dominated points where no other setting has both larger N and larger D. This allows us to focus our evaluation on challenging extrapolation scenarios: predicting performance on large-scale targets from a range of small-scale pilot runs.

Table 7 provides the number of testing scale settings and testing samples for each dataset. The number of test scale settings ranges from 1 to 5, and the number of test samples ranges from 5 to 2,247 across datasets. Notably, most datasets end up with only 1 testing scale setting, as the Pareto front naturally selects the most extreme non-dominated points. However, these testing settings represent the most challenging and practically relevant targets for validation loss prediction, as they correspond to the largest-scale pretraining runs that would be conducted in practice.

Figure 5 visualizes the scale settings distribution for all eight datasets. In each subfigure, potential training data (blue circles) are concentrated in the lower-left region of the (N, D) space, representing the cheaper, smaller-scale settings used for training. Testing data (red stars) are distributed at the edges and upper-right regions, representing the expensive, large-scale settings used for evaluation. For example, in Figure 5c, the training data forms dense clusters in the lower N and D ranges, while the testing data consists of 4 scale settings located at the extremes of the distribution, including points with maximum N or maximum D values. This clear separation ensures that the evaluation focuses on true extrapolation rather than interpolation.

B.3 THE SELECTION OF COMPARED MACHINE LEARNING MODELS IN EVALUATION

Our baseline selection prioritizes methods that are most relevant to the LLM scaling law prediction task and represent distinct modeling approaches. We compare against: (1) individual scaling laws (the domain-specific priors), (2) `RF` as a strong non-parametric baseline, (3) NCPL as a configuration-aware neural scaling law, and (4) expert-designed functions (`HUMAN`) where available. `RF` serves as a representative tree-based ensemble method; while gradient boosting methods (`XGBoost`/`LightGBM`) are powerful alternatives, our central comparison asks whether residual augmentation improves upon the same `RF` used alone. We do not claim that `RF` is the uniquely optimal residual estimator.

Gaussian processes (`GP`s) and Bayesian neural networks (`BNN`s) are valuable for uncertainty quantification, but our focus is on point prediction accuracy under limited pilot budgets. Moreover, `GP`s scale poorly with dataset size (our training sets range from 14 to 30,844 samples) and require careful kernel design for extrapolation (Wilson & Adams, 2013), while `BNN`s are computationally expensive and may not offer clear advantages over `RF` for residual modeling in this setting (Wenzel et al., 2020; Snoek et al., 2019). The dynamic weight α in `ReScaL` provides a form of uncertainty-aware gating by reducing residual correction when the scaling law ensemble is reliable, addressing a similar concern without the computational overhead.

C ADDITIONAL EXPERIMENT RESULTS

C.1 DYNAMIC AND FIXED RESIDUAL WEIGHTS

Table 8 replaces Equation 12 with one globally fixed coefficient while leaving all other components unchanged. The dynamic gate achieves the lowest average across systems (0.0746), compared with 0.0838 for the strongest globally fixed choice, $\alpha = 0.8$. Individual systems can favor different post-hoc fixed values—for example, DENSE4 favors $\alpha = 0.8$—but choosing these values requires target

Table 7: Details of testing data for each dataset.

Dataset	# Testing Scale Settings	# Testing Samples
DENSE1	5	5
DENSE2	2	167
DENSE3	4	8
DENSE4	1	2,247
DENSE5	1	418
DENSE6	1	5
MOE1	1	59
MOE2	1	44

Table 8: Mean MAPE for the data-derived gate and fixed residual weights. Bold marks the dynamic result when it is no worse than all fixed settings after rounding.

System	Dynamic	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
DENSE1	**0.022**	0.022	0.022	0.022	0.023	0.023	0.024	0.025	0.026	0.027	0.028	0.029
DENSE2	0.047	0.066	0.061	0.058	0.055	0.054	0.052	0.050	0.049	0.048	0.047	0.046
DENSE3	**0.076**	0.078	0.084	0.093	0.102	0.111	0.120	0.129	0.139	0.149	0.159	0.169
DENSE4	0.173	0.466	0.419	0.372	0.325	0.280	0.234	0.189	0.147	0.136	0.140	0.173
DENSE5	0.167	0.454	0.396	0.341	0.290	0.243	0.201	0.172	0.161	0.161	0.169	0.197
DENSE6	0.037	0.037	0.033	0.034	0.036	0.040	0.043	0.047	0.051	0.055	0.059	0.064
MOE1	**0.018**	0.026	0.024	0.023	0.022	0.021	0.021	0.021	0.022	0.023	0.025	0.026
MOE2	**0.057**	0.136	0.128	0.120	0.112	0.104	0.096	0.088	0.080	0.072	0.064	0.057
Mean	**0.0746**	0.1604	0.1459	0.1327	0.1206	0.1093	0.0989	0.0902	0.0842	0.0838	0.0864	0.0951

losses. The learned gate instead uses only the pilot set. Realized extrapolation weights also reflect strong system differences: they remain near zero for DENSE1 and DENSE6, near one for DENSE4 and MOE2, and vary with the pilot budget for MOE1. This confirms that α is a pilot-set-level gate rather than a continuously target-specific coefficient.

C.2 RQ1 RESULTS WITH ADDITIONAL METRICS

We additionally showcase the results for **RQ1** using two other error metrics, mean absolute error (MAE) and root mean square error (RMSE) in Tables 9 and 10, respectively. As can be seen, the conclusion remain consistent with the that of MAPE: `ReScaL` achieves the best overall rank with considerable improvements over the others.

C.3 COMPLETE DECISION-ORIENTED EVALUATION

For every dataset–budget case, we compute each decision-oriented metric over the candidate runs in its held-out target scale settings. Spearman correlations that are undefined because a method predicts a constant value are counted as zero. PairAcc assigns half credit to predicted ties and excludes pairs whose observed losses are tied. Tables 11–15 report the raw case-level results for Sp, PairAcc, Top1, Reg, and NReg, with one table per metric, for every available method at every pilot budget. MAPE results are already reported in the main paper and are therefore not repeated here. For the aggregate summary in Table 3, Scott–Knott ranks are first computed within each dataset and metric over its budget-level results and then averaged over the covered datasets. The expert-designed baseline is available only on DENSE3–5 (15 cases), so its values are not directly comparable to methods evaluated on all 52 cases.

Table 9: RQ1 results using MAE. The formate follows Table 2.

Dataset	ReScaL		Farseer		Farseer$_d$		Chinchilla		Chinchilla$_d$		Kaplan		Kaplan$_d$		RF		RF$_{ND}$		HUMAN	
	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)
DENSE1																				
$\rho = 0.1$	**1**	**0.0383 (0.0303)**	5	0.2776 (0.2436)	9	1.3984 (0.0748)	3	0.2235 (0.0662)	6	0.3228 (0.033)	2	0.1655 (0.0675)	4	0.2473 (0.0864)	8	0.811 (0.1079)	7	0.7507 (0.1039)		N/A
$\rho = 0.2$	**1**	**0.0248 (0.009)**	2	0.1365 (0.1857)	7	1.3984 (0.0748)	3	0.2007 (0.0592)	5	0.3228 (0.033)	2	0.1501 (0.064)	4	0.2473 (0.0864)	6	0.6688 (0.1044)	6	0.6585 (0.1039)		N/A
$\rho = 0.3$	**1**	**0.0277 (0.0098)**	2	0.1155 (0.1736)	8	1.3984 (0.0748)	3	0.1719 (0.0537)	5	0.3228 (0.033)	2	0.1369 (0.0606)	4	0.2473 (0.0864)	7	0.589 (0.1046)	6	0.568 (0.1023)		N/A
$\rho = 0.4$	**1**	**0.0292 (0.022)**	2	0.1118 (0.168)	8	1.3984 (0.0748)	3	0.1606 (0.0528)	5	0.3228 (0.033)	2	0.1209 (0.0548)	4	0.2473 (0.0864)	7	0.503 (0.1133)	6	0.4521 (0.1038)		N/A
$\rho = 0.5$	**1**	**0.0346 (0.0276)**	2	0.1065 (0.1582)	8	1.3984 (0.0748)	3	0.1503 (0.0511)	5	0.3228 (0.033)	2	0.1112 (0.0518)	4	0.2473 (0.0864)	7	0.4486 (0.1032)	6	0.4184 (0.104)		N/A
$\rho = 0.6$	**1**	**0.0427 (0.0271)**	2	0.1087 (0.1535)	8	1.3984 (0.0748)	3	0.1426 (0.0514)	5	0.3228 (0.033)	2	0.1005 (0.0478)	4	0.2473 (0.0864)	7	0.3899 (0.1068)	6	0.3513 (0.0991)		N/A
$\rho = 0.7$	**1**	**0.0457 (0.0271)**	2	0.1091 (0.15)	7	1.3984 (0.0748)	3	0.1302 (0.0498)	6	0.3228 (0.033)	2	0.0928 (0.0447)	4	0.2473 (0.0864)	6	0.3186 (0.1054)	5	0.2942 (0.0958)		N/A
$\rho = 0.8$	**1**	**0.0443 (0.027)**	3	0.1094 (0.1501)	7	1.3984 (0.0748)	3	0.1104 (0.0456)	6	0.3228 (0.033)	2	0.0842 (0.0413)	5	0.2473 (0.0864)	5	0.2352 (0.0846)	4	0.2155 (0.0769)		N/A
$\rho = 0.9$	**1**	**0.0409 (0.0271)**	4	0.1087 (0.1508)	8	1.3984 (0.0748)	3	0.0843 (0.0438)	7	0.3228 (0.033)	2	0.0758 (0.0379)	6	0.2473 (0.0864)	5	0.1672 (0.0788)	5	0.1559 (0.0722)		N/A
DENSE2																				
$\rho = 0.1$	2	0.2533 (0.4409)	4	0.6217 (0.3331)	7	1.779 (0.0968)	3	0.4653 (0.0457)	2	0.2536 (0.0497)	5	0.9511 (0.1175)	**1**	**0.2104 (0.0482)**	3	0.4749 (0.4149)	6	1.141 (0.118)		N/A
$\rho = 0.3$	**1**	**0.1075 (0.0631)**	2	0.1709 (0.0452)	8	1.779 (0.0968)	5	0.3373 (0.0456)	4	0.2536 (0.0497)	6	0.4927 (0.0451)	3	0.2104 (0.0482)	2	0.1692 (0.1045)	7	0.7637 (0.1047)		N/A
$\rho = 0.5$	**1**	**0.0642 (0.0369)**	4	0.2542 (0.1408)	8	1.779 (0.0968)	5	0.3794 (0.126)	4	0.2536 (0.0497)	6	0.4381 (0.0486)	3	0.2104 (0.0482)	2	0.1489 (0.0876)	7	0.598 (0.144)		N/A
$\rho = 0.7$	**1**	**0.0858 (0.0996)**	2	0.1208 (0.0957)	8	1.779 (0.0968)	4	0.2257 (0.0636)	5	0.2536 (0.0497)	7	0.3287 (0.0626)	3	0.2104 (0.0482)	2	0.114 (0.1808)	6	0.2998 (0.0661)		N/A
$\rho = 0.9$	**1**	**0.0563 (0.0344)**	4	0.1481 (0.1245)	8	1.779 (0.0968)	3	0.1113 (0.1099)	7	0.2536 (0.0497)	5	0.1732 (0.0967)	6	0.2104 (0.0482)	**1**	**0.0591 (0.051)**	2	0.0859 (0.0698)		N/A
DENSE3																				
$\rho = 0.1$	**1**	**0.2527 (0.0885)**	4	0.8888 (0.4161)	6	2.182 (0.5277)	**1**	**0.2512 (0.2107)**	2	0.3515 (0.2519)	2	0.3154 (0.1484)	3	0.4711 (0.2298)	7	2.7298 (0.5439)	7	2.7184 (0.5205)	5	1.1215 (0.3743)
$\rho = 0.2$	**1**	**0.1547 (0.2246)**	5	0.8574 (0.7377)	7	2.182 (0.5277)	3	0.4099 (0.3378)	2	0.3515 (0.2519)	2	0.3154 (0.1484)	4	0.4711 (0.2298)	6	1.3848 (0.4755)	6	1.3461 (0.4595)	5	0.8114 (0.1878)
$\rho = 0.3$	**1**	**0.1356 (0.2105)**	5	0.4708 (0.8128)	8	2.182 (0.5277)	4	0.414 (0.289)	3	0.3515 (0.2519)	2	0.2961 (0.1431)	5	0.4711 (0.2298)	7	1.2299 (0.4179)	7	1.1617 (0.4023)	6	0.6207 (0.0831)
$\rho = 0.4$	**1**	**0.1238 (0.1866)**	5	0.468 (0.734)	7	2.182 (0.5277)	4	0.4018 (0.2322)	3	0.3515 (0.2519)	2	0.212 (0.1205)	5	0.4711 (0.2298)	6	0.9066 (0.3447)	6	0.974 (0.3644)	5	0.5178 (0.0714)
$\rho = 0.5$	**1**	**0.1455 (0.1165)**	3	0.5137 (0.706)	6	2.182 (0.5277)	**1**	**0.1705 (0.2457)**	2	0.3515 (0.2519)	**1**	**0.1626 (0.1078)**	3	0.4711 (0.2298)	4	0.9217 (0.3597)	5	1.6367 (0.5655)	3	0.4675(0.0899)
$\rho = 0.6$	2	0.2323 (0.0816)	4	0.4508 (0.7162)	6	2.182 (0.5277)	3	0.3171 (0.1167)	3	0.3515 (0.2519)	**1**	**0.2076 (0.0974)**	4	0.4711 (0.2298)	5	1.3143 (0.5589)	7	3.1818 (1.1852)	4	0.4622 (0.0921)
$\rho = 0.7$	**1**	**0.2518 (0.1296)**	**1**	**0.2987 (0.8038)**	6	2.182 (0.5277)	3	0.4242 (0.1892)	2	0.3515 (0.2519)	**1**	**0.2293 (0.1065)**	4	0.4711 (0.2298)	4	0.4625 (0.254)	5	1.2269 (0.4235)	2	0.3855 (0.0506)
$\rho = 0.8$	**1**	**0.2592 (0.1358)**	2	0.3293 (0.8055)	5	2.182 (0.5277)	3	0.4448 (0.2015)	2	0.3515 (0.2519)	**1**	**0.2511 (0.1164)**	3	0.4711 (0.2298)	**1**	**0.2349 (0.1209)**	4	0.5421 (0.2207)	2	0.3603 (0.0441)
$\rho = 0.9$	2	0.2702 (0.1437)	4	0.3893 (0.7995)	6	2.182 (0.5277)	4	0.4484 (0.2036)	3	0.3515 (0.2519)	2	0.2637 (0.1224)	4	0.4711 (0.2298)	**1**	**0.2111 (0.1085)**	5	0.5867 (0.3293)	2	0.257 (0.1137)
DENSE4																				
$\rho = 0.1$	2	0.5853 (0.1626)	4	1.4351 (0.9095)	5	1.8946 (0.9095)	3	0.7548 (0.5807)	3	0.776 (0.5116)	3	0.7547 (0.5809)	3	0.8123 (0.4058)	**1**	**0.4789 (0.1778)**	3	0.7493 (0.8828)	3	0.8218 (0.7447)
$\rho = 0.9$	**1**	**0.1417 (0.1101)**	3	0.6941 (0.8043)	4	1.8946 (0.9095)	3	0.7209 (0.697)	3	0.776 (0.5116)	3	0.7253 (0.6815)	3	0.8123 (0.4058)	2	0.2097 (0.1083)	3	0.6941 (0.8113)	3	0.7843 (0.7432)
DENSE5																				
$\rho = 0.2$	2	0.5888 (0.3228)	6	2.0568 (0.5592)	5	1.9188 (0.5592)	**1**	**0.485 (0.4331)**	**1**	**0.4573 (0.3251)**	**1**	**0.4825 (0.4293)**	**1**	**0.4753 (0.3055)**	3	0.6809 (0.3424)	4	1.0724 (0.5592)	4	1.0045 (0.5618)
$\rho = 0.4$	**1**	**0.3114 (0.191)**	3	0.4919 (0.4425)	6	1.9188 (0.5592)	3	0.4618 (0.3827)	3	0.4573 (0.3251)	3	0.4727 (0.4104)	3	0.4753 (0.3055)	2	0.3977 (0.2934)	4	0.7524 (0.545)	5	0.8661 (0.5556)
$\rho = 0.6$	**1**	**0.3466 (0.2165)**	2	0.5001 (0.4518)	5	1.9188 (0.5592)	2	0.4865 (0.4354)	2	0.4573 (0.3251)	2	0.4795 (0.4243)	2	0.4753 (0.3055)	**1**	**0.3589 (0.2773)**	3	0.6677 (0.5287)	4	0.7873 0.5484
$\rho = 0.8$	**1**	**0.2831 (0.2046)**	3	0.4572 (0.3253)	5	1.9188 (0.5592)	3	0.4704 (0.4052)	3	0.4573 (0.3251)	3	0.4723 (0.4097)	3	0.4753 (0.3055)	2	0.3307 (0.2638)	3	0.5308 (0.4756)	4	0.7198 (0.5392)
DENSE6																				
$\rho = 0.1$	2	0.1362 (0.1588)	4	0.2981 (0.1588)	8	2.2649 (0.1588)	4	0.2985 (0.1588)	3	0.2329 (0.1588)	**1**	**0.0982 (0.1357)**	5	0.4386 (0.1588)	7	1.2925 (0.3713)	6	1.1146 (0.1588)		N/A
$\rho = 0.2$	**1**	**0.1296 (0.1588)**	2	0.1814 (0.1531)	8	2.2649 (0.1588)	4	0.3147 (0.1588)	3	0.2329 (0.1588)	**1**	**0.0996 (0.1559)**	5	0.4386 (0.1588)	7	1.0156 (0.2698)	6	0.8645 (0.1588)		N/A
$\rho = 0.3$	**1**	**0.1162 (0.1588)**	2	0.1587 (0.075)	8	2.2649 (0.1588)	4	0.3149 (0.1588)	3	0.2329 (0.1588)	**1**	**0.1073 (0.104)**	5	0.4386 (0.1588)	7	0.8211 (0.2768)	6	0.6729 (0.1588)		N/A
$\rho = 0.4$	**1**	**0.1144 (0.1588)**	2	0.1449 (0.0283)	8	2.2649 (0.1588)	4	0.3108 (0.1588)	3	0.2329 (0.1588)	**1**	**0.1217 (0.0546)**	5	0.4386 (0.1588)	7	0.6717 (0.279)	6	0.549 (0.1588)		N/A
$\rho = 0.5$	**1**	**0.0997 (0.1561)**	3	0.138 (0.0101)	9	2.2649 (0.1588)	5	0.3066 (0.1588)	4	0.2329 (0.1588)	2	0.1258 (0.0407)	6	0.4386 (0.1588)	8	0.8664 (0.0485)	7	0.5198 (0.1588)		N/A
$\rho = 0.6$	**1**	**0.0968 (0.151)**	2	0.131 (0.0237)	7	2.2649 (0.1588)	4	0.3024 (0.1588)	3	0.2329 (0.1588)	2	0.1283 (0.0325)	5	0.4386 (0.1588)	6	0.7852 (0.0643)	5	0.4466 (0.1588)		N/A
$\rho = 0.7$	**1**	**0.0962 (0.1424)**	2	0.1254 (0.0422)	8	2.2649 (0.1588)	4	0.274 (0.1588)	3	0.2329 (0.1588)	2	0.1291 (0.0299)	6	0.4386 (0.1588)	7	0.8613 (0.1306)	5	0.397 (0.1588)		N/A
$\rho = 0.8$	**1**	**0.0961 (0.1428)**	2	0.1175 (0.0692)	7	2.2649 (0.1588)	3	0.2541 (0.1588)	3	0.2329 (0.1588)	2	0.1282 (0.0327)	5	0.4386 (0.1588)	6	0.6201 (0.0538)	4	0.3622 (0.1588)		N/A
$\rho = 0.9$	**1**	**0.0957 (0.1434)**	6	0.9536 (0.1588)	7	2.2649 (0.1588)	3	0.2272 (0.1588)	3	0.2329 (0.1588)	2	0.1264 (0.0388)	4	0.4386 (0.1588)	5	0.8394 (0.6202)	3	0.252 (0.1588)		N/A
MOE1																				
$\rho = 0.1$	**1**	**0.038 (0.026)**	6	0.3421 (0.0355)	9	1.6871 (0.0355)	4	0.1233 (0.0355)	5	0.3048 (0.0355)	3	0.1151 (0.0355)	2	0.0789 (0.0355)	7	0.4834 (0.0468)	8	0.5077 (0.0355)		N/A
$\rho = 0.2$	**1**	**0.0442 (0.027)**	5	0.3528 (0.0355)	8	1.6871 (0.0355)	3	0.113 (0.0355)	4	0.3048 (0.0355)	7	0.4851 (0.0355)	2	0.0789 (0.0355)	6	0.4517 (0.0489)	6	0.4567 (0.0355)		N/A
$\rho = 0.3$	**1**	**0.0431 (0.0275)**	4	0.3612 (0.0355)	8	1.6871 (0.0355)	5	0.4152 (0.0355)	3	0.3048 (0.0355)	7	0.4733 (0.0355)	2	0.0789 (0.0355)	6	0.4318 (0.0506)	6	0.4394 (0.0355)		N/A
$\rho = 0.4$	**1**	**0.0823 (0.0448)**	2	0.1254 (0.0355)	8	1.6871 (0.0355)	3	0.1872 (0.0355)	5	0.3048 (0.0355)	4	0.2359 (0.0355)	**1**	**0.0789 (0.0355)**	6	0.3314 (0.0339)	7	0.3672 (0.0355)		N/A
$\rho = 0.5$	**1**	**0.0423 (0.0292)**	**1**	**0.0476 (0.0319)**	8	1.6871 (0.0355)	3	0.1038 (0.0355)	7	0.3048 (0.0355)	4	0.1368 (0.0355)	2	0.0789 (0.0355)	5	0.2691 (0.0418)	6	0.2897 (0.0355)		N/A
$\rho = 0.6$	**1**	**0.0287 (0.0205)**	3	0.1179 (0.0355)	9	1.6871 (0.0355)	4	0.1547 (0.0355)	8	0.3048 (0.0355)	5	0.1889 (0.0355)	2	0.0789 (0.0355)	6	0.2501 (0.0464)	7	0.2693 (0.0355)		N/A
$\rho = 0.7$	**1**	**0.0312 (0.0208)**	3	0.097 (0.0355)	8	1.6871 (0.0355)	4	0.1383 (0.0355)	7	0.3048 (0.0355)	5	0.1695 (0.0355)	2	0.0789 (0.0355)	6	0.2104 (0.0432)	6	0.2093 (0.0355)		N/A
$\rho = 0.8$	**1**	**0.03 (0.0205)**	2	0.0763 (0.0355)	7	1.6871 (0.0355)	3	0.1175 (0.0355)	6	0.3048 (0.0355)	4	0.1454 (0.0355)	2	0.0789 (0.0355)	5	0.1783 (0.0389)	5	0.1781 (0.0355)		N/A
$\rho = 0.9$	**1**	**0.0285 (0.0207)**	3	0.089 (0.0355)	7	1.6871 (0.0355)	4	0.1051 (0.0355)	6	0.3048 (0.0355)	5	0.1145 (0.0355)	2	0.0789 (0.0355)	2	0.0748 (0.0147)	4	0.0991 (0.0355)		N/A
MOE2																				
$\rho = 0.1$	**1**	**0.1119 (0.1521)**	6	1.8801 (1.6944)	6	2.1707 (1.6944)	4	0.9839 (1.4181)	2	0.4736 (1.6923)	2	0.4464 (1.6849)	3	0.7994 (1.6944)	3	0.7195 (0.1416)	5	1.2737 (1.2956)		N/A
$\rho = 0.3$	**1**	**0.0778 (0.1146)**	5	1.6723 (1.6944)	6	2.1707 (1.6944)	3	0.9521 (1.4316)	2	0.4736 (1.6923)	2	0.4451 (1.6842)	3	0.7994 (1.6944)	2	0.4573 (0.0974)	4	1.2232 (1.3169)		N/A
$\rho = 0.5$	**1**	**0.2232 (0.0584)**	5	1.38 (1.6944)	6	2.1707 (1.6944)	4	0.8882 (1.4586)	3	0.4736 (1.6923)	3	0.4441 (1.6836)	4	0.7994 (1.6944)	2	0.2922 (0.0642)	5	1.1398 (1.3522)		N/A
$\rho = 0.7$	2	0.234 (0.0533)	5	1.5878 (1.6944)	6	2.1707 (1.6944)	4	0.8417 (1.4783)	3	0.4736 (1.6923)	3	0.4431 (1.6831)	4	0.7994 (1.6944)	**1**	**0.1261 (0.0338)**	4	0.8171 (1.4887)		N/A
$\rho = 0.9$	2	0.0906 (0.0381)	4	0.8338 (1.6944)	5	2.1707 (1.6944)	4	0.7905 (1.4999)	3	0.4736 (1.6923)	3	0.4424 (1.6826)	4	0.7994 (1.6944)	**1**	**0.0539 (0.0154)**	4	0.7702 (1.5085)		N/A
Average r	**1.15**		3.33		7.06		3.37		3.87		2.83		3.58		4.58		5.31			

Table 10: RQ1 results using RMSE. The formate follows Table 2.

Dataset	ReScaL		Farseer		Farseer$_d$		Chinchilla		Chinchilla$_{ad}$		Kaplan		Kaplan$_d$		RF		RF$_{ND}$		HUMAN
	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)
Dense1																			
$\rho = 0.1$	**1**	**0.0022 (0.0028)**	5	0.1245 (0.182)	8	1.9601 (0.2166)	3	0.0535 (0.0244)	5	0.1051 (0.0202)	2	0.031 (0.0188)	4	0.0671 (0.0378)	7	0.6671 (0.1606)	6	0.5721 (0.1422)		N/A
$\rho = 0.2$	**1**	**0.0007 (0.0005)**	3	0.0462 (0.093)	7	1.9601 (0.2166)	3	0.0431 (0.0194)	5	0.1051 (0.0202)	2	0.0258 (0.0158)	4	0.0671 (0.0378)	6	0.4561 (0.126)	6	0.4423 (0.1231)		N/A
$\rho = 0.3$	**1**	**0.0008 (0.0005)**	3	0.0374 (0.0794)	8	1.9601 (0.2166)	3	0.0319 (0.0148)	5	0.1051 (0.0202)	2	0.0217 (0.0134)	4	0.0671 (0.0378)	7	0.3557 (0.1093)	6	0.331 (0.1029)		N/A
$\rho = 0.4$	**1**	**0.0012 (0.0011)**	3	0.0351 (0.0748)	8	1.9601 (0.2166)	3	0.028 (0.0135)	5	0.1051 (0.0202)	2	0.017 (0.0105)	4	0.0671 (0.0378)	7	0.2633 (0.0977)	6	0.213 (0.0801)		N/A
$\rho = 0.5$	**1**	**0.0018 (0.0016)**	3	0.0314 (0.0665)	8	1.9601 (0.2166)	3	0.0247 (0.0121)	5	0.1051 (0.0202)	2	0.0145 (0.009)	4	0.0671 (0.0378)	7	0.2097 (0.0792)	6	0.1837 (0.0733)		N/A
$\rho = 0.6$	**1**	**0.0024 (0.002)**	3	0.0307 (0.064)	8	1.9601 (0.2166)	3	0.0224 (0.0114)	5	0.1051 (0.0202)	2	0.0119 (0.0073)	4	0.0671 (0.0378)	7	0.1611 (0.0694)	6	0.1313 (0.0573)		N/A
$\rho = 0.7$	**1**	**0.0027 (0.0022)**	4	0.0299 (0.0616)	8	1.9601 (0.2166)	3	0.0189 (0.0099)	7	0.1051 (0.0202)	2	0.0102 (0.0063)	5	0.0671 (0.0378)	7	0.1104 (0.0535)	6	0.0939 (0.0449)		N/A
$\rho = 0.8$	**1**	**0.0025 (0.0021)**	4	0.03 (0.0618)	8	1.9601 (0.2166)	3	0.0139 (0.0074)	7	0.1051 (0.0202)	2	0.0084 (0.0052)	6	0.0671 (0.0378)	6	0.0611 (0.0312)	5	0.0512 (0.0261)		N/A
$\rho = 0.9$	**1**	**0.0023 (0.0019)**	4	0.03 (0.0621)	8	1.9601 (0.2166)	3	0.0086 (0.005)	7	0.1051 (0.0202)	2	0.0069 (0.0042)	6	0.0671 (0.0378)	5	0.0329 (0.0187)	4	0.0285 (0.0164)		N/A
Dense2																			
$\rho = 0.1$	3	0.2574 (1.0606)	4	0.4967 (0.313)	7	3.1742 (0.3405)	3	0.2186 (0.0401)	2	0.0668 (0.0231)	5	0.9183 (0.2133)	**1**	**0.0466 (0.0188)**	4	0.3967 (1.0542)	6	1.3158 (0.2783)		N/A
$\rho = 0.3$	**1**	**0.0155 (0.0162)**	2	0.0312 (0.0133)	9	3.1742 (0.3405)	6	0.1158 (0.0284)	5	0.0668 (0.0231)	7	0.2448 (0.0424)	4	0.0466 (0.0188)	3	0.0395 (0.0424)	8	0.5941 (0.1661)		N/A
$\rho = 0.5$	**1**	**0.0055 (0.0052)**	5	0.0843 (0.0873)	9	3.1742 (0.3405)	6	0.1598 (0.107)	4	0.0668 (0.0231)	7	0.1943 (0.0402)	3	0.0466 (0.0188)	2	0.0298 (0.0316)	8	0.3782 (0.1883)		N/A
$\rho = 0.7$	**1**	**0.0172 (0.088)**	**1**	**0.0237 (0.0307)**	5	3.1742 (0.3405)	2	0.055 (0.0282)	2	0.0668 (0.0231)	4	0.1119 (0.0404)	2	0.0466 (0.0188)	2	0.0455 (0.3029)	3	0.0942 (0.0393)		N/A
$\rho = 0.9$	**1**	**0.0043 (0.005)**	5	0.0373 (0.0494)	8	3.1742 (0.3405)	4	0.0244 (0.0359)	7	0.0668 (0.0231)	5	0.0393 (0.04)	6	0.0466 (0.0188)	2	0.0061 (0.0086)	3	0.0122 (0.015)		N/A
Dense3																			
$\rho = 0.1$	**1**	**0.071 (0.0587)**	5	0.9474 (0.8763)	7	5.0141 (2.8967)	2	0.1035 (0.1501)	3	0.1812 (0.2629)	2	0.1195 (0.1275)	4	0.2699 (0.326)	8	7.7207 (2.3932)	8	7.6357 (2.3165)	6	1.3851 (1.019)
$\rho = 0.2$	**1**	**0.0698 (0.1565)**	6	1.2298 (2.5926)	8	5.0141 (2.8967)	4	0.2718 (0.4456)	3	0.1812 (0.2629)	2	0.1195 (0.1275)	4	0.2699 (0.326)	7	2.1232 (0.9286)	7	2.0039 (0.8845)	5	0.6903 (0.332)
$\rho = 0.3$	**1**	**0.0587 (0.1232)**	6	0.8223 (2.5331)	9	5.0141 (2.8967)	4	0.2473 (0.3482)	3	0.1812 (0.2629)	2	0.1063 (0.1155)	4	0.2699 (0.326)	8	1.6713 (0.7609)	7	1.4967 (0.6971)	5	0.3915 (0.1028)
$\rho = 0.4$	**1**	**0.047 (0.0924)**	4	0.7088 (2.1126)	7	5.0141 (2.8967)	2	0.2105 (0.2444)	2	0.1812 (0.2629)	**1**	**0.0582 (0.0695)**	3	0.2699 (0.326)	5	0.9299 (0.4816)	6	1.0694 (0.5367)	3	0.2727 (0.0721)
$\rho = 0.5$	**1**	**0.0335 (0.0567)**	5	0.7169 (2.0209)	7	5.0141 (2.8967)	2	0.084 (0.1552)	3	0.1812 (0.2629)	**1**	**0.037 (0.0473)**	4	0.2699 (0.326)	5	0.9671 (0.4966)	6	2.9694 (1.2317)	4	0.2295 (0.0733)
$\rho = 0.6$	2	0.06 (0.0303)	7	0.6695 (2.0051)	9	5.0141 (2.8967)	3	0.1129 (0.0675)	4	0.1812 (0.2629)	**1**	**0.0517 (0.0485)**	6	0.2699 (0.326)	8	2.0114 (1.0526)	10	11.401 (4.9556)	5	0.2213 (0.0732)
$\rho = 0.7$	2	0.0787 (0.0504)	6	0.6765 (2.2228)	8	5.0141 (2.8967)	4	0.2125 (0.108)	4	0.1812 (0.2629)	**1**	**0.0629 (0.0606)**	5	0.2699 (0.326)	5	0.2725 (0.173)	7	1.6683 (0.6663)	3	0.151 (0.038)
$\rho = 0.8$	2	0.0839 (0.0531)	7	0.6982 (2.2847)	8	5.0141 (2.8967)	5	0.2348 (0.1192)	4	0.1812 (0.2629)	**1**	**0.0754 (0.0747)**	5	0.2699 (0.326)	**1**	**0.0685 (0.0442)**	6	0.3381 (0.1757)	3	0.1316 (0.0311)
$\rho = 0.9$	3	0.0918 (0.0577)	6	0.7327 (2.3529)	7	5.0141 (2.8967)	5	0.2387 (0.1207)	4	0.1812 (0.2629)	2	0.0832 (0.0838)	5	0.2699 (0.326)	**1**	**0.0553 (0.0359)**	6	0.4428 (0.2837)	2	0.078 (0.0795)
Dense4																			
$\rho = 0.1$	2	0.3687 (0.2375)	5	2.8788 (3.1994)	6	4.4089 (2.89)	3	0.9036 (1.1457)	3	0.8614 (0.998)	3	0.9037 (1.1461)	3	0.8229 (0.7527)	**1**	**0.2607 (0.2648)**	4	1.3333 (1.9038)	4	1.2246 (1.6155)
$\rho = 0.9$	**1**	**0.0321 (0.0931)**	4	1.1225 (1.6062)	5	4.4089 (2.89)	3	1.0007 (1.3846)	3	0.8614 (0.998)	3	0.986 (1.3534)	3	0.8229 (0.7527)	2	0.0556 (0.0795)	4	1.1336 (1.6241)	4	1.1621 (1.5056)
Dense5																			
$\rho = 0.2$	2	0.4507 (0.441)	6	4.5424 (2.5263)	5	3.9935 (1.979)	2	0.4224 (0.6768)	**1**	**0.3145 (0.4441)**	2	0.4167 (0.6686)	**1**	**0.319 (0.3773)**	3	0.5805 (0.5156)	4	1.462 (1.4455)	4	1.3239 (1.3765)
$\rho = 0.4$	**1**	**0.1334 (0.1578)**	4	0.4374 (0.6976)	6	3.9935 (1.979)	3	0.3593 (0.5727)	3	0.3145 (0.4441)	4	0.3914 (0.6299)	3	0.319 (0.3773)	2	0.2441 (0.2988)	5	0.8624 (1.0916)	5	1.0582 (1.2216)
$\rho = 0.6$	**1**	**0.1668 (0.2033)**	3	0.4538 (0.7191)	5	3.9935 (1.979)	3	0.4259 (0.6818)	2	0.3145 (0.4441)	3	0.4095 (0.6581)	2	0.319 (0.3773)	**1**	**0.2055 (0.2605)**	4	0.7247 (0.9874)	4	0.9198 (1.1317)
$\rho = 0.8$	**1**	**0.1219 (0.1533)**	3	0.3146 (0.4447)	6	3.9935 (1.979)	3	0.3851 (0.6195)	3	0.3145 (0.4441)	3	0.3905 (0.6285)	3	0.319 (0.3773)	2	0.1788 (0.2347)	4	0.5074 (0.7833)	5	0.8081 (1.0521)
Dense6																			
$\rho = 0.1$	2	0.0354 (0.0576)	4	0.1057 (0.0804)	8	5.1465 (0.7335)	4	0.1059 (0.1091)	3	0.071 (0.0883)	**1**	**0.0219 (0.0371)**	5	0.2091 (0.1536)	7	1.7624 (0.9135)	6	1.2592 (0.3396)		N/A
$\rho = 0.2$	**1**	**0.0336 (0.0555)**	2	0.0485 (0.0423)	8	5.1465 (0.7335)	4	0.1158 (0.1143)	3	0.071 (0.0883)	**1**	**0.0261 (0.0451)**	5	0.2091 (0.1536)	7	1.0799 (0.5268)	6	0.7642 (0.2602)		N/A
$\rho = 0.3$	2	0.0303 (0.0513)	2	0.0289 (0.0208)	8	5.1465 (0.7335)	4	0.116 (0.1143)	3	0.071 (0.0883)	**1**	**0.0187 (0.0284)**	5	0.2091 (0.1536)	7	0.7252 (0.4342)	6	0.4696 (0.1994)		N/A
$\rho = 0.4$	3	0.0299 (0.0507)	2	0.0215 (0.008)	9	5.1465 (0.7335)	5	0.1134 (0.113)	4	0.071 (0.0883)	**1**	**0.0168 (0.0148)**	6	0.2091 (0.1536)	8	0.503 (0.3642)	7	0.3182 (0.16)		N/A
$\rho = 0.5$	3	0.0262 (0.0451)	2	0.0191 (0.0028)	9	5.1465 (0.7335)	5	0.1108 (0.1117)	4	0.071 (0.0883)	**1**	**0.0169 (0.011)**	6	0.2091 (0.1536)	8	0.7522 (0.0831)	7	0.287 (0.1508)		N/A
$\rho = 0.6$	2	0.0246 (0.0424)	**1**	**0.0175 (0.0063)**	7	5.1465 (0.7335)	4	0.1083 (0.1104)	3	0.071 (0.0883)	**1**	**0.0172 (0.0087)**	5	0.2091 (0.1536)	6	0.6194 (0.1022)	5	0.2162 (0.1275)		N/A
$\rho = 0.7$	2	0.0228 (0.0389)	**1**	**0.0169 (0.0114)**	8	5.1465 (0.7335)	4	0.0919 (0.1014)	3	0.071 (0.0883)	**1**	**0.0173 (0.008)**	6	0.2091 (0.1536)	7	0.7532 (0.2293)	5	0.1744 (0.1118)		N/A
$\rho = 0.8$	**1**	**0.0228 (0.0391)**	**1**	**0.017 (0.0188)**	6	5.1465 (0.7335)	2	0.0814 (0.095)	2	0.071 (0.0883)	**1**	**0.0172 (0.0088)**	4	0.2091 (0.1536)	5	0.3865 (0.0653)	3	0.148 (0.1007)		N/A
$\rho = 0.9$	2	0.0229 (0.0392)	5	0.9263 (0.3171)	6	5.1465 (0.7335)	3	0.0684 (0.0865)	3	0.071 (0.0883)	**1**	**0.017 (0.0104)**	4	0.2091 (0.1536)	5	0.961 (1.2629)	3	0.0803 (0.0658)		N/A
MoE1																			
$\rho = 0.1$	**1**	**0.0021 (0.0023)**	6	0.1183 (0.0242)	9	2.8474 (0.1197)	4	0.0164 (0.009)	5	0.0941 (0.0218)	3	0.0145 (0.0084)	2	0.0075 (0.0059)	7	0.2358 (0.0455)	8	0.259 (0.0362)		N/A
$\rho = 0.2$	**1**	**0.0027 (0.0029)**	5	0.1257 (0.025)	8	2.8474 (0.1197)	3	0.014 (0.0083)	4	0.0941 (0.0218)	7	0.2366 (0.0346)	2	0.0075 (0.0059)	6	0.2064 (0.0444)	6	0.2098 (0.0326)		N/A
$\rho = 0.3$	**1**	**0.0026 (0.0026)**	4	0.1317 (0.0256)	8	2.8474 (0.1197)	5	0.1736 (0.0296)	3	0.0941 (0.0218)	7	0.2253 (0.0337)	2	0.0075 (0.0059)	6	0.189 (0.0433)	6	0.1943 (0.0313)		N/A
$\rho = 0.4$	**1**	**0.0087 (0.0074)**	2	0.017 (0.0091)	8	2.8474 (0.1197)	3	0.0363 (0.0135)	5	0.0941 (0.0218)	4	0.0569 (0.0169)	**1**	**0.0075 (0.0059)**	6	0.111 (0.0223)	7	0.1361 (0.0262)		N/A
$\rho = 0.5$	**1**	**0.0026 (0.0033)**	**1**	**0.0033 (0.0036)**	8	2.8474 (0.1197)	3	0.012 (0.0076)	7	0.0941 (0.0218)	4	0.0199 (0.0099)	2	0.0075 (0.0059)	5	0.0741 (0.0227)	6	0.0851 (0.0207)		N/A
$\rho = 0.6$	**1**	**0.0012 (0.0015)**	3	0.0151 (0.0086)	9	2.8474 (0.1197)	4	0.0252 (0.0112)	8	0.0941 (0.0218)	5	0.0369 (0.0136)	2	0.0075 (0.0059)	6	0.0646 (0.0244)	7	0.0738 (0.0193)		N/A
$\rho = 0.7$	**1**	**0.0014 (0.0018)**	3	0.0106 (0.0071)	8	2.8474 (0.1197)	4	0.0204 (0.01)	7	0.0941 (0.0218)	5	0.03 (0.0122)	2	0.0075 (0.0059)	6	0.0461 (0.0188)	6	0.045 (0.015)		N/A
$\rho = 0.8$	**1**	**0.0013 (0.0017)**	2	0.007 (0.0057)	7	2.8474 (0.1197)	3	0.015 (0.0086)	6	0.0941 (0.0218)	4	0.0224 (0.0105)	2	0.0075 (0.0059)	5	0.0333 (0.0142)	5	0.033 (0.0128)		N/A
$\rho = 0.9$	**1**	**0.0012 (0.0015)**	4	0.0091 (0.0066)	9	2.8474 (0.1197)	6	0.0123 (0.0077)	8	0.0941 (0.0218)	7	0.0143 (0.0084)	3	0.0075 (0.0059)	2	0.0058 (0.0022)	5	0.011 (0.0073)		N/A
MoE2																			
$\rho = 0.1$	**1**	**0.0348 (0.1283)**	4	6.2954 (19.7732)	4	7.4726 (20.7563)	3	2.9018 (12.1499)	3	2.978 (14.9935)	3	2.9288 (14.8042)	3	3.3995 (16.1187)	2	0.5369 (0.1307)	3	3.2364 (11.0914)		N/A
$\rho = 0.3$	**1**	**0.0187 (0.0714)**	4	5.557 (19.0701)	4	7.4726 (20.7563)	3	2.8771 (12.266)	3	2.978 (14.9935)	3	2.9255 (14.7904)	3	3.3995 (16.1187)	2	0.2182 (0.0593)	3	3.164 (11.2756)		N/A
$\rho = 0.5$	**1**	**0.0531 (0.388)**	4	4.6647 (18.0814)	4	7.4726 (20.7563)	3	2.8346 (12.4995)	3	2.978 (14.9935)	3	2.9227 (14.7785)	3	3.3995 (16.1187)	2	0.0893 (0.0276)	3	3.0573 (11.5803)		N/A
$\rho = 0.7$	2	0.0575 (0.0367)	4	5.2816 (18.7843)	4	7.4726 (20.7563)	3	2.8097 (12.6696)	3	2.978 (14.9935)	3	2.9201 (14.7677)	3	3.3995 (16.1187)	**1**	**0.017 (0.007)**	3	2.7986 (12.7594)		N/A
$\rho = 0.9$	2	0.0096 (0.0127)	3	3.4557 (16.2351)	4	7.4726 (20.7563)	3	2.7882 (12.8565)	3	2.978 (14.9935)	3	2.918 (14.7588)	3	3.3995 (16.1187)	**1**	**0.0031 (0.0016)**	3	2.7813 (12.9309)		N/A
Average r	**1.38**		3.75		7.21		3.48		4.08		2.83		3.67		4.77		5.50			

Table 11: Complete per-budget Spearman correlation (Sp) results for all 52 dataset–budget cases (↑). HUMAN is available only on DENSE3–5; unavailable entries are marked N/A. Undefined correlations caused by constant predictions are reported as 0.000.

Dataset / ρ	ReScaL	RF	RF_{ND}	$\text{Farseer}_{\text{d}}$	Kaplan_{d}	$\text{Chinchilla}_{\text{d}}$	Farseer	Chinchilla	Kaplan	NCPL	HUMAN
DENSE1											
$\rho = 0.1$	0.900	-0.821	0.000	1.000	0.900	0.700	0.900	0.700	0.900	0.700	N/A
$\rho = 0.2$	0.900	-0.400	0.000	1.000	0.900	0.700	0.900	0.900	0.900	0.700	N/A
$\rho = 0.3$	0.900	-0.894	0.707	1.000	0.900	0.700	0.900	1.000	0.900	0.700	N/A
$\rho = 0.4$	0.900	-0.700	0.707	1.000	0.900	0.700	0.900	1.000	0.900	0.700	N/A
$\rho = 0.5$	0.900	-0.300	-0.707	1.000	0.900	0.700	0.900	1.000	0.900	0.700	N/A
$\rho = 0.6$	0.900	-0.800	0.707	1.000	0.900	0.700	0.900	1.000	0.900	0.700	N/A
$\rho = 0.7$	0.900	-0.600	0.707	1.000	0.900	0.700	0.900	1.000	0.900	0.700	N/A
$\rho = 0.8$	0.900	0.400	0.224	1.000	0.900	0.700	0.900	1.000	0.900	0.700	N/A
$\rho = 0.9$	0.900	0.900	0.205	1.000	0.900	0.700	0.900	1.000	0.900	0.700	N/A
DENSE2											
$\rho = 0.1$	0.320	0.047	0.000	0.779	0.779	0.779	-0.779	0.779	0.779	0.725	N/A
$\rho = 0.3$	0.442	0.376	0.779	0.779	0.779	0.779	0.779	0.779	0.779	0.725	N/A
$\rho = 0.5$	0.726	0.785	-0.779	0.779	0.779	0.779	-0.779	-0.779	0.779	0.725	N/A
$\rho = 0.7$	0.780	0.498	0.779	0.779	0.779	0.779	0.779	0.779	0.779	0.725	N/A
$\rho = 0.9$	0.842	0.863	0.779	0.779	0.779	0.779	-0.779	-0.779	0.779	0.725	N/A
DENSE3											
$\rho = 0.1$	0.786	0.500	0.500	0.775	0.775	0.775	0.740	0.786	0.775	0.510	0.982
$\rho = 0.2$	0.786	0.500	0.500	0.775	0.775	0.775	-0.266	0.786	0.775	0.519	0.982
$\rho = 0.3$	0.775	0.500	0.500	0.775	0.775	0.775	0.150	0.786	0.775	0.510	0.982
$\rho = 0.4$	0.775	0.370	0.135	0.775	0.775	0.775	0.150	0.786	0.775	0.510	0.982
$\rho = 0.5$	0.558	0.200	0.135	0.775	0.775	0.775	0.139	0.786	0.775	0.510	0.982
$\rho = 0.6$	0.775	0.190	-0.266	0.775	0.775	0.775	0.139	0.786	0.775	0.510	0.982
$\rho = 0.7$	0.908	0.782	0.786	0.775	0.775	0.775	0.150	0.786	0.775	0.510	0.982
$\rho = 0.8$	0.961	0.943	0.775	0.775	0.775	0.775	0.150	0.786	0.775	0.510	0.982
$\rho = 0.9$	0.972	0.982	-0.139	0.775	0.775	0.775	0.150	0.786	0.775	0.510	0.982
DENSE4											
$\rho = 0.1$	0.848	0.845	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.711	-0.024
$\rho = 0.9$	0.889	0.884	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.711	-0.029
DENSE5											
$\rho = 0.2$	0.677	0.677	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.126	-0.042
$\rho = 0.4$	0.682	0.683	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.126	-0.041
$\rho = 0.6$	0.676	0.640	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.126	-0.041
$\rho = 0.8$	0.662	0.636	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.126	-0.041
DENSE6											
$\rho = 0.1$	0.500	-0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.2$	-0.500	-0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.3$	-0.500	-0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.4$	-0.500	-0.500	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.5$	0.500	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.6$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.7$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.8$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
$\rho = 0.9$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.000	N/A
MOE1											
$\rho = 0.1$	0.138	0.136	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.2$	0.166	0.136	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.3$	0.203	0.224	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.4$	0.203	0.465	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.5$	0.385	0.498	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.6$	0.432	-0.047	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.7$	0.317	0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.8$	0.163	0.136	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
$\rho = 0.9$	0.141	0.897	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.007	N/A
MOE2											
$\rho = 0.1$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.199	N/A
$\rho = 0.3$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.199	N/A
$\rho = 0.5$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.199	N/A
$\rho = 0.7$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.199	N/A
$\rho = 0.9$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.199	N/A

Table 12: Complete per-budget pairwise ranking accuracy (PairAcc) results for all 52 dataset–budget cases (↑). HUMAN is available only on DENSE3–5; unavailable entries are marked N/A.

Dataset / ρ	ReScaL	RF	RF_{ND}	$\text{Farseer}_{\text{d}}$	Kaplan_{d}	$\text{Chinchilla}_{\text{d}}$	Farseer	Chinchilla	Kaplan	NCPL	HUMAN
DENSE1											
$\rho = 0.1$	0.900	0.150	0.500	1.000	0.900	0.800	0.900	0.800	0.900	0.800	N/A
$\rho = 0.2$	0.900	0.300	0.500	1.000	0.900	0.800	0.900	0.900	0.900	0.800	N/A
$\rho = 0.3$	0.900	0.150	0.700	1.000	0.900	0.800	0.900	1.000	0.900	0.800	N/A
$\rho = 0.4$	0.900	0.200	0.700	1.000	0.900	0.800	0.900	1.000	0.900	0.800	N/A
$\rho = 0.5$	0.900	0.400	0.300	1.000	0.900	0.800	0.900	1.000	0.900	0.800	N/A
$\rho = 0.6$	0.900	0.200	0.700	1.000	0.900	0.800	0.900	1.000	0.900	0.800	N/A
$\rho = 0.7$	0.900	0.300	0.700	1.000	0.900	0.800	0.900	1.000	0.900	0.800	N/A
$\rho = 0.8$	0.900	0.700	0.550	1.000	0.900	0.800	0.900	1.000	0.900	0.800	N/A
$\rho = 0.9$	0.900	0.900	0.550	1.000	0.900	0.800	0.900	1.000	0.900	0.800	N/A
DENSE2											
$\rho = 0.1$	0.605	0.517	0.500	0.703	0.703	0.703	0.297	0.703	0.703	0.754	N/A
$\rho = 0.3$	0.661	0.637	0.703	0.703	0.703	0.703	0.703	0.703	0.703	0.754	N/A
$\rho = 0.5$	0.765	0.799	0.297	0.703	0.703	0.703	0.297	0.297	0.703	0.754	N/A
$\rho = 0.7$	0.790	0.676	0.703	0.703	0.703	0.703	0.703	0.703	0.703	0.754	N/A
$\rho = 0.9$	0.827	0.834	0.703	0.703	0.703	0.703	0.297	0.297	0.703	0.754	N/A
DENSE3											
$\rho = 0.1$	0.745	0.591	0.591	0.727	0.727	0.727	0.691	0.745	0.727	0.645	0.964
$\rho = 0.2$	0.745	0.591	0.591	0.727	0.727	0.727	0.418	0.745	0.727	0.664	0.964
$\rho = 0.3$	0.727	0.591	0.591	0.727	0.727	0.727	0.564	0.745	0.727	0.645	0.964
$\rho = 0.4$	0.727	0.600	0.527	0.727	0.727	0.727	0.564	0.745	0.727	0.645	0.964
$\rho = 0.5$	0.664	0.555	0.527	0.727	0.727	0.727	0.545	0.745	0.727	0.645	0.964
$\rho = 0.6$	0.727	0.536	0.418	0.727	0.727	0.727	0.545	0.745	0.727	0.645	0.964
$\rho = 0.7$	0.864	0.800	0.745	0.727	0.727	0.727	0.564	0.745	0.727	0.645	0.964
$\rho = 0.8$	0.927	0.909	0.727	0.727	0.727	0.727	0.564	0.745	0.727	0.645	0.964
$\rho = 0.9$	0.927	0.955	0.455	0.727	0.727	0.727	0.564	0.745	0.727	0.645	0.964
DENSE4											
$\rho = 0.1$	0.840	0.836	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.248	0.486
$\rho = 0.9$	0.870	0.865	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.248	0.493
DENSE5											
$\rho = 0.2$	0.756	0.756	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.460	0.487
$\rho = 0.4$	0.756	0.756	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.460	0.487
$\rho = 0.6$	0.756	0.739	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.460	0.487
$\rho = 0.8$	0.749	0.736	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.460	0.487
DENSE6											
$\rho = 0.1$	0.667	0.333	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.2$	0.333	0.333	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.3$	0.333	0.333	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.4$	0.333	0.333	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.5$	0.667	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.6$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.7$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.8$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
$\rho = 0.9$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.000	N/A
MOE1											
$\rho = 0.1$	0.552	0.560	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.2$	0.562	0.557	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.3$	0.574	0.580	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.4$	0.590	0.670	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.5$	0.635	0.684	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.6$	0.656	0.496	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.7$	0.616	0.513	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.8$	0.562	0.559	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
$\rho = 0.9$	0.560	0.868	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.496	N/A
MOE2											
$\rho = 0.1$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.578	N/A
$\rho = 0.3$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.578	N/A
$\rho = 0.5$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.578	N/A
$\rho = 0.7$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.578	N/A
$\rho = 0.9$	1.000	1.000	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.578	N/A

Table 13: Complete per-budget top-1 selection accuracy (Top1) results for all 52 dataset–budget cases (↑). HUMAN is available only on DENSE3–5; unavailable entries are marked N/A.

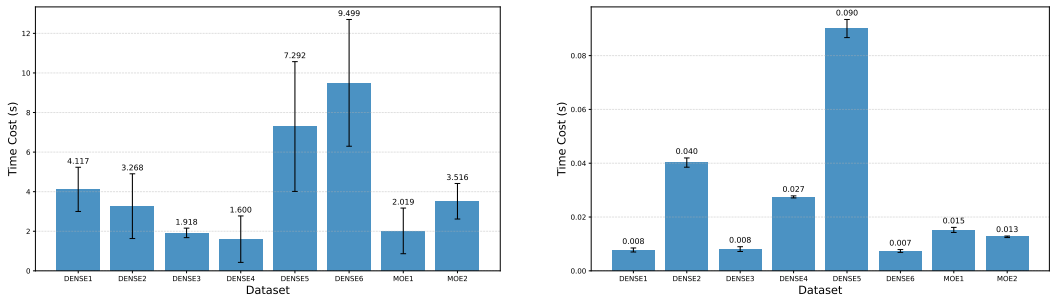
Dataset / ρ	ReScaL	RF	RF_{ND}	$\text{Farseer}_{\text{d}}$	Kaplan_{d}	$\text{Chinchilla}_{\text{d}}$	Farseer	Chinchilla	Kaplan	NCPL	HUMAN
DENSE1											
$\rho = 0.1$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	0.000	1.000	1.000	N/A
$\rho = 0.2$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	0.000	1.000	1.000	N/A
$\rho = 0.3$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.4$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.5$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.6$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.7$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.8$	1.000	0.000	0.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.9$	1.000	1.000	0.000	1.000	1.000	0.000	1.000	1.000	1.000	1.000	N/A
DENSE2											
$\rho = 0.1$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.3$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.5$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.7$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.9$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
DENSE3											
$\rho = 0.1$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.2$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.3$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.4$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.5$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.6$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.7$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.8$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
$\rho = 0.9$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
DENSE4											
$\rho = 0.1$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\rho = 0.9$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DENSE5											
$\rho = 0.2$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\rho = 0.4$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\rho = 0.6$	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\rho = 0.8$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DENSE6											
$\rho = 0.1$	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.2$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.3$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.4$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.5$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.6$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.7$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.8$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.9$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
MOE1											
$\rho = 0.1$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.2$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.3$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.4$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.5$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.6$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.7$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.8$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.9$	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
MOE2											
$\rho = 0.1$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.3$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.5$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.7$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A
$\rho = 0.9$	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	N/A

Table 14: Complete per-budget regret (Reg) results for all 52 dataset–budget cases (↓). HUMAN is available only on DENSE3–5; unavailable entries are marked N/A.

Dataset / ρ	ReScaL	RF	RF_{ND}	$\text{Farseer}_{\text{d}}$	Kaplan_{d}	$\text{Chinchilla}_{\text{d}}$	Farseer	Chinchilla	Kaplan	NCPL	HUMAN
DENSE1											
$\rho = 0.1$	0.000	0.245	0.245	0.000	0.000	0.007	0.000	0.007	0.000	0.000	N/A
$\rho = 0.2$	0.000	0.039	0.245	0.000	0.000	0.007	0.000	0.007	0.000	0.000	N/A
$\rho = 0.3$	0.000	0.245	0.039	0.000	0.000	0.007	0.000	0.000	0.000	0.000	N/A
$\rho = 0.4$	0.000	0.245	0.039	0.000	0.000	0.007	0.000	0.000	0.000	0.000	N/A
$\rho = 0.5$	0.000	0.039	0.245	0.000	0.000	0.007	0.000	0.000	0.000	0.000	N/A
$\rho = 0.6$	0.000	0.039	0.039	0.000	0.000	0.007	0.000	0.000	0.000	0.000	N/A
$\rho = 0.7$	0.000	0.039	0.039	0.000	0.000	0.007	0.000	0.000	0.000	0.000	N/A
$\rho = 0.8$	0.000	0.039	0.039	0.000	0.000	0.007	0.000	0.000	0.000	0.000	N/A
$\rho = 0.9$	0.000	0.000	0.039	0.000	0.000	0.007	0.000	0.000	0.000	0.000	N/A
DENSE2											
$\rho = 0.1$	0.143	0.143	0.012	0.012	0.012	0.012	0.311	0.012	0.012	0.032	N/A
$\rho = 0.3$	0.012	0.027	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.032	N/A
$\rho = 0.5$	0.098	0.012	0.311	0.012	0.012	0.012	0.311	0.311	0.012	0.032	N/A
$\rho = 0.7$	0.028	0.030	0.012	0.012	0.012	0.012	0.012	0.012	0.012	0.032	N/A
$\rho = 0.9$	0.028	0.028	0.012	0.012	0.012	0.012	0.311	0.311	0.012	0.032	N/A
DENSE3											
$\rho = 0.1$	0.022	0.649	0.649	0.022	0.022	0.022	0.022	0.022	0.022	0.023	0.000
$\rho = 0.2$	0.022	0.649	0.649	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
$\rho = 0.3$	0.022	0.649	0.649	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
$\rho = 0.4$	0.022	0.527	0.527	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
$\rho = 0.5$	0.115	0.115	0.527	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
$\rho = 0.6$	0.022	0.527	1.967	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
$\rho = 0.7$	0.023	0.023	0.022	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
$\rho = 0.8$	0.001	0.011	0.022	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
$\rho = 0.9$	0.000	0.000	0.527	0.022	0.022	0.022	1.967	0.022	0.022	0.023	0.000
DENSE4											
$\rho = 0.1$	0.000	0.000	2.207	2.207	2.207	2.207	2.207	2.207	2.207	2.243	2.207
$\rho = 0.9$	0.003	0.003	2.207	2.207	2.207	2.207	2.207	2.207	2.207	2.243	2.207
DENSE5											
$\rho = 0.2$	0.000	0.000	1.083	1.083	1.083	1.083	1.083	1.083	1.083	2.385	1.475
$\rho = 0.4$	0.339	0.000	1.083	1.083	1.083	1.083	1.083	1.083	1.083	2.385	1.475
$\rho = 0.6$	0.000	0.191	1.083	1.083	1.083	1.083	1.083	1.083	1.083	2.385	1.475
$\rho = 0.8$	0.013	0.191	1.083	1.083	1.083	1.083	1.083	1.083	1.083	2.385	1.475
DENSE6											
$\rho = 0.1$	0.000	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.2$	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.3$	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.4$	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.5$	0.000	0.000	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.6$	0.000	0.000	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.7$	0.000	0.000	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.8$	0.000	0.000	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
$\rho = 0.9$	0.000	0.000	0.284	0.284	0.284	0.284	0.284	0.284	0.284	0.284	N/A
MOE1											
$\rho = 0.1$	0.055	0.055	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.2$	0.055	0.055	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.3$	0.055	0.055	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.4$	0.055	0.000	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.5$	0.010	0.010	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.6$	0.010	0.058	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.7$	0.105	0.040	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.8$	0.100	0.036	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
$\rho = 0.9$	0.036	0.000	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.044	N/A
MOE2											
$\rho = 0.1$	0.000	0.000	8.779	8.779	8.779	8.779	8.779	8.779	8.779	0.313	N/A
$\rho = 0.3$	0.000	0.000	8.779	8.779	8.779	8.779	8.779	8.779	8.779	0.313	N/A
$\rho = 0.5$	0.000	0.000	8.779	8.779	8.779	8.779	8.779	8.779	8.779	0.313	N/A
$\rho = 0.7$	0.000	0.000	8.779	8.779	8.779	8.779	8.779	8.779	8.779	0.313	N/A
$\rho = 0.9$	0.000	0.000	8.779	8.779	8.779	8.779	8.779	8.779	8.779	0.313	N/A

Table 15: Complete per-budget normalized regret (NReg) results for all 52 dataset–budget cases (↓). HUMAN is available only on DENSE3–5; unavailable entries are marked N/A.

Dataset / ρ	ReScaL	RF	RF_{ND}	$\text{Farseer}_{\text{d}}$	Kaplan_{d}	$\text{Chinchilla}_{\text{d}}$	Farseer	Chinchilla	Kaplan	NCPL	HUMAN
DENSE1											
$\rho = 0.1$	0.000	1.000	1.000	0.000	0.000	0.029	0.000	0.029	0.000	0.000	N/A
$\rho = 0.2$	0.000	0.160	1.000	0.000	0.000	0.029	0.000	0.029	0.000	0.000	N/A
$\rho = 0.3$	0.000	1.000	0.160	0.000	0.000	0.029	0.000	0.000	0.000	0.000	N/A
$\rho = 0.4$	0.000	1.000	0.160	0.000	0.000	0.029	0.000	0.000	0.000	0.000	N/A
$\rho = 0.5$	0.000	0.160	1.000	0.000	0.000	0.029	0.000	0.000	0.000	0.000	N/A
$\rho = 0.6$	0.000	0.160	0.160	0.000	0.000	0.029	0.000	0.000	0.000	0.000	N/A
$\rho = 0.7$	0.000	0.160	0.160	0.000	0.000	0.029	0.000	0.000	0.000	0.000	N/A
$\rho = 0.8$	0.000	0.160	0.160	0.000	0.000	0.029	0.000	0.000	0.000	0.000	N/A
$\rho = 0.9$	0.000	0.000	0.160	0.000	0.000	0.029	0.000	0.000	0.000	0.000	N/A
DENSE2											
$\rho = 0.1$	0.316	0.316	0.026	0.026	0.026	0.026	0.688	0.026	0.026	0.071	N/A
$\rho = 0.3$	0.026	0.059	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.071	N/A
$\rho = 0.5$	0.217	0.026	0.688	0.026	0.026	0.026	0.688	0.688	0.026	0.071	N/A
$\rho = 0.7$	0.063	0.067	0.026	0.026	0.026	0.026	0.026	0.026	0.026	0.071	N/A
$\rho = 0.9$	0.063	0.063	0.026	0.026	0.026	0.026	0.688	0.688	0.026	0.071	N/A
DENSE3											
$\rho = 0.1$	0.011	0.330	0.330	0.011	0.011	0.011	0.011	0.011	0.011	0.012	0.000
$\rho = 0.2$	0.011	0.330	0.330	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
$\rho = 0.3$	0.011	0.330	0.330	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
$\rho = 0.4$	0.011	0.268	0.268	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
$\rho = 0.5$	0.058	0.058	0.268	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
$\rho = 0.6$	0.011	0.268	1.000	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
$\rho = 0.7$	0.012	0.012	0.011	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
$\rho = 0.8$	0.000	0.006	0.011	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
$\rho = 0.9$	0.000	0.000	0.268	0.011	0.011	0.011	1.000	0.011	0.011	0.012	0.000
DENSE4											
$\rho = 0.1$	0.000	0.000	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.936	0.921
$\rho = 0.9$	0.001	0.001	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.936	0.921
DENSE5											
$\rho = 0.2$	0.000	0.000	0.445	0.445	0.445	0.445	0.445	0.445	0.445	0.980	0.606
$\rho = 0.4$	0.139	0.000	0.445	0.445	0.445	0.445	0.445	0.445	0.445	0.980	0.606
$\rho = 0.6$	0.000	0.078	0.445	0.445	0.445	0.445	0.445	0.445	0.445	0.980	0.606
$\rho = 0.8$	0.005	0.078	0.445	0.445	0.445	0.445	0.445	0.445	0.445	0.980	0.606
DENSE6											
$\rho = 0.1$	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.2$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.3$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.4$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.5$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.6$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.7$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.8$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
$\rho = 0.9$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	N/A
MOE1											
$\rho = 0.1$	0.396	0.396	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.2$	0.396	0.396	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.3$	0.396	0.396	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.4$	0.396	0.000	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.5$	0.071	0.071	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.6$	0.071	0.420	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.7$	0.760	0.292	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.8$	0.727	0.260	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
$\rho = 0.9$	0.260	0.000	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.322	N/A
MOE2											
$\rho = 0.1$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.036	N/A
$\rho = 0.3$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.036	N/A
$\rho = 0.5$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.036	N/A
$\rho = 0.7$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.036	N/A
$\rho = 0.9$	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.036	N/A



(a) Training time

(b) Per-prediction time

Figure 4: Computational overhead for ReScaL across all dataset-budget pairs.

For NCPL, we retain the common train/test split and available configuration fields used by every other method. We follow its residual-to-Chinchilla formulation, but refit the Chinchilla component on each corresponding pilot split rather than using parameters preset for NCPL's original Marin/StepLaw setting. Because some source datasets lack optimizer, schedule, or other fields present in NCPL's original training logs, these results should be read as a controlled comparison on our benchmark, not as a general assessment of NCPL in its native setting.

C.4 Full RQ2 Ablation Study Results

In Section 4.4, we conduct an ablation study by removing exactly one component from ReScaL (diverse fitting, scaling law ensemble, residual learning, or dynamic α) while keeping the others unchanged; the aggregated results across all budget sizes for each dataset are summarized in Table 4.

In contrast, Table 16 provides a more granular view by reporting results for each dataset-budget pair along with corresponding average rank summaries. Note that Table 4 aggregates results *within each dataset across all* ρ before ranking, so the two average rank summaries are not directly comparable.

Under this finer-grained view, removing dynamic α remains the most consistently harmful (average $r = 2.62$), confirming that adaptive residual gating is critical across budgets. The next largest degradation comes from removing diverse fitting (average $r = 2.13$), reflecting that constrained warm-start adaptation stabilizes scaling-law refits across many low-budget cases (especially on the MoE settings). By contrast, removing residual learning yields the largest degradations on specific datasets with stronger non-scaling effects (e.g., DENSE4 and DENSE5), but its rank impact is less uniform across all 52 cases (average $r = 1.83$). Finally, removing ensembling is comparatively mild on average (average $r = 1.77$), though its absence is still largely impactful in datasets like DENSE1, DENSE3, and DENSE6.

We also additionally examine the ablation of ReScaL using MAE and RMSE, as shown in Figures 17 and 18, respectively; the conclusion remain the same.

C.5 Full RQ3 Sensitivity Analysis Results

Section 4.5 of the main paper presents an analysis of sensitivity to the joint effect of two parameters: the temperature t (controlling the softmax ensemble) and sigmoid scaling hyperparameter s (controlling the steepness of the sigmoid function in α for residual-augmented prediction).

Figures 6–13 show the sensitivity heatmaps across all pilot budget sizes for each dataset. In a nutshell, the heatmaps reveal that moderate s values (10–20) combined with low to moderate t values (0.1–0.5) generally yield the best performance across datasets, validating the default choices of $t = 0.1$ and $s = 20$ used in the main experiments, and demonstrating that ReScaL is robust to hyperparameter choices within reasonable ranges, with performance degrading primarily at extreme values (e.g., very high t or very low s).

Table 16: Full ablation study results showing per dataset-budget pair performance. The ones with the best Scott-Knott ESD ranks are highlighted in **bold**.

Dataset	ReScaL		w/o dynamic α		w/o ensemble		w/o residual		w/o diverse fitting	
	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)
DENSE1										
$\rho = 0.1$	2	0.0204 (0.02)	4	0.0893 (0.03)	**1**	**0.0146 (0.00)**	2	0.0225 (0.02)	3	0.0464 (0.01)
$\rho = 0.2$	**1**	**0.0134 (0.00)**	4	0.1108 (0.04)	2	0.0228 (0.02)	**1**	**0.0134 (0.00)**	3	0.0546 (0.02)
$\rho = 0.3$	**1**	**0.0152 (0.01)**	4	0.095 (0.03)	2	0.0269 (0.01)	**1**	**0.0152 (0.01)**	3	0.0528 (0.02)
$\rho = 0.4$	**1**	**0.0162 (0.01)**	4	0.0889 (0.03)	2	0.0368 (0.02)	**1**	**0.0162 (0.01)**	3	0.0504 (0.02)
$\rho = 0.5$	**1**	**0.0192 (0.02)**	4	0.0832 (0.03)	2	0.0385 (0.01)	**1**	**0.0192 (0.02)**	3	0.0493 (0.02)
$\rho = 0.6$	**1**	**0.0237 (0.02)**	4	0.079 (0.03)	2	0.0362 (0.02)	**1**	**0.0232 (0.02)**	3	0.0472 (0.02)
$\rho = 0.7$	**1**	**0.0253 (0.02)**	4	0.0722 (0.03)	2	0.0307 (0.02)	**1**	**0.0252 (0.02)**	3	0.0448 (0.02)
$\rho = 0.8$	**1**	**0.0246 (0.02)**	3	0.0606 (0.03)	**1**	**0.0265 (0.02)**	**1**	**0.0245 (0.02)**	2	0.0412 (0.02)
$\rho = 0.9$	2	0.0227 (0.02)	4	0.047 (0.02)	**1**	**0.0154 (0.01)**	2	0.0227 (0.02)	3	0.0365 (0.01)
DENSE2										
$\rho = 0.1$	**1**	**0.1059 (0.18)**	2	0.163 (0.19)	**1**	**0.0961 (0.14)**	**1**	**0.0918 (0.04)**	2	0.1308 (0.15)
$\rho = 0.3$	**1**	**0.0458 (0.03)**	2	0.0506 (0.02)	**1**	**0.045 (0.03)**	3	0.0739 (0.04)	**1**	**0.0469 (0.02)**
$\rho = 0.5$	2	0.0278 (0.02)	**1**	**0.0214 (0.01)**	2	0.0278 (0.02)	3	0.0544 (0.03)	2	0.0261 (0.02)
$\rho = 0.7$	2	0.0364 (0.04)	**1**	**0.0242 (0.04)**	2	0.0362 (0.03)	3	0.0589 (0.03)	2	0.0349 (0.03)
$\rho = 0.9$	3	0.0241 (0.01)	**1**	**0.0126 (0.01)**	2	0.0207 (0.01)	4	0.0495 (0.03)	2	0.0212 (0.01)
DENSE3										
$\rho = 0.1$	2	0.0955 (0.02)	**1**	**0.0857 (0.02)**	3	0.1339 (0.07)	2	0.0933 (0.02)	4	0.1674 (0.04)
$\rho = 0.2$	**1**	**0.0479 (0.06)**	2	0.1356 (0.08)	3	0.2286 (0.05)	**1**	**0.0492 (0.06)**	**1**	**0.0534 (0.06)**
$\rho = 0.3$	**1**	**0.0413 (0.06)**	3	0.1457 (0.07)	2	0.1306 (0.02)	**1**	**0.0414 (0.06)**	**1**	**0.0456 (0.07)**
$\rho = 0.4$	**1**	**0.0381 (0.05)**	3	0.1443 (0.05)	3	0.1413 (0.07)	**1**	**0.0379 (0.05)**	2	0.0759 (0.04)
$\rho = 0.5$	**1**	**0.051 (0.03)**	**1**	**0.0529 (0.07)**	2	0.0761 (0.03)	**1**	**0.051 (0.03)**	**1**	**0.0419 (0.05)**
$\rho = 0.6$	2	0.0919 (0.04)	3	0.122 (0.04)	4	0.5102 (0.24)	2	0.092 (0.04)	**1**	**0.0673 (0.03)**
$\rho = 0.7$	3	0.1052 (0.06)	4	0.1599 (0.08)	**1**	**0.0318 (0.03)**	3	0.1106 (0.05)	2	0.0905 (0.04)
$\rho = 0.8$	2	0.1084 (0.06)	3	0.1631 (0.08)	**1**	**0.0177 (0.03)**	2	0.1134 (0.05)	2	0.1043 (0.05)
$\rho = 0.9$	2	0.1131 (0.06)	4	0.1662 (0.08)	**1**	**0.0553 (0.03)**	2	0.1197 (0.06)	3	0.1347 (0.06)
DENSE4										
$\rho = 0.1$	**1**	**0.2773 (0.13)**	**1**	**0.2599 (0.12)**	**1**	**0.2772 (0.13)**	2	0.4474 (0.14)	**1**	**0.2734 (0.13)**
$\rho = 0.8$	2	0.0687 (0.06)	**1**	**0.0573 (0.05)**	2	0.0691 (0.06)	3	0.4849 (0.27)	**1**	**0.0617 (0.06)**
DENSE5										
$\rho = 0.2$	2	0.2789 (0.20)	**1**	**0.1828 (0.16)**	4	0.3734 (0.21)	3	0.3331 (0.13)	2	0.2729 (0.19)
$\rho = 0.4$	**1**	**0.1292 (0.07)**	**1**	**0.1354 (0.07)**	**1**	**0.1441 (0.13)**	2	0.4832 (0.14)	**1**	**0.128 (0.08)**
$\rho = 0.6$	**1**	**0.1381 (0.07)**	**1**	**0.1452 (0.08)**	**1**	**0.136 (0.12)**	2	0.4927 (0.14)	**1**	**0.136 (0.07)**
$\rho = 0.8$	**1**	**0.1231 (0.09)**	**1**	**0.1236 (0.08)**	**1**	**0.1323 (0.11)**	2	0.5028 (0.14)	**1**	**0.1219 (0.09)**
DENSE6										
$\rho = 0.1$	**1**	**0.0469 (0.05)**	3	0.11 (0.05)	2	0.06 (0.05)	**1**	**0.0469 (0.05)**	**1**	**0.0423 (0.02)**
$\rho = 0.2$	2	0.0445 (0.05)	4	0.1156 (0.05)	3	0.0717 (0.03)	2	0.0444 (0.05)	**1**	**0.035 (0.04)**
$\rho = 0.3$	**1**	**0.0397 (0.05)**	3	0.1117 (0.05)	2	0.0742 (0.05)	**1**	**0.0397 (0.05)**	**1**	**0.0388 (0.05)**
$\rho = 0.4$	**1**	**0.039 (0.05)**	3	0.1069 (0.05)	2	0.0792 (0.06)	**1**	**0.039 (0.05)**	4	0.1379 (0.05)
$\rho = 0.5$	**1**	**0.0338 (0.05)**	3	0.056 (0.02)	2	0.0438 (0.02)	**1**	**0.0338 (0.05)**	**1**	**0.0334 (0.05)**
$\rho = 0.6$	2	0.0328 (0.05)	3	0.0508 (0.01)	**1**	**0.0256 (0.02)**	2	0.0328 (0.05)	2	0.0335 (0.05)
$\rho = 0.7$	2	0.0327 (0.05)	3	0.0482 (0.00)	**1**	**0.0243 (0.02)**	2	0.0327 (0.05)	2	0.0344 (0.04)
$\rho = 0.8$	2	0.0327 (0.05)	3	0.0436 (0.02)	**1**	**0.0244 (0.02)**	2	0.0327 (0.05)	2	0.0334 (0.05)
$\rho = 0.9$	**1**	**0.0325 (0.05)**	2	0.0408 (0.02)	3	0.0826 (0.13)	**1**	**0.0326 (0.05)**	**1**	**0.0328 (0.05)**
MOE1										
$\rho = 0.1$	**1**	**0.0173 (0.01)**	3	0.0691 (0.02)	**1**	**0.0181 (0.01)**	2	0.0493 (0.02)	**1**	**0.0185 (0.01)**
$\rho = 0.2$	**1**	**0.0201 (0.01)**	2	0.058 (0.02)	**1**	**0.0199 (0.01)**	2	0.0544 (0.02)	**1**	**0.021 (0.01)**
$\rho = 0.3$	**1**	**0.0196 (0.01)**	4	0.2513 (0.02)	**1**	**0.0205 (0.01)**	2	0.0421 (0.02)	3	0.1427 (0.02)
$\rho = 0.4$	2	0.0377 (0.02)	3	0.0993 (0.02)	2	0.0371 (0.02)	**1**	**0.0128 (0.01)**	3	0.1001 (0.02)
$\rho = 0.5$	**1**	**0.0193 (0.01)**	3	0.0539 (0.02)	**1**	**0.0217 (0.02)**	**1**	**0.0201 (0.01)**	2	0.0439 (0.02)
$\rho = 0.6$	**1**	**0.0131 (0.01)**	4	0.068 (0.02)	2	0.0268 (0.02)	**1**	**0.0131 (0.01)**	3	0.0536 (0.02)
$\rho = 0.7$	**1**	**0.0143 (0.01)**	4	0.0641 (0.02)	2	0.0362 (0.02)	**1**	**0.0139 (0.01)**	3	0.0519 (0.02)
$\rho = 0.8$	**1**	**0.0137 (0.01)**	3	0.0453 (0.02)	2	0.0302 (0.02)	**1**	**0.0137 (0.01)**	3	0.0446 (0.02)
$\rho = 0.9$	**1**	**0.013 (0.01)**	3	0.0312 (0.02)	2	0.0278 (0.02)	**1**	**0.013 (0.01)**	3	0.031 (0.02)
MOE2										
$\rho = 0.1$	**1**	**0.0384 (0.02)**	3	0.2205 (0.03)	**1**	**0.0383 (0.02)**	2	0.1079 (0.15)	4	0.3195 (0.04)
$\rho = 0.3$	**1**	**0.026 (0.02)**	2	0.0884 (0.02)	**1**	**0.0242 (0.02)**	3	0.1209 (0.15)	3	0.1315 (0.02)
$\rho = 0.5$	3	0.0946 (0.01)	**1**	**0.0423 (0.01)**	2	0.0929 (0.01)	4	0.1456 (0.15)	3	0.0961 (0.01)
$\rho = 0.7$	2	0.0996 (0.01)	**1**	**0.0484 (0.01)**	2	0.1004 (0.01)	4	0.1591 (0.15)	3	0.1135 (0.02)
$\rho = 0.9$	2	0.0377 (0.01)	**1**	**0.0257 (0.01)**	2	0.0392 (0.01)	3	0.1611 (0.14)	2	0.0383 (0.01)
Average r	**1.44**		2.62		1.77		1.83		2.13	

C.6 COMPUTATIONAL OVERHEAD ANALYSIS

To assess the practical feasibility of ReScaL, we measure the computational overhead in two phases: (1) *training time*, which includes all preprocessing steps, diverse fitting of multiple scaling law variants ($K = 9$ laws) with constrained optimization, 5-fold cross-validation for ensemble weighting, and training of the residual model (RF); and (2) *prediction time*, which includes inference of all scaling laws in the ensemble, ensemble weighting computation, inference of the residual model, and computation of the dynamic weight α based on R^2_{partial} for generating a single prediction for a target scale setting $(N^\star, D^\star, \mathbf{x}^\star)$. All experiments are conducted on a server running Ubuntu with Intel(R) Xeon(R) Platinum 40-core CPUs and 500GB RAM.

Table 17: Full ablation study results using MAE. The formate follows Table 16.

Dataset	ReScaL		w/o dynamic α		w/o ensemble		w/o residual		w/o diverse fitting	
	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)
DENSE1										
$\rho = 0.1$	2	0.0373 (0.0301)	4	0.1655 (0.0675)	**1**	**0.0269 (0.0078)**	2	0.0422 (0.0306)	3	0.0858 (0.0253)
$\rho = 0.2$	**1**	**0.0254 (0.005)**	4	0.2007 (0.0592)	2	0.0411 (0.0286)	**1**	**0.0249 (0.0082)**	3	0.1006 (0.0401)
$\rho = 0.3$	**1**	**0.0276 (0.0098)**	4	0.1719 (0.0537)	2	0.049 (0.0239)	**1**	**0.0276 (0.0095)**	3	0.0972 (0.0396)
$\rho = 0.4$	**1**	**0.0291 (0.0214)**	4	0.1606 (0.0528)	2	0.0668 (0.0282)	**1**	**0.0293 (0.0228)**	3	0.0926 (0.0395)
$\rho = 0.5$	**1**	**0.0346 (0.0275)**	4	0.1503 (0.0511)	2	0.0702 (0.0258)	**1**	**0.0346 (0.0276)**	3	0.0906 (0.0421)
$\rho = 0.6$	**1**	**0.0421 (0.0271)**	4	0.1426 (0.0514)	2	0.0656 (0.0282)	**1**	**0.0419 (0.0272)**	3	0.0866 (0.0418)
$\rho = 0.7$	**1**	**0.0454 (0.0271)**	4	0.1302 (0.0498)	2	0.0554 (0.0292)	**1**	**0.0455 (0.0271)**	3	0.0822 (0.0379)
$\rho = 0.8$	**1**	**0.0444 (0.0269)**	3	0.1091 (0.0457)	**1**	**0.0478 (0.027)**	**1**	**0.0443 (0.027)**	2	0.0754 (0.0283)
$\rho = 0.9$	2	0.041 (0.027)	4	0.0843 (0.0438)	**1**	**0.0279 (0.0191)**	2	0.0408 (0.0271)	3	0.0666 (0.0189)
DENSE2										
$\rho = 0.1$	**1**	**0.2387 (0.3859)**	2	0.3824 (0.4573)	**1**	**0.2296 (0.3445)**	**1**	**0.2178 (0.1017)**	2	0.3119 (0.3701)
$\rho = 0.3$	**1**	**0.108 (0.0629)**	2	0.1178 (0.0539)	**1**	**0.1057 (0.0626)**	3	0.1763 (0.1008)	**1**	**0.1099 (0.0582)**
$\rho = 0.5$	2	0.0614 (0.0358)	**1**	**0.0487 (0.0325)**	2	0.0639 (0.0368)	3	0.1298 (0.0767)	2	0.06 (0.0347)
$\rho = 0.7$	2	0.0864 (0.0994)	**1**	**0.0571 (0.1116)**	2	0.0853 (0.0848)	3	0.1404 (0.0823)	2	0.0826 (0.0826)
$\rho = 0.9$	3	0.0564 (0.0343)	**1**	**0.0292 (0.0259)**	2	0.0481 (0.034)	4	0.118 (0.0705)	2	0.0499 (0.0304)
DENSE3										
$\rho = 0.1$	**1**	**0.2496 (0.0922)**	**1**	**0.2313 (0.1008)**	2	0.3287 (0.1482)	**1**	**0.247 (0.084)**	3	0.4482 (0.1924)
$\rho = 0.2$	**1**	**0.1549 (0.2277)**	2	0.393 (0.321)	3	0.5801 (0.0669)	**1**	**0.1592 (0.2328)**	**1**	**0.1753 (0.2747)**
$\rho = 0.3$	**1**	**0.1365 (0.2142)**	3	0.4138 (0.2891)	2	0.3367 (0.0502)	**1**	**0.136 (0.2108)**	**1**	**0.1581 (0.2927)**
$\rho = 0.4$	**1**	**0.1214 (0.1826)**	4	0.4013 (0.2323)	3	0.346 (0.1372)	**1**	**0.1231 (0.1864)**	2	0.2226 (0.2064)
$\rho = 0.5$	**1**	**0.1441 (0.1158)**	2	0.1698 (0.2459)	2	0.1925 (0.0627)	**1**	**0.1452 (0.1145)**	**1**	**0.1365 (0.2188)**
$\rho = 0.6$	2	0.2326 (0.0819)	3	0.3133 (0.1153)	4	1.2322 (0.5616)	2	0.2325 (0.0816)	**1**	**0.1874 (0.1529)**
$\rho = 0.7$	3	0.2513 (0.1298)	5	0.3828 (0.1911)	**1**	**0.0852 (0.0997)**	4	0.2754 (0.1019)	2	0.2199 (0.0911)
$\rho = 0.8$	2	0.2593 (0.136)	4	0.3912 (0.1888)	**1**	**0.049 (0.0731)**	3	0.2805 (0.1081)	2	0.2532 (0.1019)
$\rho = 0.9$	2	0.2706 (0.1441)	4	0.3989 (0.1894)	**1**	**0.1347 (0.0605)**	2	0.2958 (0.1184)	3	0.3265 (0.1295)
DENSE4										
$\rho = 0.1$	2	0.5841 (0.1635)	**1**	**0.5519 (0.1648)**	2	0.5855 (0.1623)	3	1.205 (0.6546)	2	0.5776 (0.1629)
$\rho = 0.9$	**1**	**0.1422 (0.107)**	**1**	**0.12 (0.1113)**	**1**	**0.1422 (0.1108)**	2	1.4209 (0.904)	**1**	**0.1276 (0.1105)**
DENSE5										
$\rho = 0.2$	2	0.5891 (0.3226)	**1**	**0.3809 (0.2868)**	3	0.8213 (0.3506)	3	0.8535 (0.4396)	2	0.5758 (0.3199)
$\rho = 0.4$	**1**	**0.3116 (0.1907)**	**1**	**0.3392 (0.2094)**	**1**	**0.3107 (0.2414)**	2	1.2516 (0.5399)	**1**	**0.3065 (0.1907)**
$\rho = 0.6$	2	0.3468 (0.2176)	2	0.3745 (0.2467)	**1**	**0.2951 (0.2285)**	3	1.275 (0.5432)	2	0.3381 (0.2112)
$\rho = 0.8$	**1**	**0.2834 (0.2053)**	**1**	**0.2953 (0.2014)**	**1**	**0.2902 (0.2214)**	2	1.2993 (0.5461)	**1**	**0.2835 (0.2048)**
DENSE6										
$\rho = 0.1$	**1**	**0.1359 (0.1588)**	3	0.3116 (0.1588)	2	0.1722 (0.1589)	**1**	**0.1363 (0.1588)**	**1**	**0.1198 (0.0612)**
$\rho = 0.2$	**1**	**0.1295 (0.1588)**	3	0.3272 (0.1588)	2	0.2017 (0.0848)	**1**	**0.1293 (0.1588)**	**1**	**0.1017 (0.1236)**
$\rho = 0.3$	**1**	**0.116 (0.1588)**	3	0.3165 (0.1588)	2	0.2098 (0.1516)	**1**	**0.1162 (0.1588)**	**1**	**0.1136 (0.1588)**
$\rho = 0.4$	**1**	**0.1143 (0.1588)**	3	0.303 (0.1588)	2	0.2211 (0.1595)	**1**	**0.1143 (0.1588)**	4	0.3892 (0.1588)
$\rho = 0.5$	**1**	**0.0997 (0.1561)**	3	0.1536 (0.0574)	2	0.123 (0.0665)	**1**	**0.0997 (0.1561)**	**1**	**0.0986 (0.1541)**
$\rho = 0.6$	2	0.0968 (0.151)	3	0.1407 (0.0154)	**1**	**0.0738 (0.0696)**	2	0.0968 (0.151)	2	0.0981 (0.1358)
$\rho = 0.7$	2	0.0962 (0.1425)	3	0.1343 (0.0139)	**1**	**0.0691 (0.0469)**	2	0.0962 (0.1425)	2	0.1003 (0.1282)
$\rho = 0.8$	2	0.0961 (0.1428)	3	0.1229 (0.0504)	**1**	**0.0701 (0.0593)**	2	0.0961 (0.1429)	2	0.0979 (0.1367)
$\rho = 0.9$	**1**	**0.0957 (0.1434)**	**1**	**0.1161 (0.0728)**	2	0.2442 (0.3979)	**1**	**0.0959 (0.1437)**	**1**	**0.0965 (0.1408)**
MOE1										
$\rho = 0.1$	**1**	**0.0363 (0.0258)**	3	0.1521 (0.0467)	**1**	**0.0396 (0.0259)**	2	0.1086 (0.0355)	**1**	**0.0405 (0.0254)**
$\rho = 0.2$	**1**	**0.0478 (0.0286)**	2	0.1267 (0.0493)	**1**	**0.0437 (0.0261)**	2	0.1199 (0.0355)	**1**	**0.0461 (0.0282)**
$\rho = 0.3$	**1**	**0.0495 (0.0297)**	4	0.5515 (0.0517)	**1**	**0.0449 (0.0264)**	2	0.093 (0.0355)	3	0.3126 (0.0522)
$\rho = 0.4$	2	0.0823 (0.0464)	3	0.2176 (0.0396)	2	0.081 (0.0448)	**1**	**0.0281 (0.0215)**	3	0.2193 (0.0402)
$\rho = 0.5$	**1**	**0.0422 (0.0298)**	3	0.118 (0.0479)	**1**	**0.0476 (0.0378)**	**1**	**0.0446 (0.0281)**	2	0.0959 (0.0506)
$\rho = 0.6$	**1**	**0.0288 (0.0205)**	4	0.1486 (0.0355)	2	0.0587 (0.0486)	**1**	**0.0287 (0.0206)**	3	0.117 (0.0355)
$\rho = 0.7$	**1**	**0.0312 (0.0208)**	4	0.14 (0.0355)	2	0.0789 (0.0378)	**1**	**0.0302 (0.0206)**	3	0.1133 (0.0349)
$\rho = 0.8$	**1**	**0.0301 (0.0205)**	3	0.0989 (0.0355)	2	0.0657 (0.0388)	**1**	**0.0299 (0.0204)**	3	0.0973 (0.0355)
$\rho = 0.9$	**1**	**0.0285 (0.0207)**	3	0.0679 (0.0353)	2	0.0605 (0.0363)	**1**	**0.0285 (0.0207)**	3	0.0674 (0.0353)
MOE2										
$\rho = 0.1$	**1**	**0.1117 (0.1532)**	2	0.5237 (0.1521)	**1**	**0.1115 (0.1514)**	2	0.5141 (1.6944)	2	0.7481 (0.1545)
$\rho = 0.3$	**1**	**0.0788 (0.1161)**	2	0.1994 (0.0326)	**1**	**0.074 (0.116)**	3	0.5435 (1.6944)	3	0.3108 (0.0899)
$\rho = 0.5$	2	0.223 (0.0574)	**1**	**0.0968 (0.0259)**	2	0.2196 (0.0593)	3	0.5994 (1.6944)	2	0.2266 (0.0604)
$\rho = 0.7$	2	0.2337 (0.0538)	**1**	**0.1132 (0.0299)**	2	0.2356 (0.0532)	4	0.63 (1.6944)	3	0.265 (0.0548)
$\rho = 0.9$	2	0.0885 (0.0377)	**1**	**0.0595 (0.0255)**	2	0.0934 (0.0363)	3	0.6343 (1.6944)	2	0.0918 (0.0348)
Average r	**1.40**		2.65		1.69		1.83		2.10	

Figure 4 shows the average training and prediction time costs for ReScaL across the eight datasets, where each bar displays the mean and standard deviation of per-ρ time values for the dataset. Overall, on training sample size up to $30,844$, the results show that ReScaL's training overhead ranges from approximately average 1.60 seconds (DENSE4) to 9.50 seconds (DENSE6), with most datasets (6 out of 8) requiring less than 5 seconds in general. Prediction overhead is typically much lower, ranging from 0.007 seconds (DENSE6) to 0.090 seconds (DENSE5), with most datasets requiring less than 0.05 seconds per prediction.

Notably, this computational overhead is negligible compared to the cost of actual LLM pretraining runs, which can require weeks to months of training time on thousands of GPUs. For example, pretraining LLaMA-65B on 1.4T tokens of data took approximately 21 days on 2,048 A100 GPUs (Touvron et al., 2023). In contrast, ReScaL's overhead of a few seconds per prediction, even considering its training, enables rapid exploration of scaling decisions before committing to expensive

Table 18: Full ablation study results using RMSE. The formate follows Table 16.

Dataset	ReScaL		w/o dynamic α		w/o ensemble		w/o residual		w/o diverse fitting	
	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)	r	Mean (SD)
DENSE1										
$\rho = 0.1$	2	0.0021 (0.0027)	4	0.031 (0.0188)	**1**	**0.0008 (0.0004)**	2	0.0025 (0.0031)	3	0.0079 (0.0041)
$\rho = 0.2$	**1**	**0.0007 (0.0003)**	4	0.0431 (0.0194)	2	0.0023 (0.0021)	**1**	**0.0007 (0.0004)**	3	0.0114 (0.0065)
$\rho = 0.3$	**1**	**0.0008 (0.0005)**	4	0.0319 (0.0148)	2	0.0029 (0.0021)	**1**	**0.0008 (0.0005)**	3	0.0107 (0.0062)
$\rho = 0.4$	**1**	**0.0012 (0.0011)**	4	0.028 (0.0135)	2	0.0051 (0.0035)	**1**	**0.0013 (0.0012)**	3	0.0098 (0.0059)
$\rho = 0.5$	**1**	**0.0018 (0.0016)**	4	0.0247 (0.0121)	2	0.0055 (0.0034)	**1**	**0.0018 (0.0016)**	3	0.0096 (0.0061)
$\rho = 0.6$	**1**	**0.0024 (0.002)**	4	0.0224 (0.0114)	2	0.0049 (0.0034)	**1**	**0.0023 (0.002)**	3	0.0089 (0.0057)
$\rho = 0.7$	**1**	**0.0027 (0.0022)**	4	0.0189 (0.0099)	2	0.0037 (0.0029)	**1**	**0.0027 (0.0022)**	3	0.0079 (0.0051)
$\rho = 0.8$	**1**	**0.0026 (0.0021)**	3	0.0136 (0.0073)	**1**	**0.0029 (0.0023)**	**1**	**0.0025 (0.0021)**	2	0.0063 (0.0039)
$\rho = 0.9$	2	0.0023 (0.0019)	4	0.0086 (0.005)	**1**	**0.0011 (0.001)**	2	0.0022 (0.0019)	3	0.0047 (0.0026)
DENSE2										
$\rho = 0.1$	2	0.205 (0.806)	3	0.3541 (1.1772)	2	0.1707 (0.6289)	**1**	**0.0577 (0.0405)**	2	0.2334 (0.7554)
$\rho = 0.3$	**1**	**0.0156 (0.0162)**	**1**	**0.0168 (0.0138)**	**1**	**0.0151 (0.0159)**	2	0.0412 (0.0323)	**1**	**0.0155 (0.0145)**
$\rho = 0.5$	2	0.005 (0.0048)	**1**	**0.0034 (0.0042)**	2	0.0054 (0.005)	3	0.0227 (0.0206)	2	0.0048 (0.0046)
$\rho = 0.7$	**1**	**0.0173 (0.0883)**	**1**	**0.0156 (0.1113)**	**1**	**0.0144 (0.0606)**	2	0.0265 (0.0231)	**1**	**0.0136 (0.0563)**
$\rho = 0.9$	3	0.0044 (0.005)	**1**	**0.0015 (0.0032)**	2	0.0035 (0.0044)	4	0.0189 (0.0181)	2	0.0034 (0.0039)
DENSE3										
$\rho = 0.1$	**1**	**0.07 (0.0622)**	**1**	**0.0627 (0.0668)**	2	0.128 (0.0725)	**1**	**0.0674 (0.0535)**	3	0.2345 (0.2467)
$\rho = 0.2$	**1**	**0.0712 (0.159)**	2	0.2481 (0.4029)	3	0.3406 (0.0721)	**1**	**0.0746 (0.1689)**	**1**	**0.0993 (0.2659)**
$\rho = 0.3$	**1**	**0.0603 (0.1266)**	3	0.2472 (0.3482)	2	0.1156 (0.0349)	**1**	**0.0589 (0.1236)**	2	0.1029 (0.2818)
$\rho = 0.4$	**1**	**0.045 (0.0881)**	4	0.2101 (0.2444)	3	0.1368 (0.076)	**1**	**0.0467 (0.0922)**	2	0.0883 (0.1881)
$\rho = 0.5$	**1**	**0.033 (0.0557)**	2	0.0838 (0.155)	**1**	**0.0406 (0.0261)**	**1**	**0.033 (0.0552)**	2	0.0622 (0.1749)
$\rho = 0.6$	**1**	**0.0602 (0.0304)**	2	0.1103 (0.0671)	3	1.8051 (0.9242)	**1**	**0.0601 (0.0303)**	**1**	**0.0563 (0.1117)**
$\rho = 0.7$	3	0.0785 (0.0503)	4	0.1798 (0.1073)	**1**	**0.0163 (0.0369)**	3	0.0853 (0.0431)	2	0.0559 (0.0307)
$\rho = 0.8$	3	0.084 (0.0532)	4	0.1855 (0.1093)	**1**	**0.0073 (0.0158)**	3	0.0893 (0.0465)	2	0.0735 (0.0395)
$\rho = 0.9$	2	0.0921 (0.058)	4	0.1918 (0.1112)	**1**	**0.0215 (0.0125)**	2	0.1002 (0.0515)	3	0.1219 (0.0647)
DENSE4										
$\rho = 0.1$	**1**	**0.3677 (0.2379)**	**1**	**0.3315 (0.2429)**	**1**	**0.3688 (0.2372)**	2	1.8765 (1.3344)	**1**	**0.3599 (0.2384)**
$\rho = 0.9$	**1**	**0.0315 (0.0923)**	**1**	**0.0267 (0.1001)**	**1**	**0.0324 (0.0935)**	2	2.8282 (2.0293)	**1**	**0.0284 (0.0969)**
DENSE5										
$\rho = 0.2$	2	0.4509 (0.4411)	**1**	**0.2271 (0.2856)**	3	0.7972 (0.6162)	3	0.9212 (0.764)	2	0.4337 (0.4311)
$\rho = 0.4$	**1**	**0.1334 (0.1577)**	**1**	**0.1588 (0.1914)**	**1**	**0.1547 (0.2104)**	2	1.8574 (1.244)	**1**	**0.1302 (0.1556)**
$\rho = 0.6$	**1**	**0.1675 (0.2047)**	2	0.2009 (0.2433)	**1**	**0.1392 (0.1868)**	3	1.9199 (1.2707)	**1**	**0.1588 (0.1959)**
$\rho = 0.8$	**1**	**0.1224 (0.1544)**	**1**	**0.1277 (0.1616)**	**1**	**0.1331 (0.1737)**	2	1.9857 (1.2982)	**1**	**0.1222 (0.1565)**
DENSE6										
$\rho = 0.1$	2	0.0353 (0.0575)	3	0.1139 (0.1133)	2	0.0465 (0.0628)	2	0.0354 (0.0576)	**1**	**0.0168 (0.0166)**
$\rho = 0.2$	2	0.0336 (0.0555)	4	0.1239 (0.1182)	3	0.0455 (0.0312)	2	0.0335 (0.0554)	**1**	**0.0205 (0.0338)**
$\rho = 0.3$	**1**	**0.0303 (0.0512)**	3	0.117 (0.1148)	2	0.0593 (0.0509)	**1**	**0.0303 (0.0513)**	**1**	**0.0297 (0.0505)**
$\rho = 0.4$	**1**	**0.0299 (0.0507)**	3	0.1086 (0.1105)	2	0.0658 (0.0682)	**1**	**0.0299 (0.0507)**	4	0.1683 (0.1379)
$\rho = 0.5$	2	0.0262 (0.0452)	2	0.0258 (0.016)	**1**	**0.0181 (0.014)**	2	0.0262 (0.0451)	2	0.0256 (0.0441)
$\rho = 0.6$	2	0.0246 (0.0424)	2	0.0199 (0.0044)	**1**	**0.0087 (0.0131)**	2	0.0246 (0.0424)	2	0.0219 (0.0371)
$\rho = 0.7$	2	0.0228 (0.039)	2	0.0182 (0.0037)	**1**	**0.0062 (0.0068)**	2	0.0228 (0.039)	2	0.021 (0.035)
$\rho = 0.8$	3	0.0228 (0.039)	2	0.0168 (0.0136)	**1**	**0.0073 (0.0103)**	3	0.0228 (0.0391)	3	0.022 (0.0374)
$\rho = 0.9$	**1**	**0.0229 (0.0392)**	**1**	**0.017 (0.0198)**	2	0.1652 (0.2858)	**1**	**0.0229 (0.0393)**	**1**	**0.0225 (0.0385)**
MOE1										
$\rho = 0.1$	**1**	**0.002 (0.0021)**	3	0.0253 (0.0139)	**1**	**0.0022 (0.0024)**	2	0.013 (0.0078)	**1**	**0.0023 (0.0022)**
$\rho = 0.2$	**1**	**0.0031 (0.0034)**	3	0.0184 (0.0131)	**1**	**0.0026 (0.0027)**	2	0.0156 (0.0085)	**1**	**0.0029 (0.0031)**
$\rho = 0.3$	**1**	**0.0033 (0.0035)**	4	0.3067 (0.0568)	**1**	**0.0027 (0.0026)**	2	0.0099 (0.0067)	3	0.1004 (0.0325)
$\rho = 0.4$	2	0.0089 (0.0077)	3	0.0489 (0.0173)	2	0.0085 (0.0074)	**1**	**0.0012 (0.0015)**	3	0.0497 (0.0175)
$\rho = 0.5$	**1**	**0.0026 (0.0034)**	4	0.0162 (0.0119)	2	0.0037 (0.0049)	**1**	**0.0028 (0.0031)**	3	0.0117 (0.0101)
$\rho = 0.6$	**1**	**0.0012 (0.0015)**	4	0.0233 (0.0108)	2	0.0058 (0.0068)	**1**	**0.0012 (0.0015)**	3	0.0149 (0.0085)
$\rho = 0.7$	**1**	**0.0014 (0.0018)**	4	0.0208 (0.0101)	2	0.0076 (0.0056)	**1**	**0.0013 (0.0017)**	3	0.014 (0.0081)
$\rho = 0.8$	**1**	**0.0013 (0.0017)**	3	0.011 (0.0073)	2	0.0058 (0.0053)	**1**	**0.0013 (0.0016)**	3	0.0107 (0.0072)
$\rho = 0.9$	**1**	**0.0012 (0.0015)**	2	0.0058 (0.0051)	2	0.0049 (0.0048)	**1**	**0.0012 (0.0015)**	2	0.0058 (0.0051)
MOE2										
$\rho = 0.1$	**1**	**0.035 (0.1301)**	2	0.2966 (0.2505)	**1**	**0.0345 (0.1273)**	4	3.0247 (15.1545)	3	0.5826 (0.3245)
$\rho = 0.3$	**1**	**0.0192 (0.0732)**	2	0.0408 (0.0129)	**1**	**0.0184 (0.0726)**	4	3.0559 (15.254)	3	0.1044 (0.0867)
$\rho = 0.5$	2	0.0529 (0.0377)	**1**	**0.01 (0.005)**	2	0.0516 (0.0398)	3	3.1198 (15.443)	2	0.0549 (0.0438)
$\rho = 0.7$	2	0.0574 (0.0369)	**1**	**0.0137 (0.0101)**	2	0.0582 (0.0371)	4	3.1573 (15.5461)	3	0.0731 (0.0429)
$\rho = 0.9$	2	0.0092 (0.0127)	**1**	**0.0042 (0.0029)**	2	0.01 (0.0116)	3	3.1628 (15.5609)	2	0.0096 (0.0115)
Average r	**1.44**		2.58		1.65		1.87		2.13	

full-scale pretraining. For example, on DENSE1 at $\rho = 0.2$, ReScaL reduces MAPE from 0.0764 for the strongest baseline to 0.0135, as shown in Section 4.3. Such accuracy gains outweigh the comparatively small prediction overhead, especially when pilot data are limited and reliable scaling decisions are most difficult.

C.7 PILOT-COMPUTE COST EFFECTIVENESS

We additionally examine whether the accuracy gains offset the compute required to acquire pilot runs.

Method. We define a pilot job's approximate cost as $c_j = 6N_jD_j$ and charge every distinct job once. Rows corresponding to multiple checkpoints or validation domains of the same run are collapsed when the source data provide a reliable identifier; otherwise, each row is conservatively

Table 19: Pilot-compute break-even against the strongest baseline. "Match" is the fraction of cases in which ReScaL reaches the baseline MAPE; "≤Cost" is the fraction that does so with no greater pilot FLOPs; "MedSave" is median compute saving among matched cases.

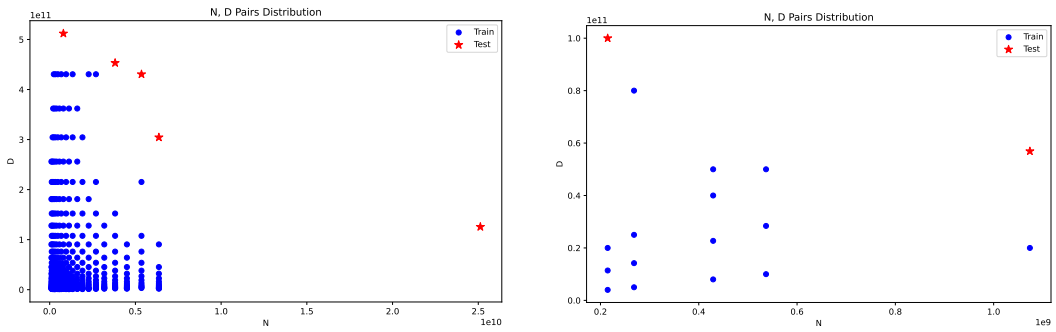
System	Match ↑	≤Cost ↑	MedSave ↑
DENSE1	100.0%	100.0%	98.0%
DENSE2	100.0%	80.0%	0.0%
DENSE3	100.0%	88.9%	97.5%
DENSE4	100.0%	50.0%	−142.9%
DENSE5	100.0%	75.0%	28.5%
DENSE6	100.0%	66.7%	81.5%
MOE1	100.0%	100.0%	83.3%
MOE2	80.0%	80.0%	93.3%
All 52 cases	98.1%	84.6%	82.1%

charged as a separate job. For each of the 52 cases, break-even is the least expensive evaluated ReScaL budget whose MAPE is no worse than the strongest baseline at that case's reported budget. We report whether such a point is reached, whether it uses no more compute, and the median saving $1 - C_{\texttt{ReScaL}}/C_{\text{baseline}}$ among matched cases.

Results. As shown in Table 19, across all 52 cases, ReScaL reaches or exceeds the strongest baseline's accuracy in 98.1% of cases and does so with no greater pilot compute in 84.6%; 69.2% achieve a strict saving. Among matched cases, the median compute saving is 82.1%. The main exception is DENSE4, whose three scale settings and coarse discrete jobs yield only two evaluated cases and a negative median saving. Hence, the accuracy improvement generally offsets the cost of acquiring broader pilot coverage, although the benefit depends on job granularity and dataset structure.

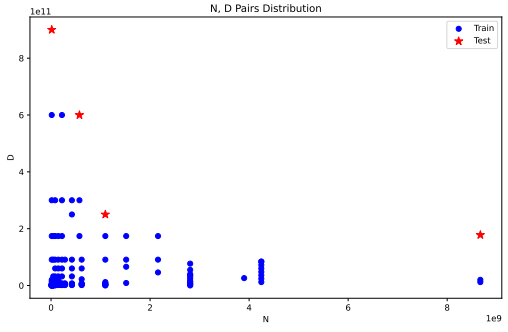
D REPRODUCIBILITY

All experiments use fixed random seeds for reproducibility. The datasets, scripts, and full experimental results are available in the anonymous repository at https://anonymous.4open.science/r/ReScaL-8D70.

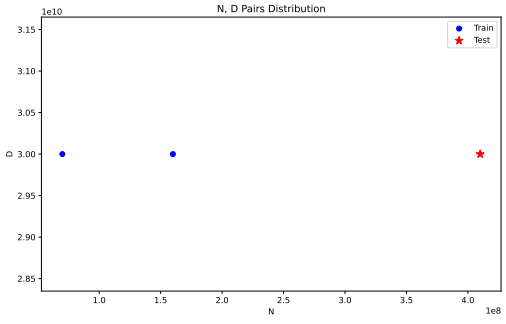


(a) DENSE1

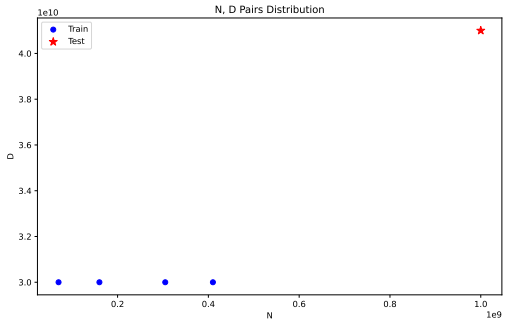
(b) DENSE2



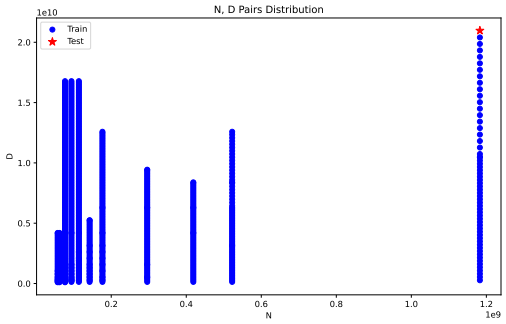
(c) DENSE3



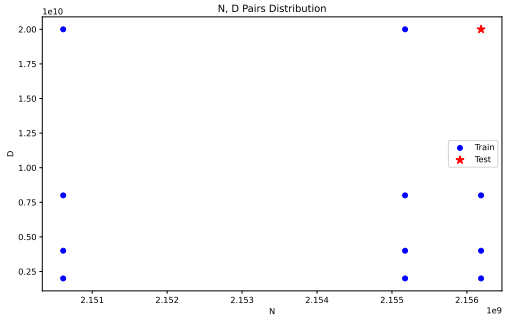
(d) DENSE4



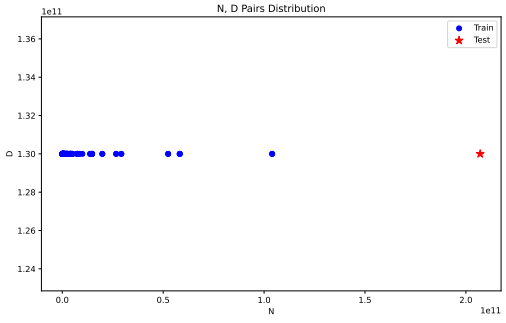
(e) DENSE5



(f) DENSE6



(g) MOE1



(h) MOE2

Figure 5: Scale settings distribution for all eight datasets. Potential training data (blue circles) are concentrated in the lower-left region of the (N, D) space, representing the cheaper, smaller-scale pilot settings. Testing data (red stars) are distributed at the edges and upper-right regions, representing the expensive, large-scale settings used for evaluation.

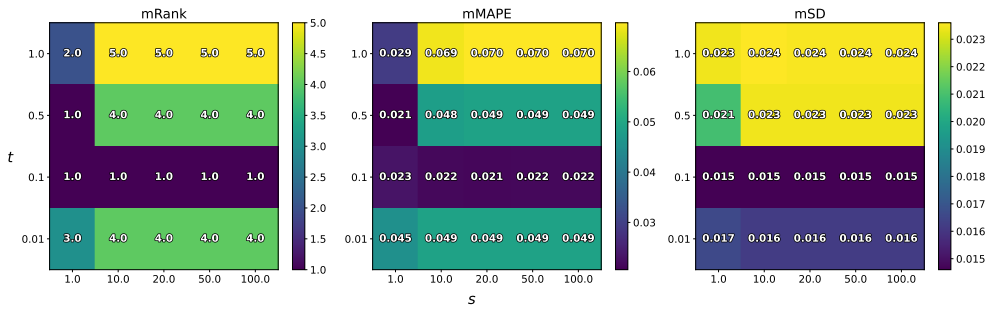


Figure 6: Sensitivity heatmap for DENSE1 showing mean MAPE across all pilot budget sizes for each (t, s) combination.

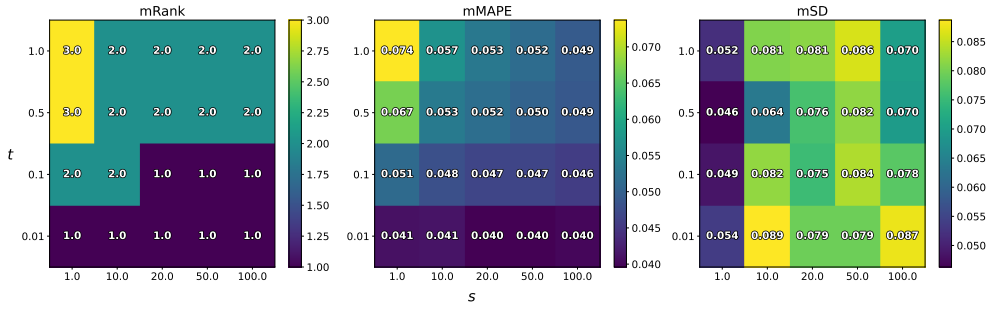


Figure 7: Sensitivity heatmap for DENSE2 showing mean MAPE across all pilot budget sizes for each (t, s) combination.

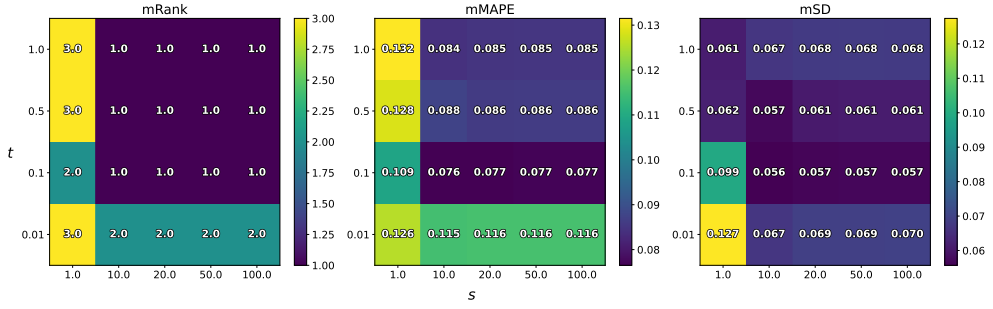


Figure 8: Sensitivity heatmap for DENSE3 showing mean MAPE across all pilot budget sizes for each (t, s) combination.

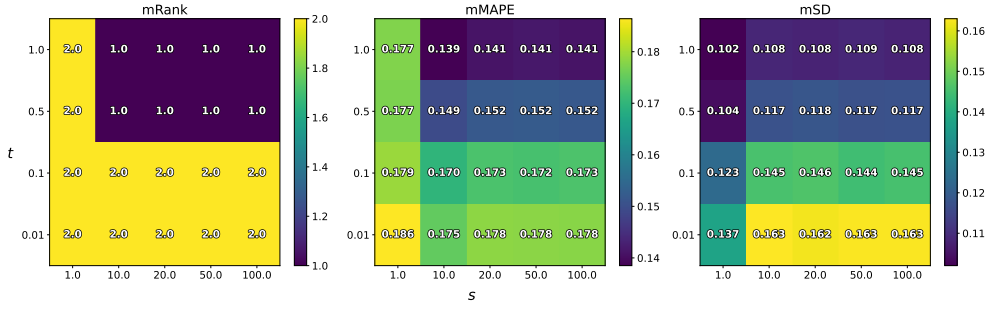


Figure 9: Sensitivity heatmap for DENSE4 showing mean MAPE across all pilot budget sizes for each (t, s) combination.

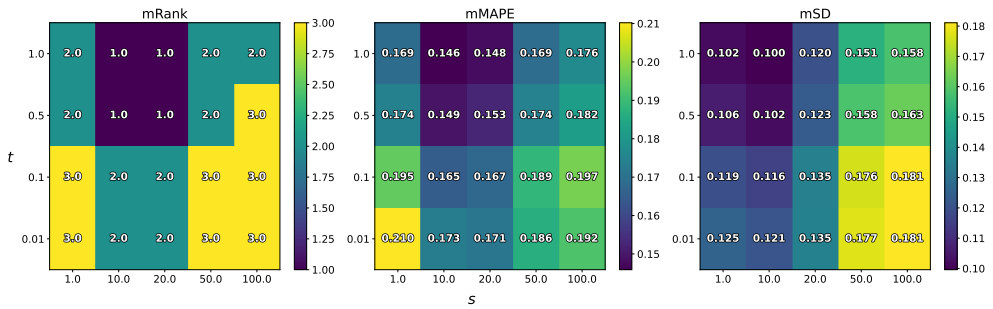


Figure 10: Sensitivity heatmap for DENSE5 showing mean MAPE across all pilot budget sizes for each (t, s) combination.

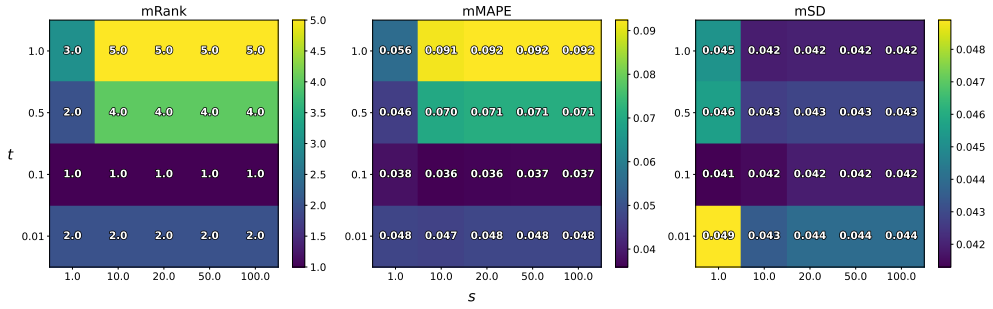


Figure 11: Sensitivity heatmap for DENSE6 showing mean MAPE across all pilot budget sizes for each (t, s) combination.

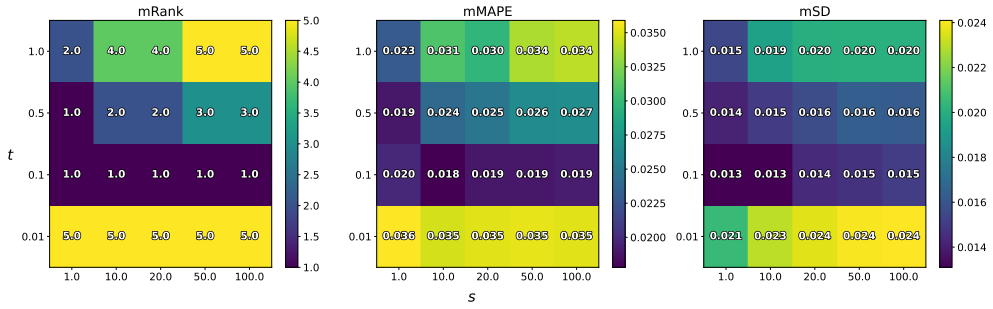


Figure 12: Sensitivity heatmap for MOE1 showing mean MAPE across all pilot budget sizes for each (t, s) combination.

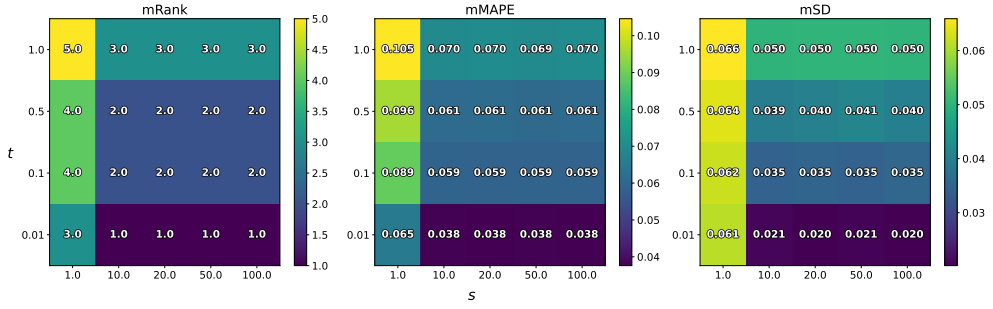


Figure 13: Sensitivity heatmap for MOE2 showing mean MAPE across all pilot budget sizes for each (t, s) combination.