BEYOND INTERACTION CAPACITY: ESTIMATOR SCALING WITH RECURSIVE MODELS FOR CTR PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Click-Through Rate (CTR) prediction, a core task in recommendation and advertising systems, relies on modeling interactions among sparse categorical features. Explicit cross networks are a central paradigm for CTR prediction, and recent progress has largely come from increasing the interaction capacity of a single predictor through deeper cross networks and more expressive cross operators. We revisit whether continually increasing interaction capacity remains the most effective way to improve predictive performance, and find that its benefits quickly exhibit diminishing returns even as capacity continues to grow. This motivates a complementary scaling direction that we call *estimator scaling*, where additional resources are used to incorporate multiple related estimators rather than only enlarging a single predictor. Through theoretical analysis and diagnostic experiments, we show that the gains from estimator scaling are governed by the amount of non-shared predictive variation available across estimators. However, exploiting this variation naively can be expensive: independently trained models provide substantial estimator diversity but require deployment cost to grow with ensemble size. This motivates a parameter-efficient realization of estimator scaling that can incorporate diversity from multiple estimator sources without maintaining multiple full models. Building on this view, we introduce **REC**ursive **A**veraged **P**redictor (RECAP), a parameter-efficient recursive CTR model that operationalizes estimator scaling at three levels: distillation across independently trained models, exponential moving averaging over training trajectories, and aggregation over inference-time routes within a weight-shared recursive backbone. This design improves predictive performance without requiring parameter growth proportional to the ensemble size. Experiments across multiple CTR benchmarks show that RECAP approaches the accuracy of a five-model ensemble of a strong cross-network baseline at one fifth of its deployed parameters, while doubling the baseline's interaction capacity recovers only a third of that gain. Overall, these results establish new state-of-the-art predictive performance on standard benchmarks, while placing the proposed approach on a favorable performance–parameter Pareto frontier. The code will be available at https://anonymous.4open.science/r/RECAP_CTR.

1 INTRODUCTION

Click-through rate (CTR) prediction is a central component of modern recommendation and advertising systems, where accurate ranking depends critically on modeling interactions among sparse categorical and numerical features Zhu et al. (2022); Wang et al. (2017). A long line of work has therefore improved CTR models by increasing the *interaction capacity* of a single predictor through deeper cross networks, and richer interaction operators Wang et al. (2017; 2021a); Guo et al. (2017); Zhu et al. (2023). In explicit cross networks, this capacity can be increased along several architectural dimensions, most directly by making interaction networks deeper or by combining multiple interaction branches with complementary operators. Deep & Cross Networks (DCNs) Wang et al. (2017) and their derivatives exemplify this paradigm, explicitly scaling the complexity of feature interactions represented by the model. Recent architectures such as Quadratic Neural Networks (QNN) and Fusing Cross Networks (FCN) push this direction further by constructing interactions whose order grows exponentially with depth Li et al. (2025; 2026). This progression raises a natural question: *once a sufficiently expressive interaction model is available, is continually increasing interaction capacity still the most effective way to improve CTR prediction?*

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.

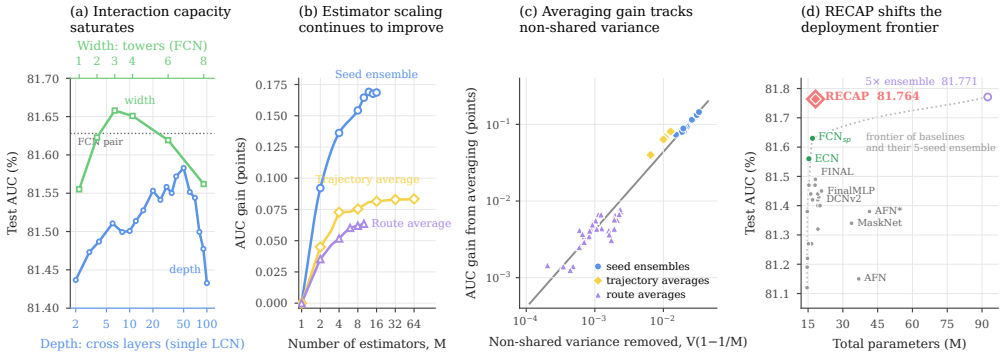

Figure 1: **Diagnostic analysis of interaction-capacity and estimator scaling on Criteo.** (a) Increasing interaction depth and breadth exhibits diminishing returns. (b) Additional estimators provide further gains, with independent runs outperforming trajectories and shared-model routes. (c) Averaging gains track the non-shared predictive variance available to aggregation (Section 3). (d) RECAP captures much of the ensemble gain at approximately a single-model deployment budget. Diagnosis results on additional datasets can be found in Appendix A.

We test this assumption by scaling interaction capacity along two concrete architectural axes: depth, through additional cross layers, and breadth, through additional interaction branches. As shown in Figure 1(a), increasing interaction depth or adding interaction branches initially improves performance, but the marginal benefit quickly shrinks as interaction capacity increases. This diminishing-return regime raises a second question: if additional resources are no longer best spent increasing the capacity of a single predictor, how else can they improve prediction?

In Section 3, we conduct a theoretical analysis that points to a complementary scaling direction that we call **estimator scaling**. Rather than enlarging the function class of a single predictor, estimator scaling uses additional training or inference resources to incorporate multiple estimators of the same predictive function. Prior work has theoretically analyzed the underlying benefit of averaging less-correlated estimators Wood et al. (2023); our focus is on its role as a scaling principle for CTR models, where recent progress has predominantly come from increasing feature-interaction capacity. We show that its benefit is governed by the non-shared predictive variation exposed by different estimator sources which offer distinct diversity–deployment-cost trade-offs. This yields a simple scaling principle: gains are larger when additional estimators expose more non-shared variation and diminish as their predictions become increasingly correlated.

As shown in Figure 1(b), our diagnostic experiments support this prediction across three distinct sources of estimators: averaging independently trained models, averaging predictors sampled along a single training trajectory, and averaging alternative inference-time routes through a shared model. Despite their different origins, the improvement from averaging closely tracks the amount of *non-shared predictive variation* available along each estimator axis (Figure 1(c)). Additional diagnosis results on other datasets can be found in Appendix A. These results reveal both the promise and the challenge of estimator scaling: large gains arise from averaging multiple estimators but maintaining individual copies of the models would incur substantial deployment cost.

To address these challenges, we introduce **RECursive Averaged Predictor (RECAP)**, a parameter-efficient recursive CTR model that realizes estimator scaling at three levels. RECAP uses a weight-shared recursive interaction backbone, allowing interaction computation to increase without proportional growth in depth-specific parameters. On top of this backbone, RECAP incorporates information across *independent training runs* through ensemble distillation, across the *training trajectory* through exponential moving weight averaging, and across *inference-time routes* through route aggregation. As summarized in Figure 1, these mechanisms expose progressively different sources of estimator variation while keeping the deployed model close to a single-model parameter budget, thereby placing RECAP on a favorable deployment frontier.

Our main contributions are as follows:

- **Estimator scaling for CTR prediction.** We identify estimator scaling as a complementary axis to interaction-capacity scaling and show that estimator scaling continues to provide gains after additional interaction depth and branch capacity exhibit diminishing returns.

- **RECAP: parameter-efficient multi-level estimator scaling.** We introduce a recursive CTR model that combines parameter sharing with estimator scaling across training trajectories, independent runs, and inference-time routes through exponential moving weight averaging, ensemble distillation, and route aggregation.
- **Improved deployment efficiency.** Across multiple CTR benchmarks, RECAP improves the performance–parameter trade-off over strong cross-network baselines and approaches the performance of substantially larger multi-model ensembles while retaining approximately a single-model deployment footprint.

2 Related Work

Feature Interaction and Cross-Network Scaling: A central line of CTR research improves predictive accuracy by scaling the learned feature interactions. Early models combine factorization-based interactions with deep networks Guo et al. (2017), while DCN explicitly constructs bounded-degree feature crosses through stacked cross layers Wang et al. (2017). Subsequent work increases or adapts this interaction capacity in different ways: xDeepFM introduces explicit vector-wise high-order interactions Lian et al. (2018), AutoInt uses stacked self-attention to model increasingly high-order combinations Song et al. (2019), and DCNv2 improves the expressiveness of cross networks while retaining computational efficiency through low-rank parameterizations Wang et al. (2021a). Most recently, FCN and QNN use exponential cross networks to explicitly represent interactions spanning a broad range of orders Li et al. (2025; 2026). Collectively, these methods primarily improve CTR prediction by enlarging or selectively allocating the *interaction capacity* of a predictor. In contrast, we study what should scale once further increases along this axis yield diminishing predictive returns.

Ensembling, Averaging, and Distillation: Our work is also related to methods that combine multiple learned predictors. Deep ensembles improve prediction by aggregating independently trained models Lakshminarayanan et al. (2017), while trajectory-based methods such as stochastic weight averaging (SWA) aggregate solutions encountered during optimization Izmailov et al. (2019). Model Soups and DiWA further study weight averaging across independently trained or fine-tuned models, highlighting the importance of model diversity for successful aggregation Wortsman et al. (2022); Rame et al. (2022). The role of diversity in ensemble performance has also been studied theoretically Wood et al. (2023). Our contribution is not a new averaging operator in isolation. Instead, we formulate *estimator scaling* as a complementary scaling axis and study its gains, saturation, and cost across independent runs, optimization trajectories, and inference-time routes within a common variance-based framework

Recursive and Parameter-Efficient Computation: A complementary literature seeks to increase computation or interaction complexity without proportional parameter growth. Low-rank cross networks reduce the parameter cost of expressive interaction operators Wang et al. (2021a), while recent LoopCTR introduces loop scaling for CTR prediction, recursively reusing shared layers to decouple additional computation from model size Tang et al. (2026). RECAP shares the principle of recursive parameter reuse but serves a different purpose: recursion provides a compact substrate on which multiple related estimators can be realized. We combine this backbone with cross-run distillation, trajectory averaging, and inference-time route aggregation, using parameter sharing to enable parameter-efficient estimator scaling.

3 From Interaction Capacity to Estimator Scaling

In this section, we formalize why estimator scaling can provide a complementary source of improvement to increasing the interaction capacity of a single predictor, characterize when such gains are available, and analyze how they combine across different estimator sources.

Consider a CTR dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i contains categorical and numerical features and $y_i \in \{0, 1\}$ denotes a click. A model with parameters θ produces a logit z and click probability $p_\theta(x) = \sigma(z)$, and is trained using binary cross-entropy, which is defined as:

$$\ell(y, z) = \log(1 + \exp(z)) - yz. \tag{1}$$

A fixed model family can yield multiple related predictors through independent training runs, checkpoints along an optimization trajectory, or alternative inference-time routes. We refer to each such

predictor as an *estimator*, and denote these three estimator sources by S, T, and R, respectively. Given M estimators with logits $\{Z_m(x)\}_{m=1}^M$, their mean-logit prediction is

$$\bar{Z}_M(x) = \frac{1}{M}\sum_{m=1}^{M} Z_m(x). \tag{2}$$

Two Complementary Scaling Axes As motivated in the introduction, **interaction-capacity scaling** increases the expressive capacity of a single predictor, whereas **estimator scaling** increases the number of estimators incorporated into its prediction while largely holding the underlying model family fixed. We next formalize why these two axes can provide distinct sources of improvement.

Let c index the interaction capacity of a model family, let $Z_c(x)$ denote the logit of a randomly drawn estimator from that family, and let $\mu_c(x) = \mathbb{E}[Z_c(x) \mid x]$ denote the estimator-family mean. For M estimators, define $\bar{Z}_{c,M}(x) = \frac{1}{M}\sum_{m=1}^{M} Z_{c,m}(x)$, and let $\mathcal{L}(c, M)$ denote its expected binary cross-entropy over the data distribution and estimator randomness. We similarly define $\mathcal{L}(\mu_c) = \mathbb{E}_{x,y}[\ell(y, \mu_c(x))]$. A second-order expansion of Eq. 1 around $\mu_c(x)$ gives

$$\mathcal{L}(c, M) \approx \mathcal{L}(\mu_c) + \frac{1}{2}\mathbb{E}_x\left[\sigma(\mu_c(x))\left(1 - \sigma(\mu_c(x))\right)\mathrm{Var}(\bar{Z}_{c,M}(x) \mid x)\right]. \tag{3}$$

Equation 3 separates two mechanisms for improvement. Increasing interaction capacity can improve the mean predictor μ_c and thereby reduce $\mathcal{L}(\mu_c)$, whereas estimator scaling reduces variation around that mean through aggregation. Thus, even when additional interaction capacity yields little further improvement in the mean predictor, estimator scaling can still reduce expected loss whenever reducible estimator variation remains.

Non-Shared Variation Governs Estimator Scaling We next characterize how much estimator variation can be removed by aggregation. For a fixed input x and interaction capacity c, suppose the M estimators are exchangeable (i.e their joint probability distribution is invariant under any permutation of their indices) , with conditional logit variance $\tau_c^2(x)$ and common pairwise correlation $\rho_c(x)$. The variance of their mean is

$$\mathrm{Var}(\bar{Z}_{c,M}(x) \mid x) = \tau_c^2(x)\left(\rho_c(x) + \frac{1 - \rho_c(x)}{M}\right). \tag{4}$$

Relative to a single estimator, the variance removed by averaging is therefore

$$V_{\mathrm{ns}}(c, M; x) = \tau_c^2(x)\left(1 - \rho_c(x)\right)\left(1 - \frac{1}{M}\right), \tag{5}$$

which we call the *non-shared*, or reducible, predictive variation. Combining Eqs. 3 and 5, the reduction in expected BCE loss is locally

$$\Delta\mathcal{L}(c, M) \approx \frac{1}{2}\mathbb{E}_x\left[\sigma(\mu_c(x))\left(1 - \sigma(\mu_c(x))\right)V_{\mathrm{ns}}(c, M; x)\right]. \tag{6}$$

Thus, BCE improvement is approximately proportional to the reducible predictive variance, up to the local curvature of the loss. When the estimator families have similar mean predictions, this curvature factor is approximately shared, so differences in averaging gain are primarily determined by V_{ns}. Further derivations of the relationship between non-shared variance and estimator scaling are provided in Appendix B.

Combining Estimator Axes Different estimator sources may expose overlapping predictive variation, so their gains need not be additive. Let A and B denote two estimator axes and let $Z_{A,B}(x)$ denote the corresponding random logit for a fixed input. By the law of total variance,

$$\mathrm{Var}(Z_{A,B} \mid x) = \underbrace{\mathbb{E}_B\left[\mathrm{Var}_A(Z_{A,B} \mid B, x)\right]}_{V_A(x)} + \underbrace{\mathrm{Var}_B\left[\mathbb{E}_A(Z_{A,B} \mid B, x)\right]}_{V_{B|A}(x)}. \tag{7}$$

Here $V_A(x)$ is the variation accessible to averaging along axis A, whereas $V_{B|A}(x)$ is the residual variation remaining after averaging over A. Up to the same local curvature weighting as in Eq. 6, the standalone reduction in expected loss from an estimator axis is therefore governed by its accessible variation, while the marginal improvement from adding another axis is governed by the residual variation not already removed. This prediction is evaluated empirically in Section 5.2; the full three-axis decomposition for independent runs, trajectories, and routes is given in Appendix C

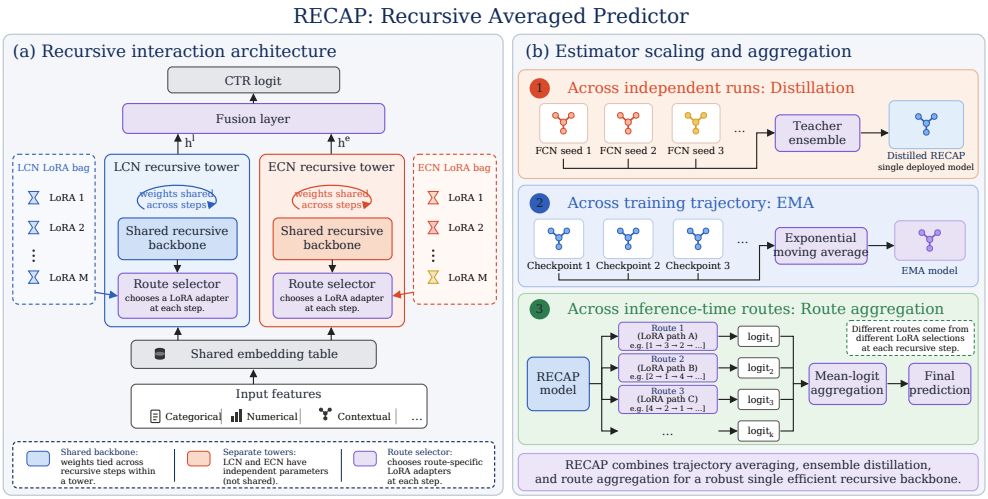

Figure 2: **RECAP architecture and multi-level estimator scaling.** (a) RECAP builds on the linear and exponential cross-network (LCN and ECN) operators of FCN (Li et al., 2026), which we organize as two recursive interaction towers. Each tower shares its core cross-network parameters across recursive steps, while a route selector chooses lightweight LoRA adaptations from a tower-specific expert bank. The two tower representations are fused to produce the CTR logit. (b) RECAP incorporates estimators at three levels: cross-run information is transferred through ensemble distillation, within-run information is aggregated through exponential moving averaging, and multiple recursive routes are combined through mean-logit aggregation at inference.

4 RECAP: RECursive Averaged Predictor

We now introduce **REC**ursive **A**veraged **P**redictor (RECAP), a parameter-efficient realization of the estimator-scaling principle in Section 3. The theory identifies non-shared and residual predictive variation as the quantities governing the benefit of additional estimator axes; RECAP focuses on realizing three such axes: independent runs, optimization trajectories, and inference-time routes, without deploying a separate full model for each. These axes connect to the output-space analysis in different ways: ensemble distillation compresses a cross-run mean-logit predictor into a single model, EMA provides a parameter-space approximation to trajectory prediction averaging, and route aggregation performs the mean-logit averaging analyzed directly in Section 3.

4.1 Parameter-Efficient Recursive Interaction Backbone

The estimator-scaling analysis does not prescribe a particular CTR backbone; it specifies that useful estimator axes should expose non-shared predictive variation at acceptable cost. We therefore use recursion for two complementary purposes: to increase interaction computation through parameter sharing, and to expose an additional inference-time estimator axis through alternative computation routes. We instantiate RECAP using the Fusing Cross Network (FCN) architecture (Li et al., 2026), a strong recent explicit-interaction model for CTR prediction. FCN combines a linear cross network (LCN), whose interaction order grows progressively with depth, with an exponential cross network (ECN), which constructs higher-order interactions more rapidly. Given input features x, RECAP first maps sparse categorical fields and numerical features to a shared representation h_0, which is then processed by recursive LCN and ECN towers. The two towers use independent parameters, while the core interaction weights within each tower are shared across recursive steps. Lightweight route-specific adaptations allow different adapter sequences to define multiple related predictors over the same shared backbone, providing a third source of estimator variation that can be aggregated at inference without maintaining separate full models. Thus, recursion serves both as a parameter-efficient computational substrate and as the mechanism that enables RECAP's route-scaling axis. Additionally, the broader estimator-scaling principle is not tied to these particular interaction operators; Section 5.3 further examines estimator scaling independently of the recursive architecture.

Let $q \in \{\mathrm{L}, \mathrm{E}\}$ denote a tower and let h_t^q be its hidden representation after t recursive steps. RECAP updates

$$h_{t+1}^q = F_q\left(h_t^q, x; W_q + \Delta W_{q,r_t}\right), \tag{8}$$

where W_q is the shared interaction operator and $\Delta W_{q,r_t}$ is a lightweight route-dependent adaptation selected at step t.

Rather than assigning a full independent parameter matrix to every recursive step, RECAP parameterizes each adaptation using a low-rank residual,

$$\Delta W_{q,r} = A_{q,r} B_{q,r}^{\top}, \qquad \operatorname{rank}(\Delta W_{q,r}) \le r_{\mathrm{LoRA}}. \tag{9}$$

Each tower maintains a small bank of such residuals, and a route selector determines which adapter is applied at each recursive step. A route is therefore defined by the sequence $\tau_q = (r_1, r_2, \ldots, r_T)$, while all routes continue to reuse the same core operator W_q.

After T recursive steps, the two tower representations are fused, $h_{\mathrm{fuse}} = \mathrm{Fuse}\left(h_T^{\mathrm{L}}, h_T^{\mathrm{E}}\right)$, and a prediction head produces the CTR logit $z_\theta(x) = g(h_{\mathrm{fuse}})$. This recursive parameterization decouples interaction computation from parameter depth: increasing the number of recursive steps increases computation while reusing the same backbone parameters. The lightweight route residuals additionally allow RECAP to expose multiple related predictors without storing multiple models.

4.2 Estimator Scaling at Three Levels

RECAP applies the estimator-scaling principle of Section 3 at three complementary levels.

Training-trajectory averaging: During optimization, successive checkpoints provide different estimators from the same training run. RECAP maintains an exponential moving average (EMA) of the model parameters,

$$\bar{\theta}_t = \beta \bar{\theta}_{t-1} + (1-\beta)\theta_t, \tag{10}$$

where β controls the averaging horizon. The EMA parameters are used for validation, checkpoint selection, and final inference. This incorporates information across the optimization trajectory without adding deployed parameters or inference-time computation.

EMA averages parameters rather than logits directly. Its connection to the output-space analysis of Section 3 follows under a local linearization of the predictor. If $\bar{\theta} = \sum_t \alpha_t \theta_t$ denotes the EMA parameters, then for checkpoints lying in a locally approximately linear region,

$$z_{\bar{\theta}}(x) \approx \sum_t \alpha_t z_{\theta_t}(x). \tag{11}$$

Thus, EMA can be viewed as a parameter-efficient approximation to trajectory-level prediction averaging. This approximation need not hold globally, so we empirically evaluate trajectory scaling rather than treating it as an exact consequence of the logit-averaging analysis.

Cross-run averaging through distillation: Independent training runs expose substantially more non-shared predictive variation, but deploying an ensemble of M complete CTR models multiplies storage and inference cost. RECAP instead transfers this cross-run aggregation into a single model through ensemble distillation.

Let $z^{(1)}(x), \ldots, z^{(M)}(x)$ denote the logits of independently trained teacher models. We form the teacher aggregate

$$\bar{z}_{\mathrm{teach}}(x) = \frac{1}{M} \sum_{m=1}^{M} z^{(m)}(x), \tag{12}$$

and train RECAP to match the corresponding soft prediction in addition to the ground-truth label. The training objective is

$$\mathcal{L}_{\mathrm{train}} = \mathcal{L}_{\mathrm{sup}} + \lambda_{\mathrm{KD}} \mathcal{L}_{\mathrm{KD}}, \tag{13}$$

where

$$\mathcal{L}_{\mathrm{KD}} = \mathrm{BCE}\left(\sigma(\bar{z}_{\mathrm{teach}}), z_\theta(x)\right). \tag{14}$$

Thus, distillation transfers the benefit of expensive cross-run averaging into a single deployed parameterization.

This objective has a direct connection to the mean-logit predictor analyzed in Section 3. For a fixed input, BCE with soft target $q = \sigma(\bar{z}_{\mathrm{teach}})$ is minimized when $\sigma(z_\theta) = q$, or equivalently $z_\theta = \bar{z}_{\mathrm{teach}}$. Thus, with sufficient student capacity and optimization, distillation compresses the cross-run mean-logit aggregate into a single deployed predictor. In practice this equality is approximate, and the remaining compression error is measured empirically.

Table 1: **Performance comparison of different deep CTR models.** Typically, CTR researchers consider an improvement of *0.001 (0.1%)* in Logloss and AUC to be practically meaningful Zhu et al. (2021); Li et al. (2026). We conduct a two-tailed t-test over five independent runs to assess the statistical significance between our models and the best baseline (*: p<0.01). Best results are highlighted in **bold**, second-best are underlined.

Models	Avazu		Criteo		ML-1M		KDD12		KKBox	
	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑
DNN Covington et al. (2016)	0.3721	79.27	0.4380	81.40	0.3100	90.30	0.1502	80.52	0.4811	85.01
PNN Qu et al. (2016)	0.3712	79.44	0.4378	81.42	0.3070	90.42	0.1504	80.47	0.4793	85.15
Wide & Deep Cheng et al. (2016)	0.3720	79.29	0.4376	81.42	0.3056	90.45	0.1504	80.48	0.4852	85.04
DeepFM Guo et al. (2017)	0.3719	79.30	0.4375	81.43	0.3073	90.51	0.1501	80.60	0.4785	85.31
DCNv1 Wang et al. (2017)	0.3719	79.31	0.4376	81.44	0.3156	90.38	0.1501	80.59	0.4766	85.31
xDeepFM Lian et al. (2018)	0.3718	79.33	0.4376	81.43	0.3054	90.47	0.1501	80.62	0.4772	85.35
AutoInt Song et al. (2019)	0.3746	79.02	0.4390	81.32	0.3112	90.45	0.1502	80.57	0.4773	85.34
AFN Cheng et al. (2020)	0.3726	79.29	0.4384	81.38	0.3048	90.53	0.1499	80.70	0.4842	84.89
DCNv2 Wang et al. (2021a)	0.3718	79.31	0.4376	81.45	0.3098	90.56	0.1502	80.59	0.4787	85.31
EDCN Chen et al. (2021)	0.3716	79.35	0.4386	81.36	0.3073	90.48	0.1501	80.62	0.4952	85.27
MaskNet Wang et al. (2021b)	0.3711	79.43	0.4387	81.34	0.3080	90.34	0.1498	80.79	0.5003	84.79
EulerNet Tian et al. (2023)	0.3723	79.22	0.4379	81.47	0.3050	90.44	0.1498	80.78	0.4922	84.27
FinalMLP Mao et al. (2023)	0.3718	79.35	0.4373	81.45	0.3058	90.52	0.1497	80.78	0.4822	85.10
FINAL Zhu et al. (2023)	0.3712	79.41	0.4371	81.49	0.3035	90.53	0.1498	80.74	0.4800	85.14
RFM Tian et al. (2024)	0.3723	79.24	0.4374	81.47	0.3048	90.51	0.1506	80.73	0.4853	84.70
QNN Li et al. (2025)	0.3712	79.47	0.4358	81.63	0.2960	90.87	0.1506	80.82	0.4730	85.76
FCN Li et al. (2026)	0.3702	79.66	0.4359	81.62	0.3001	90.74	0.1494	80.86	0.4746	85.72
RECAP (ours)	**0.3702***	**79.92***	**0.4345***	**81.76***	**0.2951***	**90.88***	**0.1494***	**81.41***	**0.4669***	**86.06***

Inference-time route aggregation: Finally, the recursive architecture provides a third estimator source at inference. Different LoRA selections define different recursive routes $\tau_1, \ldots, \tau_K$, which produce logits $z_{\tau_1}(x), \ldots, z_{\tau_K}(x)$.

RECAP combines them using mean-logit aggregation,

$$z^{(K)}_{\mathrm{RECAP}}(x) = \frac{1}{K} \sum_{k=1}^{K} z_{\tau_k}(x). \tag{15}$$

Increasing K therefore increases inference compute but leaves the stored model unchanged. This provides a test-time scaling knob that is complementary to both EMA and cross-run distillation.

4.3 Training Across Recursive Budgets

Because the recursive backbone may be evaluated at different inference depths, we train RECAP across multiple rollout budgets rather than specializing it to a single depth. Let $\mathcal{D}_{\mathrm{train}}$ denote the set of training depths. The supervised objective is

$$\mathcal{L}_{\mathrm{sup}} = \sum_{d \in \mathcal{D}_{\mathrm{train}}} \alpha_d \, \ell\left(y, z^{(d)}_{\theta}(x)\right), \tag{16}$$

where α_d controls the contribution of rollout depth d. During training, routes are sampled from the LoRA banks so that the shared backbone is exposed to multiple recursive computation paths.

We optimize RECAP with a decaying learning-rate schedule and maintain the EMA parameters from Eq. 10 throughout training. Teacher predictions are precomputed, so ensemble distillation does not require executing the teacher models during student optimization.

5 Experiments and Results

We evaluate RECAP with three questions in mind: *(1)* does estimator scaling improve predictive performance across standard CTR benchmarks; *(2)* are these gains explained by the non-shared predictive variation characterized in Section 3; and *(3)* can RECAP realize these gains without the deployment cost of a conventional ensemble?

Datasets and metrics: We evaluate on five public CTR benchmarks: Avazu Zhu et al. (2021), Criteo Zhu et al. (2021), MovieLens-1M (ML-1M) Song et al. (2019), KDD12 Song et al. (2019), and KKBox Zhu et al. (2022). We follow the preprocessing and evaluation protocol defined in the FuxiCTR framework which is used by prior open CTR benchmarks including FCN Zhu et al. (2021); Li et al. (2026); Zhu et al. (2022) . We report Area Under the ROC Curve (AUC), where higher is better, and binary cross-entropy (Logloss), where lower is better. Because improvements in mature CTR benchmarks are often small in absolute magnitude, we report AUC in percentage points throughout the paper.

Model	EMA	KD	Routes	AUC
Recursive backbone	–	–	–	81.59
+ trajectory scaling	✓	–	–	81.65
+ cross-run scaling	–	✓	–	81.68
+ route scaling	–	–	✓	81.61
RECAP	✓	✓	✓	**81.76**

KD denotes ensemble distillation; Routes denotes inference-time route aggregation.

Table 2: **Ablation of estimator-scaling components on the recursive backbone.** Trajectory scaling through EMA, cross-run scaling through ensemble distillation, and route scaling each improve performance individually, while combining all three yields the highest AUC.

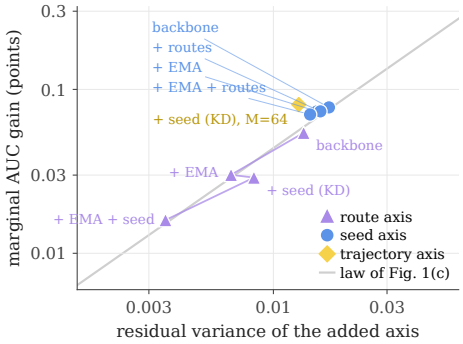

Figure 3: **Marginal estimator-scaling gains track residual predictive variation.** Across trajectory, seed, and inference-time route estimators, the marginal AUC improvement from adding an estimator axis increases with the residual non-shared variance available along that axis.

Baselines: We compare against representative factorization, deep interaction, and explicit cross-network models, including DNN Covington et al. (2016), PNN Qu et al. (2016), Wide & Deep Cheng et al. (2016), DeepFM Guo et al. (2017), DCN Wang et al. (2017), xDeepFM Lian et al. (2018), AutoInt Song et al. (2019), AFN Cheng et al. (2020), DCNv2 Wang et al. (2021a), EDCN Chen et al. (2021), MaskNet Wang et al. (2021b), EulerNet Tian et al. (2023), FinalMLP Mao et al. (2023), FINAL Zhu et al. (2023), RFM Tian et al. (2024), QNN Li et al. (2025), FCN Li et al. (2026). FCN is our primary reference architecture because it provides a strong recent explicit-interaction baseline and directly scales interaction capacity through complementary linear and exponential cross networks.

5.1 RECAP Improves CTR Prediction Across Benchmarks

Table 1 compares RECAP with representative CTR prediction models across five public benchmarks. RECAP achieves the highest AUC among the compared methods on all five datasets. Relative to the strongest compared baseline on each dataset, RECAP improves AUC by 0.26, 0.13, 0.01, 0.55, and 0.30 points on Avazu, Criteo, ML-1M, KDD12, and KKBox, respectively. RECAP also improves or matches Logloss on all of the datasets as compared to the next best baseline. These results indicate that the benefit of estimator scaling is not restricted to the Criteo diagnostic setting used in our analysis, but transfers across datasets with substantially different sparsity and scale.

5.2 Non-Shared Predictive Variation Explains Estimator-Scaling Gains

Section 3 makes two related predictions about estimator scaling. First, the standalone benefit of scaling a particular estimator source should depend on the non-shared predictive variation accessible along that axis. Second, when multiple estimator sources are combined, the marginal benefit of adding a new axis should depend on the residual variation that remains after the existing axes have already been incorporated. We test both predictions using the estimator families introduced in Figure 1(b).

Standalone estimator scaling: Figure 1(c) compares the gain from averaging estimators within each axis against the corresponding reducible predictive variation. Despite their different origins, the three estimator families follow a common trend: larger non-shared variation is associated with larger averaging gains. As shown in Figure 1(b), independent training runs occupy the high-variation, high-gain regime, checkpoints from a single optimization trajectory expose less reducible variation and provide smaller gains, and alternative inference-time routes through a shared model expose the least variation and provide the smallest gains. Equation 6 formally predicts expected BCE improvement. Empirically, AUC exhibits the same ordering: estimator families with greater reducible variation provide larger averaging gains. Results on additional datasets can be found in Appendix A.

Marginal gains from combining estimator axes: We next test the marginal improvement prediction from Section 3. Figure 3 plots the marginal AUC improvement obtained by adding an estimator axis to different existing configurations against the residual non-shared variance available to that axis. The same estimator source can provide different gains depending on what has already been averaged: when previous mechanisms remove overlapping variation, less residual variation remains and the incremental benefit decreases. Across seed, trajectory, and route estimators, configurations with

greater residual variation consistently yield larger marginal improvements. This relationship explains why the gains from different estimator-scaling mechanisms are complementary but not simply additive, and is consistent with the decomposition in Eq. 7. Table 2 further confirms complementarity within RECAP: each estimator source improves the same recursive backbone individually, while combining trajectory, cross-run, and route scaling yields the strongest performance.

5.3 Estimator Scaling Is Not Specific to the Recursive Backbone

To separate estimator-scaling gains from the recursive architecture, we apply the same training schedule, EMA, and ensemble-distillation objective to the FCN backbone. As shown in Table 3, both trajectory scaling through EMA and cross-run scaling through distillation improve FCN, confirming that estimator scaling is a general principle rather than a RECAP-specific effect. The key contribution of the recursive architecture is to introduce an additional *route-scaling axis*: weight sharing keeps interaction computation parameter-efficient, while lightweight route-specific adaptations define multiple related predictors within a single model. RECAP therefore scales estimators across training trajectories, independent runs, and alternative computation routes, with the latter providing a low-cost source of estimator diversity unavailable to the FCN backbone.

Model	EMA	KD	AUC
FCN	–	–	81.62
+ trajectory scaling	✓	–	81.67
+ cross-run scaling	–	✓	81.69
+ cross-run + trajectory scaling	✓	✓	81.74

Table 3: **Estimator scaling improves the FCN backbone.** Trajectory scaling through EMA and cross-run scaling through ensemble distillation each improve AUC individually, while combining both yields the strongest performance.

5.4 RECAP Shifts the Performance–Parameter Frontier

The performance–parameter analyses in Figures 1(d) and 4 provide complementary views of RECAP's deployment efficiency. Against a broad set of CTR architectures, RECAP substantially outperforms single-model baselines at a comparable parameter budget and approaches the five-model FCN ensemble (81.764 versus 81.771 AUC) while using only a fraction of its deployed parameters. The controlled scaling trajectories in Figure 4 further show that increasing interaction capacity through additional towers quickly yields diminishing returns, whereas independent-model ensembling continues to improve performance but at roughly linear deployment cost. RECAP lies above the interaction-capacity scaling trajectory and reaches the performance regime of substantially larger seed ensembles at approximately a single-model parameter budget. Together, these results suggest that once interaction-capacity scaling becomes inefficient, allocating resources toward estimator scaling provides a more favorable accuracy–parameter trade-off. RECAP captures much of this estimator-scaling benefit while avoiding the deployment cost of maintaining multiple full models.

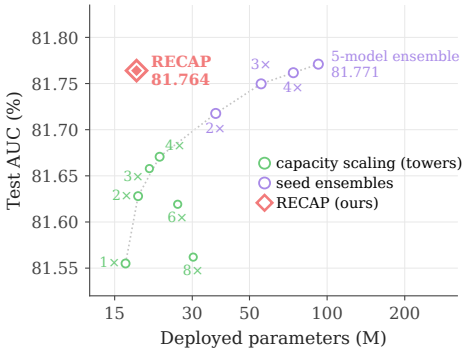

Figure 4: **RECAP improves the deployment performance–parameter frontier.** Increasing interaction capacity through additional towers yields diminishing returns, while seed ensembling continues to improve AUC at substantially higher deployment cost. RECAP reaches near five-model ensemble performance while retaining approximately a single-model parameter footprint.

6 Conclusion

In this work, we tackle the problem of CTR prediction and show that scaling interaction capacity in CTR models can exhibit diminishing predictive returns even when estimator diversity remains exploitable. This motivates *estimator scaling* as a complementary axis, with gains governed by the non-shared predictive variation available to aggregation. Building on this view, RECAP combines cross-run distillation, trajectory averaging, and route-based estimator scaling within a parameter-efficient recursive backbone. Across five CTR benchmarks, RECAP achieves state-of-the-art performance, while approaching five-model ensemble performance at approximately a single-model parameter budget.

REFERENCES

Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 3757–3766, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3481915. URL https://doi.org/10.1145/3459637.3481915.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, pp. 7–10, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450347952. doi: 10.1145/2988450.2988454. URL https://doi.org/10.1145/2988450.2988454.

Weiyu Cheng, Yanyan Shen, and Linpeng Huang. Adaptive factorization network: Learning adaptive-order feature interactions, 2020. URL https://arxiv.org/abs/1909.03276.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pp. 191–198, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959190. URL https://doi.org/10.1145/2959100.2959190.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pp. 1725–1731. AAAI Press, 2017. ISBN 9780999241103.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2019. URL https://arxiv.org/abs/1803.05407.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Honghao Li, Yiwen Zhang, Yi Zhang, Lei Sang, and Jieming Zhu. Revisiting feature interactions from the perspective of quadratic neural networks for click-through rate prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 1365–1375, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3737106. URL https://doi.org/10.1145/3711896.3737106.

Honghao Li, Yiwen Zhang, Yi Zhang, Hanwei Li, Lei Sang, and Jieming Zhu. Fcn: Fusing exponential and linear cross network for click-through rate prediction. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '26, pp. 681–691, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722585. doi: 10.1145/3770854.3780177. URL https://doi.org/10.1145/3770854.3780177.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 1754–1763, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220023. URL https://doi.org/10.1145/3219819.3220023.

Kelong Mao, Jieming Zhu, Liangcai Su, Guohao Cai, Yuru Li, and Zhenhua Dong. Finalmlp: an enhanced two-stream mlp model for ctr prediction. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*,

AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i4.25577. URL https://doi.org/10.1609/aaai.v37i4.25577.

Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1149–1154, 2016. doi: 10.1109/ICDM.2016.0151.

Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*, 2022.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pp. 1161–1170, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3357925. URL https://doi.org/10.1145/3357384.3357925.

Jiakai Tang, Runfeng Zhang, Weiqiu Wang, Yifei Liu, Chuan Wang, Xu Chen, Yeqiu Yang, Jian Wu, Yuning Jiang, and Bo Zheng. Loopctr: Unlocking the loop scaling power for click-through rate prediction, 2026. URL https://arxiv.org/abs/2604.19550.

Zhen Tian, Ting Bai, Wayne Xin Zhao, Ji-Rong Wen, and Zhao Cao. Eulernet: Adaptive feature interaction learning via euler's formula for ctr prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pp. 1376–1385, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591681. URL https://doi.org/10.1145/3539618.3591681.

Zhen Tian, Yuhong Shi, Xiangkun Wu, Wayne Xin Zhao, and Ji-Rong Wen. Rotative factorization machines. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 2912–2923, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671740. URL https://doi.org/10.1145/3637528.3671740.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, ADKDD'17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450351942. doi: 10.1145/3124749.3124754. URL https://doi.org/10.1145/3124749.3124754.

Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 1785–1797, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450078. URL https://doi.org/10.1145/3442381.3450078.

Zhiqiang Wang, Qingyun She, and Junlin Zhang. Masknet: Introducing feature-wise multiplication to ctr ranking models by instance-guided mask, 2021b. URL https://arxiv.org/abs/2102.07619.

Danny Wood, Tingting Mu, Andrew M. Webb, Henry W. J. Reeve, Mikel Luján, and Gavin Brown. A unified theory of diversity in ensemble learning. *Journal of Machine Learning Research*, 24 (359):1–49, 2023. URL http://jmlr.org/papers/v24/23-0041.html.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 2759–2769, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482486. URL https://doi.org/10.1145/3459637.3482486.

Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. Bars: Towards open benchmarking for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pp. 2912–2923, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450387323. doi: 10.1145/3477495.3531723. URL https://doi.org/10.1145/3477495.3531723.

Jieming Zhu, Qinglin Jia, Guohao Cai, Quanyu Dai, Jingjie Li, Zhenhua Dong, Ruiming Tang, and Rui Zhang. Final: Factorized interaction layer for ctr prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pp. 2006–2010, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591988. URL https://doi.org/10.1145/3539618.3591988.

A Diagnosis Experiments on Additional Datasets

To test whether the diagnostic findings from Criteo generalize beyond a single benchmark, we repeat the same analysis on additional CTR datasets. Figures 5, 6, 7 and 8 report the corresponding results on KKBox, ML-1M, KDD12, and Avazu. As in the main text, these diagnostics compare interaction-capacity scaling against estimator scaling and relate averaging gains to the amount of non-shared predictive variation available across estimator sources.

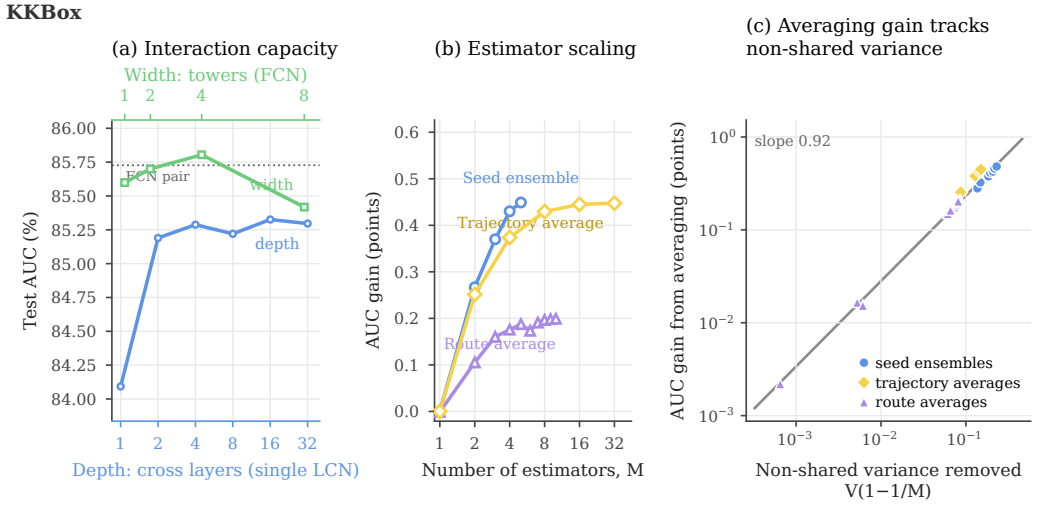

Figure 5: **Diagnostic analysis of interaction-capacity and estimator scaling on KKBox.** The same qualitative picture holds on KKBox: capacity scaling saturates, estimator scaling continues to help, and the gains from averaging track the amount of non-shared predictive variation exposed by each estimator source.

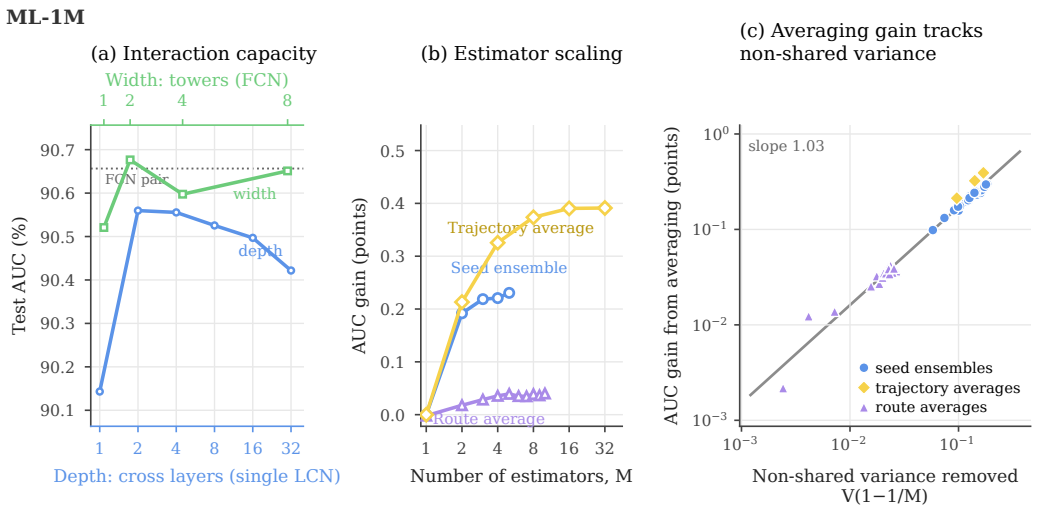

Figure 6: **Diagnostic analysis of interaction-capacity and estimator scaling on ML-1M.** ML-1M also exhibits diminishing returns from additional interaction capacity together with continued gains from estimator scaling. The observed averaging improvements remain strongly aligned with the amount of non-shared predictive variation available across estimator families.

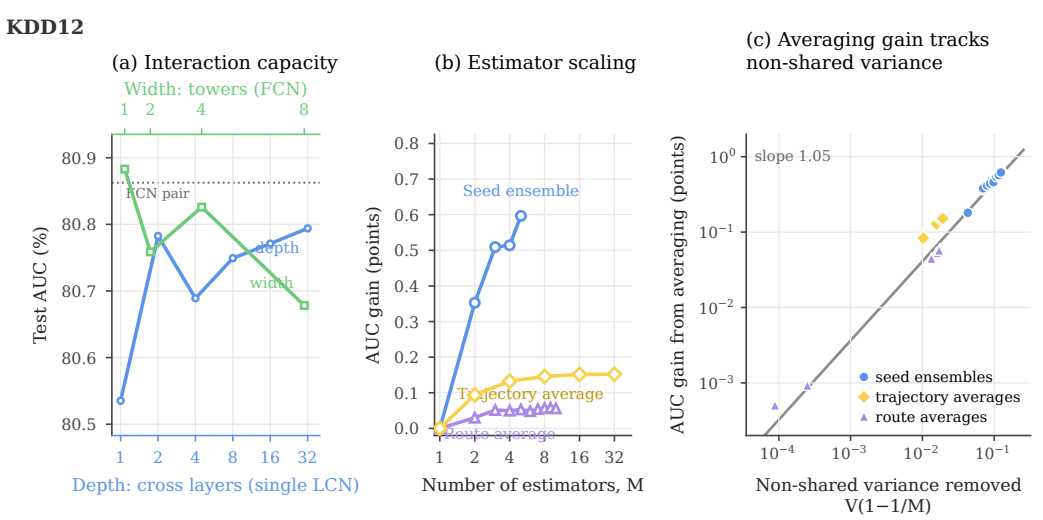

Figure 7: **Diagnostic analysis of interaction-capacity and estimator scaling on KDD12.** Increasing interaction capacity exhibits diminishing returns, whereas estimator scaling continues to provide gains. As in the main text, the improvement from averaging is closely associated with the non-shared predictive variation available to aggregation.

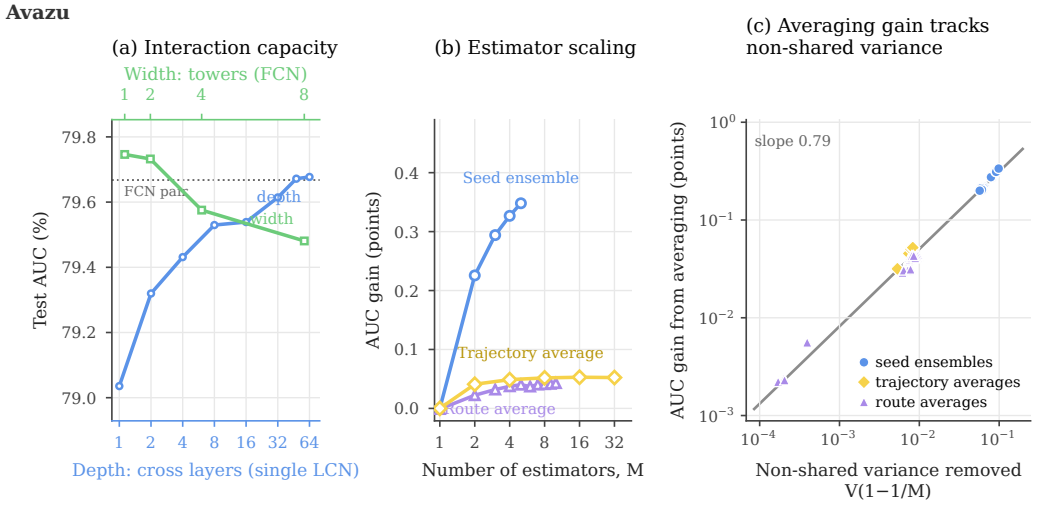

Figure 8: **Diagnostic analysis of interaction-capacity and estimator scaling on Avazu.** Increasing interaction capacity exhibits diminishing returns, whereas estimator scaling continues to provide gains. As in the main text, the improvement from averaging is closely associated with the non-shared predictive variation available to aggregation.

Across all four datasets, we observe the same qualitative pattern as on Criteo. First, increasing interaction capacity through greater depth or breadth yields diminishing returns once the underlying interaction model becomes sufficiently expressive. Second, averaging additional estimators continues to improve predictive performance after this capacity-scaling regime begins to saturate. Third, the magnitude of these gains is consistently associated with the amount of non-shared predictive variation exposed by each estimator source. These results support the main claim of the paper: interaction-capacity scaling and estimator scaling are complementary, and the usefulness of estimator scaling is governed by the reducible variation available to aggregation.

At the same time, the relative strength of different estimator sources varies across datasets. On some datasets, independent training runs provide the largest gains, while on others trajectory-based averaging is comparatively stronger. This variation is consistent with the theory in Section 3: the benefit of an estimator axis is determined not by its label (seed, trajectory, or route) but by how much non-shared predictive variation it contributes. This further motivates RECAP's use of multiple estimator axes rather than relying on a single source of diversity.

Taken together, these additional results show that the diagnosis in Figure 1 is not specific to Criteo. Across diverse CTR benchmarks, increasing the capacity of a single predictor eventually becomes inefficient, while estimator scaling remains beneficial whenever additional non-shared predictive variation is available. This consistency across datasets provides further support for the estimator-scaling perspective developed in the main paper.

B Additional Theory for Estimator Scaling

This appendix provides the derivations underlying Section 3. We first extend the equicorrelated analysis in the main text to arbitrary estimator covariance, and then derive the local relationship between reducible predictive variation and expected binary cross-entropy. Throughout this appendix, estimator variation is defined conditionally on the input x and for a fixed interaction capacity c.

B.1 General Covariance Formulation

For a fixed input x and interaction capacity c, let

$$\mathbf{Z}_c(x) = [Z_{c,1}(x), \dots, Z_{c,M}(x)]^\top \tag{17}$$

denote the logits of M estimators, with conditional covariance matrix

$$\Sigma_c(x) = \mathrm{Cov}(\mathbf{Z}_c(x) \mid x)\,. \tag{18}$$

Their mean-logit aggregate is

$$\bar{Z}_{c,M}(x) = \frac{1}{M}\mathbf{1}^\top \mathbf{Z}_c(x), \tag{19}$$

and therefore

$$\mathrm{Var}\left(\bar{Z}_{c,M}(x) \mid x\right) = \frac{1}{M^2}\mathbf{1}^\top \Sigma_c(x)\mathbf{1}. \tag{20}$$

The average conditional variance of an individual estimator is

$$V_{\mathrm{ind}}(c; x) = \frac{1}{M}\,\mathrm{tr}(\Sigma_c(x))\,. \tag{21}$$

We define the variation removed by averaging as

$$V_{\mathrm{ns}}(c, M; x) = \frac{1}{M}\,\mathrm{tr}(\Sigma_c(x)) - \frac{1}{M^2}\mathbf{1}^\top \Sigma_c(x)\mathbf{1}. \tag{22}$$

Equation 22 shows that averaging gains depend jointly on estimator variance and covariance. A collection of estimators can exhibit substantial individual variation while providing little reducible variation if that variation is strongly shared.

For the equicorrelated setting used in the main text, suppose

$$\mathrm{Var}\left(Z_{c,m}(x) \mid x\right) = \tau_c^2(x), \qquad \mathrm{Corr}\left(Z_{c,i}(x), Z_{c,j}(x) \mid x\right) = \rho_c(x), \quad i \neq j. \tag{23}$$

Positive semidefiniteness requires

$$\rho_c(x) \geq -\frac{1}{M-1}. \tag{24}$$

Substituting into Eq. 20 gives

$$\mathrm{Var}\left(\bar{Z}_{c,M}(x) \mid x\right) = \tau_c^2(x)\left(\rho_c(x) + \frac{1-\rho_c(x)}{M}\right). \tag{25}$$

Since a single estimator has variance $\tau_c^2(x)$, the variation removed by averaging is

$$V_{\mathrm{ns}}(c, M; x) = \tau_c^2(x)\,(1-\rho_c(x))\left(1 - \frac{1}{M}\right), \tag{26}$$

recovering Eq. 5.

B.2 From Reducible Variation to Binary Cross-Entropy

We next derive Eqs. 3 and 6. Recall that binary cross-entropy in logit space is

$$\ell(y, z) = \log(1 + \exp(z)) - yz. \tag{27}$$

For a fixed x, write the M-estimator aggregate as

$$\bar{Z}_{c,M}(x) = \mu_c(x) + \epsilon_{c,M}(x), \qquad \mathbb{E}\left[\epsilon_{c,M}(x) \mid x\right] = 0, \tag{28}$$

where $\mu_c(x) = \mathbb{E}[Z_c(x) \mid x]$ is the estimator-family mean.

A second-order Taylor expansion around $\mu_c(x)$ gives

$$\begin{aligned}\ell\left(y, \mu_c + \epsilon_{c,M}\right) = \ell(y, \mu_c) + \ell'(y, \mu_c)\epsilon_{c,M} \\ + \frac{1}{2}\ell''(y, \mu_c)\epsilon_{c,M}^2 + O\left(|\epsilon_{c,M}|^3\right).\end{aligned} \tag{29}$$

Taking expectation over estimator randomness conditional on x eliminates the first-order term:

$$\begin{aligned}\mathbb{E}\left[\ell\left(y, \bar{Z}_{c,M}(x)\right) \mid x\right] \approx \ell\left(y, \mu_c(x)\right) \\ + \frac{1}{2}\ell''\left(y, \mu_c(x)\right) \operatorname{Var}\left(\bar{Z}_{c,M}(x) \mid x\right).\end{aligned} \tag{30}$$

For binary cross-entropy,

$$\ell''(y, z) = \sigma(z)\left(1 - \sigma(z)\right), \tag{31}$$

which is independent of the label y. Averaging over the data distribution therefore gives

$$\mathcal{L}(c, M) \approx \mathcal{L}(\mu_c) + \frac{1}{2}\mathbb{E}_x\left[\sigma(\mu_c(x))\left(1 - \sigma(\mu_c(x))\right) \operatorname{Var}\left(\bar{Z}_{c,M}(x) \mid x\right)\right], \tag{32}$$

which is Eq. 3 in the main text, up to the omitted third-order remainder.

Comparing a single estimator with an M-estimator aggregate and using Eq. 5, the corresponding reduction in expected BCE is

$$\Delta\mathcal{L}(c, M) \approx \frac{1}{2}\mathbb{E}_x\left[\sigma(\mu_c(x))\left(1 - \sigma(\mu_c(x))\right) V_{\mathrm{ns}}(c, M; x)\right]. \tag{33}$$

Thus, BCE improvement is locally proportional to *curvature-weighted reducible predictive variation*. When the estimator families being compared have similar mean predictions, the curvature term $\sigma(\mu_c(x))(1 - \sigma(\mu_c(x)))$ is approximately shared, so differences in averaging benefit are primarily governed by $V_{\mathrm{ns}}(c, M; x)$. This motivates the reducible-variance diagnostic used in the main experiments. The theoretical proportionality applies directly to expected BCE; the corresponding association with AUC reported in the main text is empirical rather than implied by this derivation.

The approximation is most accurate when estimator disagreement around $\mu_c(x)$ is modest. For broader or heavy-tailed estimator distributions, the omitted third-order and higher-order terms may become non-negligible.

B.3 Exact Identity for Literal Mean-Logit Aggregation

For literal mean-logit aggregation, binary cross-entropy also admits an exact inequality. For a fixed example (x, y) and estimator logits $z_1, \ldots, z_M$, let

$$\bar{z} = \frac{1}{M}\sum_{m=1}^{M} z_m. \tag{34}$$

Since

$$\ell(y, z) = \mathrm{softplus}(z) - yz, \tag{35}$$

we obtain

$$\begin{aligned}\frac{1}{M}\sum_{m=1}^{M} \ell(y, z_m) - \ell(y, \bar{z}) = \frac{1}{M}\sum_{m=1}^{M} \mathrm{softplus}(z_m) - \mathrm{softplus}(\bar{z}) \\ \geq 0,\end{aligned} \tag{36}$$

where the inequality follows from convexity of $\mathrm{softplus}$.

Equation 36 shows that literal mean-logit aggregation cannot have higher BCE than the average BCE of its constituent logits on the same example. This exact result does not, however, quantify the gain in terms of reducible variance, nor does it directly characterize parameter averaging such as EMA or a distilled student that only approximates an ensemble prediction. For this reason, the main text uses the local variance formulation as the common explanatory framework across estimator sources.

C Combining Multiple Estimator Axes

C.1 Two-Axis Decomposition

Let A and B denote two estimator sources and let $Z_{A,B}(x)$ denote the corresponding random logit for a fixed input x. Applying the law of total variance while treating A as the first averaging axis gives

$$\mathrm{Var}\left(Z_{A,B}(x) \mid x\right) = \mathbb{E}_B\left[\mathrm{Var}_A\left(Z_{A,B}(x) \mid B, x\right)\right] + \mathrm{Var}_B\left[\mathbb{E}_A\left(Z_{A,B}(x) \mid B, x\right)\right]. \tag{37}$$

Define

$$V_A(x) = \mathbb{E}_B\left[\mathrm{Var}_A\left(Z_{A,B}(x) \mid B, x\right)\right], \tag{38}$$

which is the variation directly accessible to averaging along axis A. After averaging over A, the remaining predictor is

$$\mu_A(B; x) = \mathbb{E}_A\left[Z_{A,B}(x) \mid B, x\right], \tag{39}$$

whose residual variation across B is

$$V_{B|A}(x) = \mathrm{Var}_B\left[\mu_A(B; x)\right]. \tag{40}$$

Hence

$$\mathrm{Var}\left(Z_{A,B}(x) \mid x\right) = V_A(x) + V_{B|A}(x). \tag{41}$$

Up to the same local BCE-curvature weighting derived in Appendix B.2, $V_A(x)$ determines the standalone loss reduction available from averaging along axis A, while $V_{B|A}(x)$ determines the residual variation available to a subsequent estimator axis B. This is the basis for the marginal-gain analysis in Section 5.2.

The decomposition is order-dependent. Reversing the averaging order gives

$$\mathrm{Var}\left(Z_{A,B}(x) \mid x\right) = V_B(x) + V_{A|B}(x), \tag{42}$$

and in general

$$V_A(x) \neq V_{A|B}(x), \qquad V_B(x) \neq V_{B|A}(x). \tag{43}$$

The total predictive variance is unchanged, but its attribution to the estimator axes depends on the averaging order.

C.2 Three-Axis Decomposition for RECAP

Let S, T, and R denote independent training runs, checkpoints along a training trajectory, and inference-time routes, respectively. For a fixed input x, let $Z_{S,T,R}(x)$ denote the resulting random logit. We use the operational nesting order

$$R \to T \to S, \tag{44}$$

corresponding to route averaging first, trajectory averaging next, and cross-run averaging last.

Applying the law of total variance recursively gives

$$\begin{aligned}\mathrm{Var}\left(Z_{S,T,R}(x) \mid x\right) = {} & \underbrace{\mathbb{E}_{S,T}\left[\mathrm{Var}_R\left(Z_{S,T,R}(x) \mid S, T, x\right)\right]}_{V_R(x)} \\ & + \underbrace{\mathbb{E}_S\left[\mathrm{Var}_T\left(\mathbb{E}_R\left[Z_{S,T,R}(x) \mid S, T, x\right] \mid S, x\right)\right]}_{V_{T|R}(x)} \\ & + \underbrace{\mathrm{Var}_S\left[\mathbb{E}_{T,R}\left(Z_{S,T,R}(x) \mid S, x\right)\right]}_{V_{S|T,R}(x)}.\end{aligned} \tag{45}$$

The three terms have direct operational interpretations:

- $V_R(x)$ is variation accessible to inference-time route averaging;
- $V_{T|R}(x)$ is trajectory variation remaining after route averaging;
- $V_{S|T,R}(x)$ is cross-run variation remaining after both route and trajectory averaging.

Equation 45 explains why gains from different estimator sources need not add linearly. If two axes expose overlapping predictive variation, averaging one reduces the residual variation available to the next; if they expose largely distinct variation, their gains can be closer to additive. There are $3! = 6$ possible nestings of (S, T, R). Their component values generally differ, although all decompositions sum to the same total predictive variance. The decomposition is an output-space attribution of the variation associated with the three estimator sources; it should not be interpreted as the literal sequence of operations implemented by RECAP. For a fixed attribution convention, we use the nesting order $R \to T \to S$, attributing route-accessible variation first, followed by residual trajectory and cross-run variation.

D Hyperparameter Sensitivity

Table 4: Hyperparameter sensitivity of RECAP on Criteo. We vary one method-specific hyperparameter at a time while keeping all other settings fixed. Logloss is the test logloss after validation-fitted affine calibration.

Hyperparameter	Value	AUC (%) ↑	Logloss ↓
EMA decay β	0.9900	81.735	0.43487
	0.9950	81.744	0.43482
	0.9990	81.760	0.43461
	0.9995	**81.765**	**0.43456**
	0.9999	81.770	0.43450
KD weight λ_{KD}	0.00	81.665	0.43535
	0.25	81.743	0.43461
	0.50	81.766	0.43443
	1.00	**81.765**	**0.43456**
	2.00	81.763	0.43454
Teacher seeds M_{seed}	1	81.704	0.43502
	2	81.744	0.43478
	3	81.756	0.43469
	5	**81.765**	**0.43456**
	8	81.768	0.43454
LoRA rank r_{LoRA}	24	81.763	0.43459
	48	81.765	0.43444
	96	**81.765**	**0.43456**
	192	81.767	0.43444
Route adapters M_{route}	2	81.759	0.43452
	4	81.767	0.43448
	8	**81.765**	**0.43456**
	16	81.764	0.43444

We study the sensitivity of RECAP to its main method-specific hyperparameters on Criteo. We vary one hyperparameter at a time while keeping all remaining settings fixed to the default configuration used in the main experiments. We consider five quantities: the EMA decay β, the distillation weight λ_{KD}, the number of teacher seed models M_{seed}, the LoRA rank r_{LoRA}, and the number of route adapters M_{route}. These parameters control trajectory averaging, the strength and breadth of cross-run distillation, and the capacity of the route-scaling mechanism. Unless otherwise specified, each configuration is evaluated using the same training protocol and data split as the main experiments.

Table 4 reports the resulting test AUC and Logloss. Overall, RECAP is stable across a broad range of settings, with the default configuration lying in regions of consistently strong performance rather than at isolated optima. Performance varies only modestly across EMA decay values, while nonzero distillation weights consistently outperform the no-distillation setting. Increasing the number of teacher seeds improves performance with diminishing returns, consistent with the saturation behavior observed for independent estimators. Likewise, increasing the LoRA rank or the number of route adapters beyond moderate values provides little additional improvement.

These results indicate that RECAP does not depend on narrowly tuned hyperparameters. In particular, its performance remains robust across the trajectory-averaging, cross-run distillation, and route-capacity settings considered here.