ORBITTUNE: INPUT-OUTPUT TASK-ORBIT SYMMETRIZATION FOR RELIABLE FINE-TUNING OF TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning tabular foundation models typically optimizes performance in a single representation of a downstream table, even though feature scales, category codes, class-label IDs, and target units can be arbitrary coordinates of the same prediction task. We propose OrbitTune, an input-output task-orbit objective that fine-tunes models across equivalent task views. OrbitTune consistently reparameterizes query features, context features, context labels, and query targets, and evaluates predictions after decoding to the original output coordinates. This preserves the semantic supervision while encouraging consistent predictions across task representations. We establish fixed-task semantic preservation and characterize Bayes equivariance under a task-orbit symmetric prior. On controlled structural causal model classification tasks, standard fine-tuning exhibits confidence saturation and loss concentration, whereas OrbitTune changes sample-level training dynamics and improves negative log-likelihood and calibration error. Ablations of query removal and task-view replacement over entire updates support the usefulness of samples that become more variable under OrbitTune. On TabZilla, OrbitTune improves mean accuracy, AUC, and macro-F1 over full-parameter fine-tuning and the evaluated augmentation baselines for both LimiX-16M and TabPFN-2.5. On TabArena, its two backbone variants attain the highest classification Elo point estimates among the compared methods. These results support task-level input-output reparameterization as a practical strategy for adapting tabular foundation models.

1 INTRODUCTION

Tabular foundation models (TFMs) predict query rows using a labeled context from the same task (Hollmann et al., 2023; 2025; Qu et al., 2025; Grinsztajn et al., 2025; Qu et al., 2026; Zhang et al., 2025). Their downstream adaptation must account for a basic property of tabular prediction: a supervised task can have multiple equivalent representations. Numerical scales, nominal category IDs, class-label IDs, and regression target units need not determine the semantic prediction problem. Reparameterizing these coordinates consistently across the query and its context should therefore produce a corresponding change in the output representation, while leaving the decoded target unchanged.

Standard fine-tuning optimizes a single observed task view without explicitly encouraging this consistency. Because TFMs condition on labeled contexts, this consistency concerns both feature representations and the label or target coordinates supplied to the model. This motivates a fine-tuning objective that uses these views while retaining the same semantic supervision. We complement this formulation with sample-level training-dynamics diagnostics (Swayamdipta et al., 2020; Seedat et al., 2022; Ponti et al., 2024) to examine how task reparameterization changes downstream adaptation.

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.

We propose OrbitTune, an input-output task-orbit objective for TFM fine-tuning. OrbitTune samples a coordinate change and applies it consistently to query features, context features, context labels, and training query targets. It uses feature-wise positive affine maps for numerical inputs, bijections of nominal category IDs, label-code permutations for classification, and positive affine target maps for regression. Predictions are decoded to the original output coordinates before evaluating the supervised loss. This preserves the semantic task while encouraging consistency across its representations. Validation and test use the original normalized view, so OrbitTune requires no inference-time ensemble.

OrbitTune builds on the interpretation of data augmentation as averaging over transformation orbits (Chen et al., 2020), applying this perspective to downstream fine-tuning of existing pretrained backbones using joint input-output task views. This differs from row-generation or interpolation methods such as SMOTE, CTGAN, and mixup (Chawla et al., 2002; Xu et al., 2019; Zhang et al., 2018), as well as corruption-based tabular augmentation (Onishi & Meguro, 2023): OrbitTune changes the coordinates of the entire supervised task without adding new semantic examples. We prove that valid reparameterizations preserve the fixed-task semantic target and characterize the Bayes predictor's equivariance under a task-orbit symmetric prior. These results motivate the objective; they do not guarantee exact equivariance of a model trained with finite data and updates.

Empirically, full-parameter fine-tuning exhibits confidence saturation and loss concentration on SCM-generated classification tasks. OrbitTune shifts some samples from high-confidence, low-variability trajectories toward more variable, ambiguous dynamics, and improves test NLL and ECE. Targeted ablations on TabZilla support the contribution of these transition samples as training queries and of task transformations on their updates; component ablations further support combining input and output reparameterizations. Across both evaluated backbones, OrbitTune improves the reported TabZilla mean accuracy, AUC, and macro-F1 over full-parameter fine-tuning and the evaluated augmentation baselines. It also achieves the two highest classification Elo point estimates in our TabArena comparison. Regression results on TabArena and TALENT provide complementary performance evidence; the training-dynamics analysis focuses on classification. The controlled backbone comparisons use single-model evaluation, with the provenance and inference-budget limits of the broader TabArena comparison described in the appendix.

Our contributions are threefold.

- We introduce OrbitTune, a joint input-output task-view objective with decoded supervision, and establish its fixed-task preservation and conditional Bayes-equivariance properties.
- We characterize sample-level training dynamics on controlled classification tasks and use targeted ablations to study the contribution of transition samples as training queries and task transformations on the corresponding updates.
- We evaluate OrbitTune on two pretrained backbones across classification and regression benchmarks, reporting improved downstream performance and complementary calibration results on controlled classification tasks.

2 Method: Input-Output Task-Orbit Symmetrization

2.1 Input-Output Task Orbits

A tabular foundation model (TFM) predicts a query x through a labeled task context

$$C = \{(x_j, y_j)\}_{j=1}^{m}.$$

For a schema s, OrbitTune defines a family of valid input-output reparameterizations

$$\mathcal{R}_s = \mathcal{G}_s \times \mathcal{H}_s, \qquad r = (g, h) \in \mathcal{R}_s,$$

where $g : \mathcal{X}_s \to \mathcal{X}_s$ acts on covariates and $h : \mathcal{Y}_s \to \mathcal{Y}_s$ acts on labels or regression targets. For a query-context pair $z = (x, C)$, the task-level action is

$$S_r z = (gx, rC), \qquad rC = \{(gx_j, h(y_j)) : (x_j, y_j) \in C\}.$$

During training, the query target is also transformed as $y \mapsto h(y)$. OrbitTune therefore fine-tunes over the orbit of the same supervised task: query features, context features, context labels, and query targets are reparameterized consistently.

A reparameterization $r = (g, h)$ is valid if it satisfies five task-level constraints:

(V1) **Bijective:** g and h are measurable bijections with measurable inverses.

(V2) **Separated:** g acts only on covariates, and h acts only on labels or targets.

(V3) **Type-preserving:** g preserves feature type and missingness structure.

(V4) **Task-consistent:** the same r is applied to the query and the entire labeled context.

(V5) **Split-safe:** r does not use validation or test data.

These constraints exclude non-invertible operations such as feature dropout and clipping, changes to missingness structure, and transformations applied inconsistently across rows or between query and context. OrbitTune additionally restricts transformations to preserve individual column semantics: it does not mix columns or permute ordinal categories. These are design restrictions beyond (V1)–(V5), which alone can admit invertible mixing of numerical columns. OrbitTune constructs alternative coordinate views of a supervised task rather than new semantic samples.

2.2 Covariate Reparameterizations

Each input row is represented as numerical features, categorical features, and missingness indicators,

$$x = (x^{\mathrm{num}}, x^{\mathrm{cat}}, m).$$

For numerical features, OrbitTune uses feature-wise positive affine transformations in the train-normalized feature space:

$$g_{\mathrm{num}}(x)_j = a_j x_j + b_j, \qquad a_j = \exp(u_j), \quad u_j \sim \mathrm{Unif}[-\rho_x, \rho_x], \qquad b_j \sim \mathrm{Unif}[-\beta_x, \beta_x].$$

Equivalently, the numerical transformation matrix is diagonal,

$$D = \mathrm{diag}(a_1, \ldots, a_{d_{\mathrm{num}}}).$$

We restrict numerical maps to diagonal transformations to retain the identity of each column, such as age, income, or a laboratory measurement. An invertible dense map can satisfy (V1)–(V5), but falls outside this chosen family because it mixes columns. Missing numerical entries and their missingness indicators are left unchanged.

For nominal categorical features, OrbitTune samples a column-wise bijection

$$\pi_j : \mathcal{C}_j^{\circ} \to \mathcal{C}_j^{\circ}, \qquad \mathcal{C}_j^{\circ} = \mathcal{C}_j \setminus \{\texttt{MISSING}, \texttt{UNK}\},$$

and sets

$$g_{\mathrm{cat}}(x)_j = \pi_j(x_j), \qquad \pi_j(\texttt{MISSING}) = \texttt{MISSING}, \qquad \pi_j(\texttt{UNK}) = \texttt{UNK}.$$

We apply categorical permutations only to nominal categorical IDs. Ordinal categories are excluded. String-valued categories are eligible only when the model treats them as anonymous nominal IDs rather than as semantic text.

2.3 Output Reparameterizations and Orbit Risk

OrbitTune also reparameterizes the output space because context labels or targets are part of the TFM input. The loss is evaluated in semantic coordinates, so output reparameterization changes the task view without changing the supervised target.

Classification.
For class IDs $\mathcal{Y}$, OrbitTune samples a label-code bijection $h \in \mathrm{Sym}(\mathcal{Y})$ and applies it to all context labels and the query target. If $q = f_\theta(gx_i, rC_i) \in \Delta(\mathcal{Y})$, then

$$\ell^r_{\mathrm{cls}}(q, y_i) = -\log q_{h(y_i)} = -\log \left((h^{-1})_{\#} q\right)_{y_i},$$

where $(h^{-1})_{\#} q$ decodes probabilities back to the original label coordinates.

Regression.
For regression, OrbitTune uses positive affine target reparameterizations in the train-normalized target space,

$$h(y) = cy + d, \qquad c = \exp(v), \quad v \sim \mathrm{Unif}[-\rho_y, \rho_y], \qquad d \sim \mathrm{Unif}[-\beta_y, \beta_y].$$

For $\widehat{y}' = f_\theta(gx_i, rC_i)$, we use the decoded squared loss

$$\ell^r_{\mathrm{reg}}(\widehat{y}', y_i) = \left(h^{-1}(\widehat{y}') - y_i\right)^2 = \frac{(\widehat{y}' - h(y_i))^2}{c^2}.$$

The factor c^{-2} prevents target reparameterization from becoming arbitrary loss reweighting.

Let

$$\mathcal{D}_{\mathrm{train}} = \{(x_i, y_i)\}_{i=1}^N, \qquad C_i \subseteq \mathcal{D}_{\mathrm{train}} \setminus \{(x_i, y_i)\}$$

be the downstream fine-tuning set and the context for query x_i. OrbitTune minimizes the empirical task-orbit risk

$$\widehat{R}_{\mathrm{OrbitTune}}(\theta) = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{r=(g,h)\sim\bar{\nu}_s} \left[\ell^r \left(f_\theta(gx_i, rC_i), y_i\right)\right],$$

where

$$\bar{\nu}_s = (1-\lambda)\delta_{(\mathrm{id}_\mathcal{X}, \mathrm{id}_\mathcal{Y})} + \lambda \nu_s.$$

Here ν_s samples valid input-output task reparameterizations and $\lambda \in [0, 1]$ controls the probability of using a non-identity task view. Each update uses a one-sample Monte Carlo estimate of this risk. At validation and test time, OrbitTune uses the original normalized task view $r = (\mathrm{id}_\mathcal{X}, \mathrm{id}_\mathcal{Y})$; it is a training-time symmetrization method, not an inference-time ensemble. Pseudocode is provided in Alg. 1.

2.4 Fixed-Task Preservation

OrbitTune does not assume that arbitrary transformations preserve labels. Preservation follows from using invertible input-output task reparameterizations.

Lemma 2.1 (Fixed-task pushforward preservation)**.** *Let τ be a fixed downstream task with distribution P_τ over $\mathcal{X} \times \mathcal{Y}$, and let $r = (g, h)$ be valid. Define the transformed task by*

$$P_{r\cdot\tau} = (g, h)_{\#} P_\tau.$$

If $C_m = \{(X_j, Y_j)\}_{j=1}^m$ and (X, Y) are i.i.d. samples from P_τ, then rC_m and $(gX, h(Y))$ are i.i.d. samples from $P_{r\cdot\tau}$. Moreover, for P_τ^{m+1}-almost every $z = (x, C_m)$,

$$\mathcal{L}_{r\cdot\tau}\left(Y' \mid X' = gx, C'_m = rC_m\right) = h_{\#}\mathcal{L}_\tau\left(Y \mid X = x, C_m\right).$$

Thus, an OrbitTune-transformed view has the same semantic prediction problem as the original task, expressed in a different input-output coordinate system.

The proof is provided in Sec. C, where we restate the shared notation before giving the formal argument.

2.5 Bayes Equivariance Under Task-Orbit Symmetry

The fixed-task lemma preserves the semantic target for each transformed view. We next state a conditional equivariance result for a chosen symmetry subfamily $\mathcal{K} \subseteq \mathcal{R}$, which need not cover the full augmentation family. Let $\mathcal{T}$ be a family of tabular tasks and $\tau \sim \Pi$, where Π is a probability distribution over tasks. Given τ, the context and query are sampled i.i.d. from P_τ. We call Π $\mathcal{K}$-invariant if

$$(r\cdot)_{\#}\Pi = \Pi \qquad \text{for every } r \in \mathcal{K},$$

meaning that the meta-distribution is unchanged by each transformation in $\mathcal{K}$. This is an assumption on the task prior, distinct from how transformations are sampled during fine-tuning.

Theorem 2.2 (Bayes equivariance under an invariant task prior)**.** *Let $Z = (X, C_m)$ and $S_r Z = (gX, rC_m)$. For each fixed $r = (g, h) \in \mathcal{K}$, assume that r is valid, $r \cdot \tau \in \mathcal{T}$ for every $\tau \in \mathcal{T}$, and Π is $\mathcal{K}$-invariant. Define*

$$\eta(z) = \mathcal{L}(Y \mid Z = z).$$

Then, for P_Z-almost every z,

$$\eta(S_r z) = h_{\#}\eta(z).$$

In particular,

$$\textit{classification:} \quad \eta_{h(y)}(S_r z) = \eta_y(z),$$
$$\textit{regression:} \quad m^\star(S_r z) = h(m^\star(z)), \qquad m^\star(z) = \mathbb{E}[Y \mid Z = z],$$

where the regression statement assumes $h(y) = cy + d$ and $\mathbb{E}[Y^2] < \infty$. After decoding, the semantic Bayes target is unchanged:

$$(h^{-1})_{\#}\eta(S_r z) = \eta(z)$$

for classification distributions, and $h^{-1}(m^\star(S_r z)) = m^\star(z)$ for affine regression targets.

The proof is given in Sec. C. The invariant-prior assumption cannot hold for the full affine family when it includes a nonzero translation of an observed real-valued coordinate; centering the sampled translations does not remove this obstruction. Sec. C.1 gives the argument and explains a valid application to finite permutation symmetries. Thus, Theorem 2.2 is a conditional characterization, while preservation of individual affine task views follows from Lem. 2.1. Neither result guarantees exact equivariance or improved generalization for a finitely trained model. Additional validity conditions, pseudocode, and a local curvature interpretation are provided in Sec. D.

3 Experiments, Results, and Discussion

We use 100 independently generated structural causal model (SCM) classification tasks to study fine-tuning dynamics and calibration, and real tabular benchmarks to evaluate downstream performance. Diagnostics use LimiX-16M unless stated otherwise. In the controlled comparisons, full-parameter FT and OrbitTune start from the same checkpoint and use the same optimization and model-selection settings; OrbitTune adds the sampled task views. Full experimental settings and benchmark details are in Secs. E and H. TabArena comparisons are conditional on the available result records, whose protocol limitations are described in the appendix.

3.1 Standard Fine-Tuning Produces Saturated Dynamics

Figs. 1 and 2 show that full-parameter FT rapidly drives many training samples into a high-confidence regime while concentrating the remaining loss on fewer samples. The saturation ratio $S_t(\alpha)$ is the fraction of samples whose true-label probability exceeds α; normalized loss effective sample size ESS_t/N measures how broadly loss is distributed across the training set. Their definitions are given in Sec. B.2. These trajectories characterize full-parameter FT; they do not by themselves establish that OrbitTune prevents saturation.

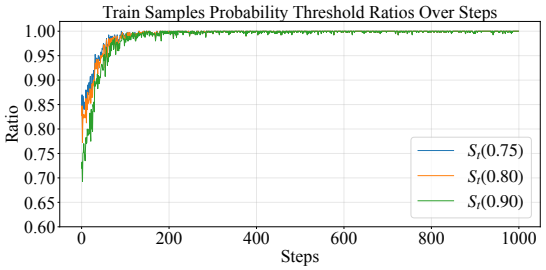


Figure 1: **Confidence saturation under full-parameter FT.** Mean trajectories over 100 SCM tasks for $\alpha \in \{0.75, 0.80, 0.90\}$.

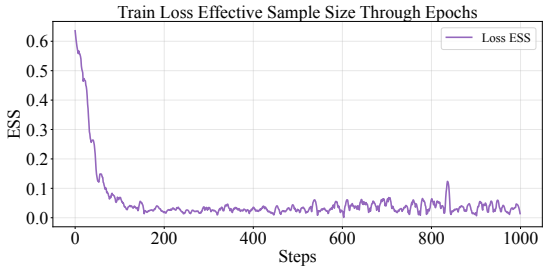


Figure 2: **Loss concentration under full-parameter FT.** Mean normalized loss effective sample size over the same 100 SCM tasks.

3.2 OrbitTune Changes Sample-Level Training Dynamics

The SCM data maps in Figs. 3 and 4 compare each sample's mean true-label probability and its variability across checkpoints, measured using the same aligned context and query views for both methods. Full-parameter FT concentrates samples in the high-confidence, low-variability region. OrbitTune shifts part of this distribution toward more variable, ambiguous dynamics, without a clear expansion of the hard-to-learn region in the displayed maps. We call samples that are easy under full-parameter FT and ambiguous under OrbitTune the transition set $\mathcal{S}_{E\to A}$. These are trajectory-based diagnostic labels, rather than intrinsic sample types; formal definitions are in Secs. B.1 and B.3.

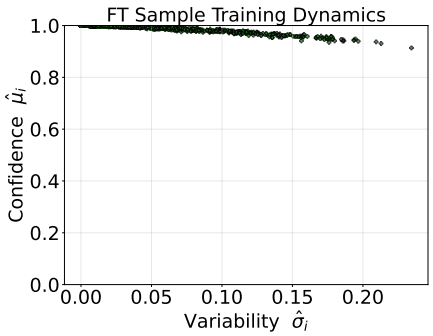


Figure 3: **Full-parameter FT data map.** Each point is a training sample; axes show variability and mean true-label probability across checkpoints.

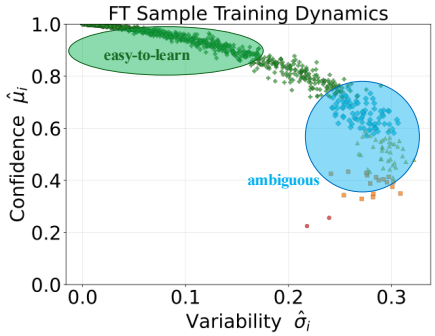


Figure 4: **OrbitTune data map.** Both axes use the same confidence and variability statistics as Fig. 3; the horizontal ranges differ.

3.3 Transition Samples Contribute to Downstream Performance

On TabZilla, we exclude $\mathcal{S}_{E\to A}$ or an equally sized random subset drawn from the entire training set from the query pool, keeping the context pool unchanged. A third intervention uses original views for entire updates involving transition queries. Sec. G.1 gives the grouping and retraining protocols. Targeted query removal causes larger performance drops than random removal (Fig. 5); original-view replacement causes smaller drops than targeted removal. Negative values denote degradation relative to OrbitTune-full. These ablations

support the contribution of transition queries and task transformations on their updates, without estimating a formal causal effect.

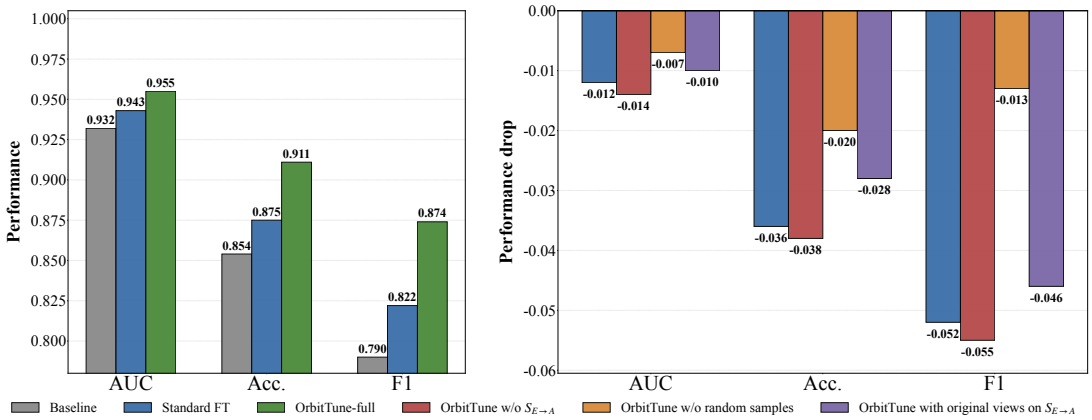


Figure 5: **Targeted ablation on TabZilla.** Left: mean performance of zero-shot inference (Baseline), full-parameter FT (Standard FT), and OrbitTune-full. Right: signed changes from OrbitTune-full for Standard FT, targeted query removal, equally sized random query removal, and original-view replacement over entire updates. Removal leaves the context pool unchanged. Each change is the evaluated setting's performance minus that of OrbitTune-full; negative values indicate degradation.

3.4 OrbitTune Improves Predictive Calibration

On SCM test sets, OrbitTune achieves lower negative log-likelihood (NLL) and 15-bin expected calibration error (ECE) than full-parameter FT (Fig. 6). Lower NLL indicates better probabilistic prediction, while lower ECE indicates closer agreement between confidence and empirical accuracy under this binning scheme. Confidence distributions for correct and incorrect predictions provide complementary descriptive evidence in Sec. E.1; they do not by themselves establish sample-wise confidence correction.

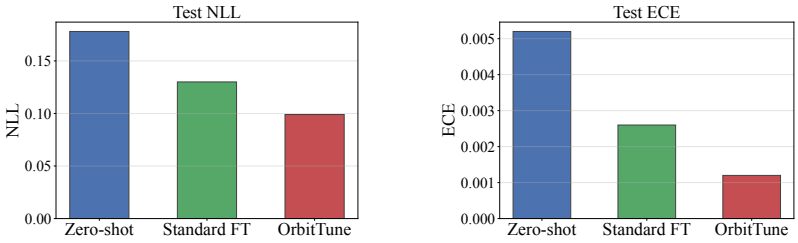


Figure 6: **Calibration on SCM-generated test sets.** OrbitTune reduces NLL (left) and 15-bin ECE (right) compared with full-parameter FT.

3.5 Performance on Real Tabular Benchmarks

We evaluate LimiX-16M and TabPFN-2.5 on TabZilla (McElfresh et al., 2023) and TabArena (Erickson et al., 2025), comparing zero-shot inference, full-parameter FT (Base FT), OrbitTune, and augmentation

baselines. We also include XGBoost, CatBoost, AutoGluon, and neural tabular baselines. The controlled backbone comparisons use single-model evaluation. Baseline configurations and the limits of inference-budget comparability are described in Secs. H.2 and H.4; regression results are in Sec. F.1.

Evaluation. TabZilla results report mean accuracy, AUC, macro-F1, and average within-dataset ranks (lower is better). TabArena classification results use a shared local Bradley–Terry Elo fit over 29 datasets and 14 methods, following TabArena's metric choices: AUC for binary tasks and log loss for multiclass tasks. Higher Elo indicates stronger relative performance within this method set. The 95% dataset-bootstrap intervals quantify uncertainty from dataset composition. Aggregation, bootstrap details, and differences from the official leaderboard are in Sec. H.3.

TabZilla. In Tab. 1, OrbitTune improves LimiX-16M's accuracy/AUC/macro-F1 from $0.875/0.943/0.822$ to $0.911/0.955/0.874$, and TabPFN-2.5's from $0.877/0.937/0.848$ to $0.906/0.952/0.885$. It also improves all three average ranks over the corresponding Base FT results. Both OrbitTune variants outperform the reported CTGAN, SMOTE, and conventional baselines on all three mean metrics. The component ablation is reported in Sec. E.2, with additional generative baselines in Sec. F.2.

TabArena. In Tab. 2, OrbitTune achieves the two highest classification Elo point estimates: 1265.46 for TabPFN-2.5 and 1218.42 for LimiX-16M. Relative to Base FT, the paired gains are $+205.19$ Elo (95% interval $[144.25, 291.62]$) and $+440.69$ ($[349.62, 562.09]$), respectively. These improvements are consistent with the TabZilla results. The interval for the difference between the two OrbitTune backbones includes zero; additional pairwise comparisons are discussed in Sec. H.3. The results support improved downstream performance on the evaluated tasks, while the SCM diagnostics provide complementary evidence about training dynamics.

Table 1: Comparison of augmentation strategies for fine-tuning tabular foundation models on the TabZilla benchmark (McElfresh et al., 2023). Performance entries are means across datasets; bold marks the best value within each backbone or baseline group. For each metric, *Rank* denotes the cross-dataset average rank, where lower values indicate better performance.

Backbone	Method	Performance			Average Rank		
		Acc.	AUC	F1	Acc.	AUC	F1
LimiX-16M	Zero-shot	0.854	0.932	0.790	9.500	8.780	9.520
	Full-parameter FT	0.875	0.943	0.822	7.840	7.120	7.220
	FT + CTGAN (Xu et al., 2019)	0.882	0.941	0.835	6.620	6.280	6.740
	FT + SMOTE (Chawla et al., 2002)	0.883	0.946	0.836	6.340	6.040	6.420
	OrbitTune (ours)	**0.911**	**0.955**	**0.874**	**3.660**	**3.700**	**3.700**
TabPFN-2.5	Zero-shot	0.852	0.928	0.785	8.500	8.840	8.480
	Full-parameter FT	0.877	0.937	0.848	7.540	6.580	7.400
	FT + CTGAN (Xu et al., 2019)	0.879	0.936	0.851	5.980	6.600	6.240
	FT + SMOTE (Chawla et al., 2002)	0.870	0.932	0.839	5.160	6.080	5.400
	OrbitTune (ours)	**0.906**	**0.952**	**0.885**	**3.600**	**3.700**	**4.120**
Conventional	XGBoost (Chen & Guestrin, 2016)	0.862	0.928	0.785	9.700	10.040	9.300
	CatBoost (Prokhorenkova et al., 2018)	0.844	0.921	0.769	11.280	11.640	11.020
	AutoGluon (Erickson et al., 2020)	**0.869**	**0.933**	**0.792**	**8.300**	**8.240**	**8.480**
	RealMLP (Holzmüller et al., 2024)	0.824	0.904	0.742	10.980	11.360	10.960

Table 2: Classification Elo on 29 TabArena datasets (14 methods), using AUC for binary tasks and log loss for multiclass tasks. Higher is better; ratings are centered to mean 1000. Intervals are 95% dataset-bootstrap intervals. Bold marks the highest point estimate. Base FT denotes full-parameter FT; these are local ratings, not official leaderboard scores. See Sec. H.3 for the fitting protocol.

Backbone	Method	Elo	95% interval
LimiX-16M	Zero-shot	1012.58	[929.44, 1100.90]
	Base FT	777.73	[677.63, 860.41]
	FT + CTGAN (Xu et al., 2019)	1174.09	[1092.11, 1275.01]
	FT + SMOTE (Chawla et al., 2002)	1121.39	[1055.03, 1200.81]
	OrbitTune (ours)	1218.42	[1152.69, 1311.12]
TabPFN-2.5	Zero-shot	1018.95	[953.28, 1100.97]
	Base FT	1060.27	[983.90, 1153.77]
	FT + CTGAN (Xu et al., 2019)	1121.39	[1056.41, 1206.82]
	FT + SMOTE (Chawla et al., 2002)	1087.93	[1016.77, 1172.39]
	OrbitTune (ours)	**1265.46**	[1185.65, 1386.34]
Conventional	XGBoost (Chen & Guestrin, 2016)	872.57	[768.23, 951.12]
	CatBoost (Prokhorenkova et al., 2018)	786.33	[658.55, 874.31]
	AutoGluon (Erickson et al., 2020)	968.52	[887.53, 1038.98]
	RealMLP (Holzmüller et al., 2024)	514.37	[269.28, 669.39]

4 Conclusion

OrbitTune fine-tunes tabular foundation models by jointly reparameterizing the input and output coordinates of an entire query-context task. Decoded predictions preserve semantic supervision while encouraging consistency across task views without inference-time ensembling. Our theoretical analysis establishes fixed-task target preservation and characterizes Bayes equivariance under task-orbit symmetry. On SCM classification tasks, OrbitTune changes sample-level training dynamics and improves calibration; targeted TabZilla ablations support the contribution of transition samples as training queries and task transformations on their updates. Across both backbones, OrbitTune improves aggregate classification and regression performance over full-parameter fine-tuning. These findings support task-level reparameterization as a practical approach to TFM adaptation. Future work will examine broader model families and directly quantify prediction consistency across task views.

5 Limitations

Although our method demonstrates superior performance compared to standard fine-tuning and other data augmentation techniques, it does require longer training times, especially with early stopping, which may affect its efficiency in time-sensitive applications. Additionally, the strength of the OrbitTune is adjustable and depends on factors like dataset and learning rate, making parameter tuning resource-intensive. Future work will focus on optimizing training efficiency and developing methods for automated selection of affine transformation parameters.

AI use statement

In this work, we used generative AI tools to check grammar and refine the wording of the manuscript. We have not used generative AI tools to develop the research ideas or methodology, design or conduct experiments, analyze experimental results, or derive scientific conclusions. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Sercan Ö. Arik and Tomas Pfister. TabNet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6679–6687, 2021. doi: 10.1609/aaai.v35i8.16826.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 2017.

Magnus Bühler, Lennart Purucker, and Frank Hutter. Causal data augmentation for robust fine-tuning of tabular foundation models. In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025. URL `https://openreview.net/forum?id=EfQv02muYG`.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In T. Leen, T. Dietterich, and V. Tresp (eds.), *Advances in Neural Information Processing Systems*, volume 13, pp. 416–422. MIT Press, 2000.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. doi: 10.1613/jair.953.

Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. ACM. doi: 10.1145/2939672.2939785.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999. PMLR, 2016.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. AutoGluon-Tabular: Robust and accurate AutoML for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-0519.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943. Curran Associates, Inc., 2021.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. TabR: Tabular deep learning meets nearest neighbors. In *International Conference on Learning Representations*, volume 2024, pp. 18209–18249, 2024.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 507–520. Curran Associates, Inc., 2022. doi: 10.52202/068431-0037.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the state of the art in tabular foundation models. arXiv preprint arXiv:2511.08667, 2025.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 26577–26658. Curran Associates, Inc., 2024. doi: 10.52202/079017-0837.

Myung Jun Kim, Leo Grinsztajn, and Gael Varoquaux. CARTE: Pretraining and transfer for tabular learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23843–23866. PMLR, 2024.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17564–17579. PMLR, 2023.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Huai-Hong Yin, Tao Zhou, Jun-Peng Jiang, and Han-Jia Ye. Talent: A tabular analytics and learning toolbox. *Journal of Machine Learning Research*, 26(226):1–16, 2025.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 172692–172722. Curran Associates, Inc., 2025. doi: 10.52202/085713-5748.

Andrei Margeloiu, Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik. TabEBM: A tabular data augmentation method with distinct class-specific energy-based models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 72094–72144, 2024. doi: 10.52202/079017-2302.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 76336–76369. Curran Associates, Inc., 2023. doi: 10.52202/075280-3337.

Soma Onishi and Shoya Meguro. Rethinking data augmentation for tabular data in deep learning. arXiv preprint arXiv:2305.10308, 2023.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 13991–14002. Curran Associates, Inc., 2019.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), 2015. doi: 10.1609/aaai.v29i1.9602.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17044–17056. Curran Associates, Inc., 2020.

Moacir Antonelli Ponti, Lucas de Angelis Oliveira, Mathias Esteban, Valentina Garcia, Juan Martín Román, and Luis Argerich. Improving data quality with training dynamics of gradient boosting decision trees. arXiv preprint arXiv:2210.11327, 2024.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 6638–6648, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847. PMLR, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *International Conference on Machine Learning*, 2026.

Nabeel Seedat, Jonathan Crabbé, Ioana Bica, and Mihaela van der Schaar. Data-IQ: Characterizing subgroups with heterogeneous outcomes in tabular data. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23660–23674. Curran Associates, Inc., 2022. doi: 10.52202/068431-1719.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. arXiv preprint arXiv:2106.01342, 2021.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.746.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447. PMLR, 2019.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 7335–7345. Curran Associates, Inc., 2019.

Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. VIME: Extending the success of self- and semi-supervised learning to tabular domain. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11033–11043. Curran Associates, Inc., 2020.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, Ningbo Dai, Renzhe Xu, Shuyang Li, Tianyang Zhang, Yue He, Yuanrui Wang, Yunjia Zhang, Zijing Xu, Dongzhe Li, Fang Gao, Hao Zou, Jiandong Liu, Jiashuo Liu, Jiawei Xu, Kaijie Cheng, Kehan Li, Linjun Zhou, Qing Li, Shaohua Fan, Xiaoyu Lin, Xinyan Han, Xuanyue Li, Yan Lu, Yuan Xue, Yuanyuan Jiang, Zimu Wang, Zhenlei Wang, and Peng Cui. LimiX: Unleashing structured-data modeling capability for generalist intelligence. arXiv preprint arXiv:2509.03505, 2025.

A Related Work

Tabular learning and tabular foundation models. Tabular prediction has long been anchored by boosted trees such as XGBoost and CatBoost (Chen & Guestrin, 2016; Prokhorenkova et al., 2018); benchmarks show that neural tabular models require careful evaluation and often still compete with strong tree baselines (Gorishniy et al., 2021; Grinsztajn et al., 2022). Neural alternatives include TabNet, SAINT, FT-Transformer/ResNet-style baselines, and retrieval-augmented TabR (Arik & Pfister, 2021; Somepalli et al., 2021; Gorishniy et al., 2021; 2024). Recent tabular foundation models move toward pretrained or in-context predictors, including TabPFN and its later extensions (Hollmann et al., 2023; 2025; Grinsztajn et al., 2025), scalable TabICL variants (Qu et al., 2025; 2026), and broader structured-data pretraining or transfer systems such as LimiX, CARTE, and TabDPT (Zhang et al., 2025; Kim et al., 2024; Ma et al., 2025). These works mainly study architectures, pretraining, scaling, transfer, and benchmark performance; we instead analyze downstream TFM fine-tuning dynamics and regularize them through task-equivalent views.

Training dynamics and data cartography. Example-level training dynamics have been used to study memorization, forgetting, and mislabeled or ambiguous samples (Arpit et al., 2017; Toneva et al., 2019; Pleiss et al., 2020). Dataset Cartography maps examples by confidence and variability (Swayamdipta et al., 2020); Data-IQ extends related diagnostics to tabular subgroup characterization (Seedat et al., 2022); and training-dynamics signals have also been used to improve structured-data quality with gradient-boosted trees (Ponti et al., 2024). We apply these diagnostics to downstream TFM fine-tuning. Our classification experiments examine confidence saturation, concentration of per-example losses, and changes in data-map structure under OrbitTune. These measurements characterize training behavior; they do not by themselves identify sensitivity to task coordinates as its cause.

Calibration and overconfident prediction. Calibration work studies whether predicted probabilities match empirical correctness, including post-hoc calibration methods and metrics (Pakdaman Naeini et al., 2015; Guo et al., 2017). Modern neural networks are often overconfident, and uncertainty quality can degrade under distribution shift (Guo et al., 2017; Ovadia et al., 2019). OrbitTune modifies the fine-tuning objective through transformed task views. We report classification NLL, ECE, and confidence distributions as complementary diagnostics of its predictive behavior, without assuming that confidence saturation necessarily implies miscalibration.

Data augmentation for tabular learning. Augmentation can be viewed through vicinal risk minimization (Chapelle et al., 2000) and includes tabular oversampling such as SMOTE (Chawla et al., 2002), input or representation interpolation such as mixup and manifold mixup (Zhang et al., 2018; Verma et al., 2019), tabular corruption objectives such as VIME (Yoon et al., 2020), and broader empirical studies of tabular augmentation (Onishi & Meguro, 2023). For TFM fine-tuning, CausalMixFT uses SCM-based synthetic samples to improve robustness and validation-based stopping under data scarcity (Bühler et al., 2025). OrbitTune constructs coordinate views of an existing supervised task by jointly transforming its inputs and outputs, with predictions decoded before computing the loss.

Equivariance and orbit-based augmentation. Symmetry-based learning includes architectural equivariance (Cohen & Welling, 2016) and group-based augmentation. Chen et al. (2020) interpret augmentation as averaging over group orbits and analyze its statistical effects under distributional invariance assumptions. OrbitTune applies the orbit-augmentation perspective to downstream fine-tuning of existing TFMs: a shared transformation acts on query features, context features, context labels, and training query targets, and a decoded loss places the resulting predictions in the original output coordinates. Its contribution is this fine-tuning formulation together with the empirical study of training dynamics and benchmark performance. Finite stochastic augmentation encourages consistency across task views but does not guarantee exact architectural equivariance.

B Diagnostic Framework for Sample-Level Saturation

This appendix defines the diagnostic framework used to analyze sample-level fine-tuning dynamics. In addition to aggregate training loss, we study how individual training samples evolve during fine-tuning. The goal is to characterize whether standard TFM fine-tuning rapidly drives most samples into a saturated high-confidence regime, and whether the remaining loss becomes concentrated on a small subset of examples.

B.1 Sample-Level Training Dynamics

We begin with classification tasks, where sample-level confidence can be measured by the probability assigned to the true label. Let

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$$

denote the fine-tuning dataset. A tabular foundation model predicts the label of a query instance x_i conditioned on a task context C_i. Let θ_t denote the model parameters at recorded fine-tuning checkpoint t. For a specified query-context view, we define the true-label probability of sample i at checkpoint t as

$$p_{i,t} = p_{\theta_t}(y_i \mid x_i, C_i).$$

Aligned context and query views. In the data-map experiments, full-parameter FT and OrbitTune use the same, aligned context and query views to record sample-level dynamics during training. At each recorded checkpoint, both methods are measured on the same query examples with the same corresponding labeled context views. We align the recorded probabilities by training-sample identity and checkpoint, so each pair of trajectories describes the same sample under a common measurement protocol. This alignment concerns the views used to construct the data maps; the optimization objectives remain those of full-parameter FT and OrbitTune, respectively.

Following Dataset Cartography (Swayamdipta et al., 2020), we characterize each training sample by the trajectory of its true-label probability during fine-tuning:

$$\mu_i = \frac{1}{T}\sum_{t=1}^{T} p_{i,t}, \qquad \sigma_i = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_{i,t} - \mu_i)^2}.$$

Here T is the number of recorded checkpoints. The mean μ_i measures the model's confidence in the true label over training, while σ_i measures the variability of that confidence across the recorded checkpoints. We compute both statistics separately for each method using the aligned trajectories described above. For the targeted ablation in Sec. G.1, *ambiguous* samples are the 30% with the highest variability among samples with confidence $\mu_i > 0.4$; *hard-to-learn* samples have confidence $\mu_i < 0.4$; all remaining samples are *easy-to-learn*. These groups are computed separately for each dataset and method. They describe the measured training trajectories rather than intrinsic properties of the data.

B.2 Confidence Saturation and Loss Concentration

To measure how quickly training samples enter a high-confidence regime, we define the saturation ratio

$$S_t(\alpha) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left[p_{\theta_t}(y_i \mid x_i, C_i) > \alpha\right],$$

where $\alpha \in (0, 1)$ is a confidence threshold. A rapidly increasing $S_t(\alpha)$ indicates that many samples become high-confidence early in fine-tuning.

We next measure whether the loss is broadly distributed across the training set or concentrated on a small subset of samples. Let $\ell_i^t \geq 0$ denote the loss of sample i at checkpoint t. When $\sum_j \ell_j^t > 0$, we define the normalized loss contribution and loss effective sample size as

$$w_i^t = \frac{\ell_i^t}{\sum_{j=1}^N \ell_j^t}, \qquad \mathrm{ESS}_t = \frac{1}{\sum_{i=1}^N (w_i^t)^2}.$$

If the loss is evenly distributed across samples, ESS_t is close to N. If the loss is dominated by a small number of high-loss samples, ESS_t becomes much smaller. This statistic measures concentration of loss values, not per-sample gradient contributions or influence on parameter updates. Those quantities also depend on the model's derivatives and require separate measurements.

B.3 OrbitTune-Induced Dynamic Shifts

The diagnostics above allow us to compare the sample-level dynamics induced by different fine-tuning strategies. Let $\mathcal{E}^{\mathrm{FT}}$ denote the set of samples classified as easy-to-learn under standard fine-tuning, and let $\mathcal{A}^{\mathrm{OrbitTune}}$ denote the set of samples classified as ambiguous under OrbitTune. We define the easy-to-ambiguous transition set as

$$\mathcal{S}_{E\to A} = \mathcal{E}^{\mathrm{FT}} \cap \mathcal{A}^{\mathrm{OrbitTune}}.$$

This set contains samples classified as easy-to-learn under standard fine-tuning and as ambiguous under OrbitTune. It describes a shift in the measured confidence trajectories; by itself, it does not establish that the shifted samples supply more useful gradients.

To examine whether such samples are useful rather than merely different, we conduct targeted ablations. For an intervened setting, we measure the signed performance change relative to OrbitTune-full:

$$\Delta_{\mathrm{intervene}} = \mathrm{Perf}(\text{intervened setting}) - \mathrm{Perf}(\text{OrbitTune-full}).$$

For higher-is-better performance metrics, a negative $\Delta_{\mathrm{intervene}}$ denotes degradation, with more negative values indicating larger decreases. If excluding $\mathcal{S}_{E\to A}$ from the query pool causes a larger degradation than excluding the same number of samples drawn at random from the entire training set, while leaving the context pool unchanged in both cases, this supports the contribution of the selected samples as training queries. If using the original task view for entire updates associated with transition queries also hurts performance, this supports a contribution from task transformations on those updates. These comparisons do not by themselves establish a general causal account of why OrbitTune improves performance.

B.4 Diagnostic Hypothesis

The diagnostic hypothesis is that training over alternative task views can change the confidence trajectories of samples that become high-confidence under standard fine-tuning. Loss concentration and data maps describe this behavior, while the interventions assess the contribution of selected samples and views. Evidence of a trajectory shift is consistent with this hypothesis, but does not establish exact equivariance or a general mechanism for improved generalization.

C Theory Details and Proofs

This appendix gives the formal proofs for Lem. 2.1 and theorem 2.2. We first restate the notation used in the main text to make the proof self-contained.

Notation recap. A task τ is a distribution P_τ over $\mathcal{X} \times \mathcal{Y}$. A context of size m and a query-label pair are sampled i.i.d. from P_τ:

$$C_m = \{(X_1, Y_1), \ldots, (X_m, Y_m)\}, \qquad (X, Y) \sim P_\tau.$$

We write $Z = (X, C_m)$. An input-output task reparameterization is $r = (g, h)$, where $g : \mathcal{X} \to \mathcal{X}$ acts on covariates and $h : \mathcal{Y} \to \mathcal{Y}$ acts on labels or regression targets. Its action on the query-context pair is

$$S_r Z = (gX, rC_m), \qquad rC_m = \{(gX_j, h(Y_j))\}_{j=1}^m.$$

The transformed task $r \cdot \tau$ is defined by the pushforward distribution

$$P_{r\cdot\tau} = (g, h)_\# P_\tau.$$

For a distribution μ on $\mathcal{Y}$, $h_\#\mu$ denotes its pushforward by h. For classification, this is the corresponding permutation of class probabilities.

Lemma C.1 (Fixed-task pushforward preservation, restated)**.** *Let τ be a fixed downstream task with distribution P_τ over $\mathcal{X} \times \mathcal{Y}$, and let $r = (g, h)$ be a valid input-output reparameterization. Then rC_m and $(gX, h(Y))$ are i.i.d. samples from $P_{r\cdot\tau}$. Moreover, for regular conditional distributions,*

$$\mathcal{L}_{r\cdot\tau}\left(Y' \mid X' = gx, C'_m = rC_m\right) = h_\# \mathcal{L}_\tau\left(Y \mid X = x, C_m\right)$$

for P_τ^{m+1}-almost every (x, C_m).

Proof. The i.i.d. statement follows directly from the pushforward definition: applying the measurable map (g, h) to each i.i.d. draw from P_τ yields i.i.d. draws from $(g, h)_\# P_\tau = P_{r\cdot\tau}$.

Because g and h are bijections with measurable inverses, $(x, y) \mapsto (gx, h(y))$ is a bijective coordinate change of the supervised task. Conditioning on $(X' = gx, C'_m = rC_m)$ under the transformed task is therefore equivalent to conditioning on $(X = x, C_m)$ under the original task, followed by applying h to the conditional label law. Hence

$$\mathcal{L}_{r\cdot\tau}\left(Y' \mid X' = gx, C'_m = rC_m\right) = h_\# \mathcal{L}_\tau\left(Y \mid X = x, C_m\right),$$

which proves the claim. □

Theorem C.2 (Bayes equivariance under an invariant task prior, restated)**.** *Let $\mathcal{T}$ be a family of tabular tasks, let Π be a probability distribution over tasks, and choose a symmetry subfamily $\mathcal{K} \subseteq \mathcal{R}$. For each fixed $r = (g, h) \in \mathcal{K}$, assume that r is valid, $r \cdot \tau \in \mathcal{T}$ for every $\tau \in \mathcal{T}$, and Π is $\mathcal{K}$-invariant:*

$$(r\cdot)_\# \Pi = \Pi \qquad \text{for every } r \in \mathcal{K}.$$

Define the posterior predictive distribution under the meta-distribution as

$$\eta(z) = \mathcal{L}(Y \mid Z = z).$$

Then, for P_Z-almost every z,

$$\eta(S_r z) = h_\# \eta(z).$$

For classification this is $\eta_{h(y)}(S_r z) = \eta_y(z)$. For regression with $h(y) = cy + d$ and $\mathbb{E}[Y^2] < \infty$, the squared-loss Bayes predictor $m^\star(z) = \mathbb{E}[Y \mid Z = z]$ satisfies

$$m^\star(S_r z) = h(m^\star(z)).$$

Proof. Let $\varphi(Z, Y)$ be any bounded measurable test function. Under the meta-distribution,

$$\mathbb{E}[\varphi(S_r Z, h(Y))] = \int_{\mathcal{T}} \mathbb{E}_{P_\tau^{m+1}}\left[\varphi(S_r Z, h(Y))\right] d\Pi(\tau).$$

Since $P_{r\cdot\tau} = (g, h)_{\#}P_\tau$, applying g to all covariates and h to all labels is equivalent to sampling directly from the transformed task $r \cdot \tau$. Therefore

$$\mathbb{E}_{P_\tau^{m+1}}\left[\varphi(S_r Z, h(Y))\right] = \mathbb{E}_{P_{r\cdot\tau}^{m+1}}\left[\varphi(Z, Y)\right].$$

Substituting this identity and using $(r\cdot)_{\#}\Pi = \Pi$ gives

$$\mathbb{E}[\varphi(S_r Z, h(Y))] = \int_{\mathcal{T}} \mathbb{E}_{P_{r\cdot\tau}^{m+1}}\left[\varphi(Z, Y)\right] \, d\Pi(\tau) = \mathbb{E}[\varphi(Z, Y)].$$

The last equality is the change of variables $\tau' = r \cdot \tau$ under the invariant prior Π. Since the identity holds for every bounded measurable φ,

$$(S_r Z, h(Y)) \overset{d}{=} (Z, Y).$$

Because S_r is invertible, conditioning on $S_r Z = S_r z$ is equivalent to conditioning on $Z = z$. Hence

$$\mathcal{L}(h(Y) \mid S_r Z = S_r z) = h_{\#}\mathcal{L}(Y \mid Z = z) = h_{\#}\eta(z).$$

The distributional equality $(S_r Z, h(Y)) \overset{d}{=} (Z, Y)$ implies that the conditional law of Y at the transformed view $S_r z$ is the conditional law of $h(Y)$ given $S_r Z = S_r z$. Thus

$$\eta(S_r z) = h_{\#}\eta(z).$$

For classification, h is a label permutation, so the preceding identity gives $\eta_{h(y)}(S_r z) = \eta_y(z)$. For regression with $h(y) = cy + d$, the squared-loss Bayes predictor is the conditional mean:

$$m^\star(S_r z) = \mathbb{E}[h(Y) \mid S_r Z = S_r z] = c\mathbb{E}[Y \mid Z = z] + d = h(m^\star(z)).$$

Applying h^{-1} recovers the original semantic target. □

C.1 Scope for Affine Transformations

Centered sampling is not prior invariance. The condition $\mathbb{E}_{\nu_s}[b_j] = 0$ concerns the distribution of sampled transformations. In contrast, Theorem 2.2 requires $(r\cdot)_{\#}\Pi = \Pi$ for each fixed $r \in \mathcal{K}$. A zero mean does not imply this distributional equality, even after averaging over transformations. For example, if an input coordinate is identically zero, adding an independent $b \sim \mathrm{Unif}[-\beta, \beta]$, with $\beta > 0$, changes its distribution from a point mass to a uniform distribution despite $\mathbb{E}[b] = 0$.

An obstruction for nonzero real translations. Suppose a transformation shifts a real-valued coordinate V by a fixed $b \neq 0$. If a probability prior Π over tasks were invariant under this transformation, then the marginal probability measure

$$\mu(A) = \int P_\tau(V \in A) \, d\Pi(\tau)$$

would satisfy $\mu = (T_b)_{\#}\mu$, where $T_b(v) = v + b$. Let $c = |b|$. Translation invariance forces all disjoint intervals $[kc, (k+1)c)$, $k \in \mathbb{Z}$, to have the same probability p. Since these intervals partition $\mathbb{R}$, countable additivity would give

$$1 = \mu(\mathbb{R}) = \sum_{k\in\mathbb{Z}} p,$$

which is impossible whether $p = 0$ or $p > 0$. No moment assumption is needed. For a feature with missing values, the same argument applies after conditioning on its being observed, provided that event has positive probability and missingness is preserved.

Consequently, no proper task prior in this setting is invariant under the full affine family containing nonzero real translations. Bounding the sampled shifts does not fix the issue: invariance under even one nonzero shift entails invariance under its integer iterates. An improper translation-invariant measure is not a probability prior and cannot be substituted into the stated theorem without a different formulation.

A nonempty scope for exact prior symmetry. For a finite group $\mathcal{K}$ of nominal-category and class-label permutations, an invariant probability prior can be constructed from any probability prior Π_0 on a task family closed under these permutations:

$$\Pi_{\mathcal{K}} = \frac{1}{|\mathcal{K}|} \sum_{r \in \mathcal{K}} (r\cdot)_{\#}\Pi_0.$$

Left multiplication by any element of $\mathcal{K}$ permutes the terms, so $\Pi_{\mathcal{K}}$ is invariant and Theorem 2.2 applies. This construction does not assert that a real benchmark's task prior is already symmetric. More generally, compact-group averaging can use a normalized Haar measure (Chen et al., 2020); a bounded sampling range for affine parameters is not itself a compact transformation group.

What remains exact for affine task views. For any probability prior Π and any fixed valid reparameterization r, let $\Pi_r = (r\cdot)_{\#}\Pi$, and write η_Π for the posterior predictive distribution under Π. Then

$$\eta_{\Pi_r}(S_r z) = h_{\#}\eta_\Pi(z) \qquad \text{for } P_Z^{\Pi}\text{-almost every } z.$$

Indeed, sampling a task from Π and transforming its context and query produces exactly the joint law under Π_r; invertibility then gives the displayed conditional-distribution identity. This compares predictors under two transported priors. If $\Pi_r = \Pi$, this identity reduces to equivariance of the same Bayes predictor. For the implemented affine views, Lem. 2.1 supplies fixed-task semantic preservation without an invariant-prior assumption; their effect on a finitely trained model is assessed empirically.

D Additional Method Details

This appendix section provides the stochastic fine-tuning procedure, a local curvature interpretation of OrbitTune, and the scope under which input-output reparameterizations should be treated as valid.

D.1 OrbitTune Pseudocode

Algorithm 1 OrbitTune input-output task-orbit fine-tuning

Require: Training set $\mathcal{D}_{\text{train}}$, TFM f_θ, schema-specific transformation distribution ν_s, transformation probability λ

for each fine-tuning update **do**
  Sample a query-index batch $I \subseteq \{1, \dots, N\}$ and contexts $\{C_i\}_{i \in I}$
  Sample $r = (g, h) \sim (1 - \lambda)\delta_{(\mathrm{id}_{\mathcal{X}}, \mathrm{id}_{\mathcal{Y}})} + \lambda\nu_s$
  for each $i \in I$ **do**
    Set $x_i' = g x_i$ and $C_i' = \{(g x_j, h(y_j)) : (x_j, y_j) \in C_i\}$
    Compute $\ell_i^r = -\log f_\theta(x_i', C_i')_{h(y_i)}$ for classification
    Compute $\ell_i^r = (h^{-1}(f_\theta(x_i', C_i')) - y_i)^2$ for regression
  end for
  Update θ by backpropagation on $L = |I|^{-1} \sum_{i \in I} \ell_i^r$
end for
Validation/test: use $r = (\mathrm{id}_{\mathcal{X}}, \mathrm{id}_{\mathcal{Y}})$ only

D.2 Local Curvature Interpretation

A local expansion describes how averaging over continuous task views changes the training objective. This calculation does not require the task-prior invariance assumption of Theorem 2.2. Centering the transformation

coordinates cancels a first-order term in the decoded-loss expansion, without establishing exact equivariance or a finite-sample performance guarantee.

Let

$$L_\theta^r(z, y) = \ell^r(f_\theta(S_r z), y)$$

be the decoded supervised loss under reparameterization r. Consider a continuous family of valid reparameterizations r_ϵ with $r_0 = (\mathrm{id}_\mathcal{X}, \mathrm{id}_\mathcal{Y})$. Let $\epsilon = \delta\xi$, where ξ has bounded support, $\mathbb{E}[\xi] = 0$, and covariance Σ_ξ. Assume that $L_\theta^{r_\epsilon}(z, y)$ is twice continuously differentiable near $\epsilon = 0$. For fixed λ, a second-order Taylor expansion as $\delta \to 0$ gives the corresponding local identity-mixture objective

$$\begin{aligned}&(1-\lambda)L_\theta^0(z, y) + \lambda\mathbb{E}_\xi\left[L_\theta^{r_{\delta\xi}}(z, y)\right] \\ &\quad = L_\theta^0(z, y) + \frac{\lambda\delta^2}{2}\,\mathrm{Tr}\left(\nabla_\epsilon^2 L_\theta^{r_\epsilon}(z, y)\big|_{\epsilon=0}\Sigma_\xi\right) + o(\delta^2).\end{aligned}$$

For the affine implementation, the continuous transformation coordinates are the log-scales and shifts $\epsilon = (u, b, v, d)$. Their centered sampling is compatible with the zero-mean condition in this local calculation. Centering the log-scales does not center the multiplicative scales themselves: for $u \sim \mathrm{Unif}[-\rho, \rho]$,

$$\mathbb{E}[\exp(u)] = \frac{\sinh\rho}{\rho} > 1 \qquad (\rho > 0).$$

The nonlinear effects of the exponential parameterization are included in the Hessian with respect to ϵ. Thus, zero-mean sampling removes the first-order term but generally leaves a second-order change.

The leading change is a second-order curvature correction, scaled by the transformation probability λ. It need not be nonnegative: the Hessian of the decoded loss with respect to transformation coordinates can be indefinite. This expansion therefore does not establish a nonnegative regularization penalty or a general reduction in prediction sensitivity. It applies only to small continuous transformations, such as numerical and target affine maps near identity; it does not cover discrete category or label permutations. The experimental ranges $\rho_x = \beta_x = \rho_y = \beta_y = 1$ are finite, so their effects must be assessed empirically rather than inferred from this local approximation.

D.3 Discussion of Validity

OrbitTune relies on valid reparameterizations, not arbitrary perturbations. The output-space transformation h is valid when label or target coordinates are arbitrary representations of the same prediction problem. For classification, label-code permutations are valid for anonymous nominal class IDs. If labels are ordinal, cost-sensitive, or exposed to the model as semantic strings, we set $h = \mathrm{id}_\mathcal{Y}$ unless the corresponding order, cost matrix, or semantic mapping is transformed consistently.

For regression, target affine transformations are valid when they represent changes of target units, offsets, or normalization coordinates. We restrict h to positive affine transformations and use decoded loss to avoid changing the semantic squared-error objective. Validity of a coordinate change does not imply invariance of a probability prior over tasks. As shown in Sec. C.1, the full affine family with nonzero real translations does not satisfy the latter requirement in the stated model. Lem. 2.1 establishes semantic preservation for each valid affine or discrete view, whereas Theorem 2.2 applies only to symmetry subfamilies with an invariant prior. The effectiveness of the finite-sample training objective remains an empirical question.

E Experimental Setup

SCM synthetic data. We construct 100 independent synthetic classification datasets from structural causal models (SCMs) by varying both SCM hyperparameters and random seeds. This gives a controlled but diverse

collection of tasks, rather than a single synthetic benchmark, and allows us to test whether the observed training-dynamics patterns are consistent across different data-generating mechanisms.

For each dataset, variables are generated according to

$$X_j = f_j(\mathrm{Pa}(X_j)) + \epsilon_j,$$

where $\mathrm{Pa}(X_j)$ denotes the parents of X_j in the causal graph and ϵ_j is an exogenous noise term. Here, f_j is modeled as a two-layer multilayer perceptron (MLP) with ReLU activation functions. Across SCM instances, we vary hyperparameters such as graph structure, functional coefficients, noise scale, feature dimensionality, sample size, and the subset of label-relevant variables. Labels are sampled from

$$Y \sim p(Y \mid X_{\mathcal{S}}),$$

where $X_{\mathcal{S}}$ denotes the label-relevant variables. For each dataset, train and test samples are drawn from the same SCM, so that $p(Y \mid X)$ is fixed across splits. This setting isolates fine-tuning dynamics from uncontrolled dataset artifacts or explicit distribution shifts. Unless otherwise specified, SCM results are averaged over all 100 generated datasets.

Backbone and full-parameter fine-tuning protocol. Diagnostic experiments use LimiX-16M unless stated otherwise, and all fine-tuning experiments run on NVIDIA H20 GPUs. Within each controlled backbone comparison, all fine-tuning variants start from the same pretrained checkpoint and update every trainable backbone parameter. The optimizer is AdamW with learning rate 10^{-5}, weight decay 5×10^{-2}, and at most 1000 optimizer updates. One update samples 20% of the training examples as the query-index batch; each query context is sampled from the remaining eligible training examples, up to a context length of 50,000, following the common cap described in Sec. H.1. The intended paired protocol uses identical query/context sampling, optimizer schedule, update budget, split, normalization, and validation metric.

OrbitTune sampling. At each update, one view $r = (g, h)$ is sampled and applied consistently to the full query batch, every associated context, the context labels, and the query targets. With probability $1 - \lambda$ the view is identity; otherwise each eligible numerical column independently samples its own a_j, b_j from the shared ranges, each eligible nominal column independently samples one bijection, and classification samples one label permutation for the complete view (regression instead samples one target affine map). A fresh view is drawn at the next update and transformations are not composed within an update. All submitted experiments use $\lambda = 0.5$ and $\rho_x = \beta_x = \rho_y = \beta_y = 1$ across columns, datasets, and backbones. Validation and test use identity only.

Early stopping protocol. Prior work (Bühler et al., 2025) has suggested that early stopping may be an unreliable model-selection strategy for fine-tuning tabular foundation models. Nevertheless, we include a standard validation-based early-stopping protocol as a practical baseline and control. Concretely, we monitor the validation-set metric during fine-tuning and stop training when no improvement is observed for 10 consecutive evaluations. For the saturation and loss-concentration diagnostics in Sec. 3.1, early stopping is disabled and training runs for all 1000 updates to expose the full trajectory.

Normalization Protocol. A practical issue is interaction with model-internal numerical normalization. If an affine task view is applied before a second z-normalization, the latter can absorb much of the intended coordinate change. We therefore replace only the paired methods' internal z-normalization with one shared external normalization whose statistics are computed from the training split; OrbitTune acts afterward in normalized numerical space. Categorical typing, categorical encodings, and missing-value handling are retained rather than disabled. Full-parameter FT and OrbitTune receive the identical externally normalized representation, so their difference remains the sampled task view.

Table 3: Implementation at a glance.

Item	Setting
Updated parameters	All trainable backbone parameters
Optimizer / learning rate / weight decay	AdamW / 10^{-5} / 5×10^{-2}
Maximum updates / early-stopping patience	1000 / 10 evaluations
Query batch / maximum context	20% of training rows / 50,000
Orbit parameters	$\lambda = 0.5$, $\rho_x = \beta_x = \rho_y = \beta_y = 1$
View sampling	One fresh view per update; no within-update composition
Validation and test view	Identity

Metrics. For TabZilla classification, we report accuracy, AUC-ROC, and macro-F1. TabArena classification uses a shared Elo fit with AUC-based outcomes on binary tasks and log-loss-based outcomes on multiclass tasks; see Sec. H.3. For TALENT regression, we report R^2, RMSE, and average ranks based on R^2; for TabArena regression, we report Elo fitted from dataset-level RMSE comparisons. Within each regression dataset, all methods use the same test rows and target scale. Every RMSE is computed in the original target units after inverting any training-time target standardization. For training-dynamics analysis, we report saturation ratios, sample-level data maps, loss effective sample size, and signed performance changes under sample ablations.

E.1 Additional Calibration Diagnostics

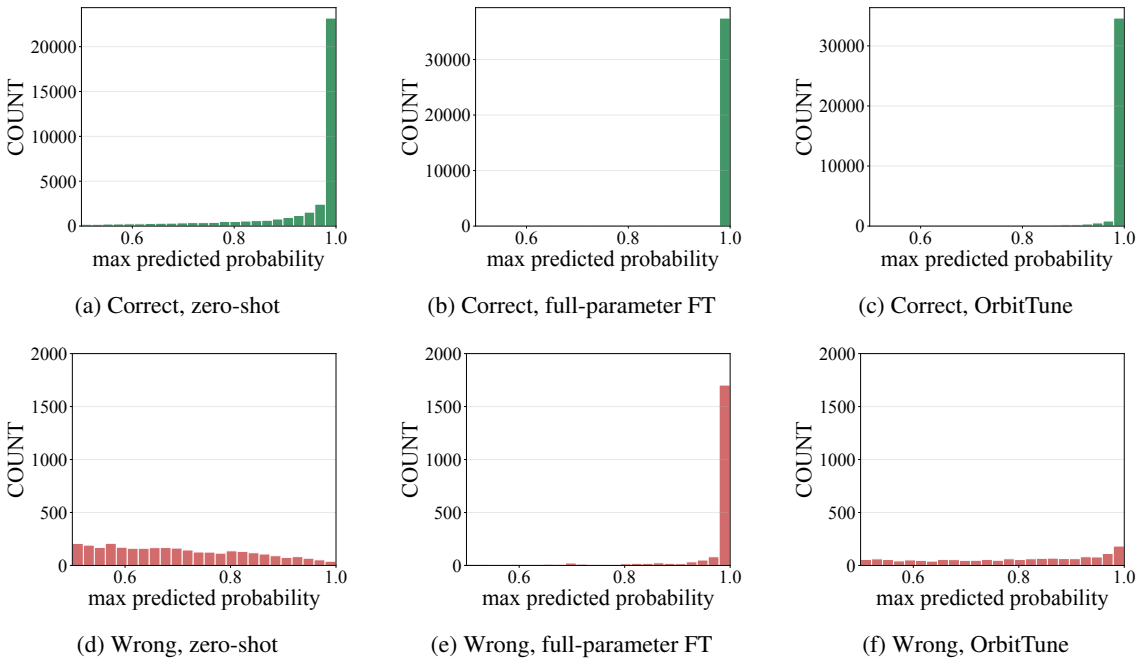
We evaluate probabilistic predictions on SCM-generated test sets using NLL, 15-bin ECE, and the maximum predicted probability. Fig. 7 displays confidence distributions grouped by prediction correctness for zero-shot inference, full-parameter FT, and OrbitTune. The distributions describe lower confidence among incorrect predictions under OrbitTune while retaining high confidence among correct predictions. The plots show counts, so group sizes must also be considered when comparing their shapes. If correctness groups are formed separately for each model, their membership can change after adaptation; these distributions therefore do not establish a within-sample change in confidence. The aggregate NLL and ECE comparison appears in Fig. 6.

E.2 Input–Output Component Ablation

Tab. 4 separates covariate reparameterization g from output reparameterization h. Input-only reparameterization improves AUC and macro-F1 over full-parameter FT, while its mean accuracy decreases from 0.875 to 0.867. Output-only reparameterization improves all three means. Joint input-output reparameterization achieves the highest accuracy, AUC, and macro-F1 among these variants, supporting the use of both components.

Table 4: Input/output component ablation on TabZilla with LimiX-16M. Values are mean performance across datasets.

Variant	Acc.	AUC	Macro-F1
Full-parameter FT	0.875	0.943	0.822
Input only (g, id_Y)	0.867	0.949	0.848
Output only (id_X, h)	0.900	0.944	0.856
Full OrbitTune (g, h)	**0.911**	**0.955**	**0.874**



(a) Correct, zero-shot (b) Correct, full-parameter FT (c) Correct, OrbitTune

(d) Wrong, zero-shot (e) Wrong, full-parameter FT (f) Wrong, OrbitTune

Figure 7: **Maximum predicted probabilities on SCM test samples.** Confidence distributions are shown separately for correct predictions (top) and incorrect predictions (bottom).

F Additional Experimental Results

In Sec. 3.5, we report classification results on real-world tabular benchmarks. This appendix adds regression results on TabArena (Erickson et al., 2025) and TALENT (Liu et al., 2025) for LimiX-16M and TabPFN-2.5.

F.1 Regression Results

We evaluate whether OrbitTune also benefits continuous-label prediction. Tabs. 5 and 6 report results on TabArena and TALENT, respectively, using different reporting protocols.

TabArena regression Elo. We fit Bradley–Terry Elo scores from the recorded RMSE values on the 13 datasets complete for all 10 methods. Each dataset receives equal weight, and each method pair is compared once per dataset, giving 585 comparisons; a lower RMSE wins and exact equality is a tie. Scores are centered at a mean of 1000 over the evaluated method set. This is a local analysis of the available results, not the official TabArena leaderboard, and the regression scores cannot be compared numerically with classification Elo scores. Differences from the official protocol are summarized in Sec. H.3. The 95% intervals in Tab. 5 are percentile intervals from 200 dataset bootstrap samples, with every method on a sampled dataset resampled together and the model refitted each time. They describe uncertainty from dataset composition.

OrbitTune attains Elo scores of 1235.44 on LimiX-16M and 1113.59 on TabPFN-2.5. Relative to the corresponding Base FT scores of 953.80 and 842.61, these are increases of $+281.64$ and $+270.98$ Elo, respectively. Relative to zero-shot inference, the increases are $+154.29$ for LimiX-16M and $+6.56$ for TabPFN-2.5. LimiX-16M with OrbitTune has the highest point estimate among the ten methods; TabPFN-2.5 with OrbitTune improves on Base FT but remains close to its zero-shot result. The table reports intervals for individual ratings. These intervals do not directly provide confidence intervals for rating differences, so we do not infer pairwise statistical significance from the point-estimate gains.

Regression comparability. For TabArena regression, all methods are evaluated on exactly the same test rows and against the same target values within each dataset. All RMSE values are computed in the dataset's original target units, rather than in standardized or augmented target coordinates. If targets are standardized during training, predictions are inverse-transformed before RMSE is computed against the original test targets. Thus, the RMSE-based pairwise outcomes on the 13 TabArena datasets and 10 methods in Tab. 5 compare errors on a common test set and scale within each dataset. Dataset-bootstrap intervals characterize variation across dataset resamples, rather than independent training-seed variability; method-specific feature preprocessing and inference budgets are not assumed to be identical across all baselines.

TALENT regression. On TALENT, OrbitTune improves LimiX-16M from 0.734 to 0.751 in R^2 and reduces RMSE from 0.435 to 0.407, while TabPFN-2.5 with OrbitTune improves over both zero-shot inference and full-parameter FT. It also improves the average R^2 rank over full-parameter FT on each backbone. Together with the classification results in Sec. 3.5, these results support benefits for continuous-label prediction on the evaluated backbones under the reported evaluation settings. These performance results do not characterize analogous training dynamics for regression.

Table 5: Local TabArena regression Elo fitted from RMSE comparisons on 13 common datasets and all 10 methods; higher is better. Methods use identical test rows, and RMSE is computed in each dataset's original target units. Intervals are 95% dataset bootstrap intervals. Bold marks the highest Elo point estimate. Scores are relative to this method set (mean 1000), not official leaderboard scores; see Sec. F.1 for details.

Backbone	Method	Elo	95% interval
LimiX-16M	Zero-shot	1081.15	[987.17, 1208.68]
	Base FT	953.80	[744.88, 1143.29]
	OrbitTune (ours)	**1235.44**	[1115.92, 1391.18]
TabPFN-2.5	Zero-shot	1107.03	[1011.92, 1264.14]
	Base FT	842.61	[679.64, 991.42]
	OrbitTune (ours)	1113.59	[1003.27, 1293.91]
baseline	XGBoost (Chen & Guestrin, 2016)	947.26	[831.88, 1058.58]
	CatBoost (Prokhorenkova et al., 2018)	973.20	[883.36, 1047.88]
	AutoGluon (Erickson et al., 2020)	1126.83	[1066.46, 1242.99]
	RealMLP (Holzmüller et al., 2024)	619.08	[306.35, 741.79]

Table 6: Comparison of fine-tuning strategies for tabular foundation models on the TALENT benchmark. All methods use the same test rows within each dataset, with RMSE computed in its original target units. *Average Rank* is computed from within-dataset R^2 ranks, where lower is better. We report only R^2 ranks because higher R^2 and lower RMSE induce the same ordering on a fixed test set with nonconstant targets. Within each backbone, all fine-tuning methods use the same protocol, without post-hoc ensembling or test-time model aggregation.

Backbone	Method	Performance		Average Rank
		R^2	RMSE	R^2
LimiX-16M	Zero-shot	0.684	0.442	6.071
	Full-parameter FT	0.734	0.435	4.510
	OrbitTune	**0.751**	**0.407**	**2.541**
TabPFN-2.5	Zero-shot	0.721	0.449	4.357
	Full-parameter FT	0.706	0.463	6.592
	OrbitTune	**0.739**	**0.428**	**3.153**
baseline	XGBoost (Chen & Guestrin, 2016)	0.717	0.459	6.224
	CatBoost (Prokhorenkova et al., 2018)	0.708	0.468	7.531
	AutoGluon (Erickson et al., 2020)	**0.730**	**0.446**	**5.000**
	RealMLP (Holzmüller et al., 2024)	0.625	0.550	9.020

F.2 Additional Generative-Augmentation Baselines

We screened TabDDPM (Kotelnikov et al., 2023) and TabEBM (Margeloiu et al., 2024) under the same checkpoint, full-update rule, optimization, split, normalization, and early-stopping protocol. Tab. 7 reports the TabPFN-2.5 TabZilla results. These generators underperform Full-parameter FT in this screening, whereas OrbitTune improves it. This result is limited to the evaluated configurations and does not imply that generative augmentation is universally ineffective.

Table 7: Additional row-generation baselines on TabZilla with TabPFN-2.5. Values are mean performance across datasets.

Method	Acc.	AUC	Macro-F1
Full-parameter FT	0.877	0.937	0.848
FT + CTGAN (Xu et al., 2019)	0.879	0.936	0.851
FT + SMOTE (Chawla et al., 2002)	0.870	0.932	0.839
FT + TabDDPM (Kotelnikov et al., 2023)	0.855	0.917	0.804
FT + TabEBM (Margeloiu et al., 2024)	0.863	0.928	0.812
OrbitTune	**0.906**	**0.952**	**0.885**

G Ablations

G.1 Targeted Transition-Sample Ablation

Objective and transition-set identification. On TabZilla, we test the contribution of transition samples as training queries and of task transformations on their updates. Pilot full-parameter FT and OrbitTune runs produce sample-level data maps using the aligned context/query protocol in Sec. B.1.

For each dataset and each pilot method, we use the mean true-label probability μ_i as confidence and its standard deviation σ_i across recorded checkpoints as variability. We assign samples to three disjoint groups:

- **Ambiguous**: among samples with confidence $\mu_i > 0.4$, select the 30% with the highest variability σ_i.
- **Hard-to-learn**: samples with confidence $\mu_i < 0.4$.
- **Easy-to-learn**: all remaining samples, including those with confidence exactly 0.4.

The variability ranking is computed within the confidence-above-0.4 subset, separately for each dataset and method. We then take the intersection of the easy-to-learn group under full-parameter FT and the ambiguous group under OrbitTune:

$$\mathcal{S}_{E\to A} = \mathcal{E}^{\mathrm{FT}} \cap \mathcal{A}^{\mathrm{OrbitTune}}.$$

We intersect sample identities across the two pilot runs and use this fixed transition set to define the interventions.

Retraining variants. After identifying the transition set, we restart each variant from the same pretrained base checkpoint. The pilot runs provide the sample selection; their adapted parameters are not used to initialize the ablation runs. The backbone, data partitions, normalization, optimizer, update budget, and model-selection protocol follow the controlled setup in Secs. E and H.4. We compare the following settings:

- **Full-parameter FT**: supervised fine-tuning on the full training set without OrbitTune task views, providing the standard adaptation reference.

- **OrbitTune-full**: OrbitTune with all training samples, providing the reference for the signed performance changes.
- **OrbitTune w/o $E \to A$**: exclude the pilot-selected transition samples from the query pool during OrbitTune fine-tuning. These samples remain eligible for inclusion in labeled contexts, so the intervention removes their role as supervised queries while leaving the context pool unchanged.
- **OrbitTune w/o random**: randomly sample the same number of examples from the entire training set of each dataset and exclude them only from the query pool. The context pool is unchanged. This controls for the reduction in query-pool size when comparing random and targeted removal.
- **OrbitTune with original $E \to A$**: retain all samples in both pools and replace transformed views with original views for the updates associated with transition queries. Replacement is applied to the entire update: the identity transformation is used for the full query batch, every associated context, context labels, and query targets.

Evaluation and interpretation. All variants are evaluated on the held-out test split using the original task view. For accuracy, AUC, and macro-F1, where higher values indicate better performance, we report

$$\Delta_{\text{variant}} = \text{Perf}(\text{variant}) - \text{Perf}(\text{OrbitTune-full}),$$

so zero denotes the OrbitTune-full reference and negative values denote degradation. Fig. 5 shows the overall performance comparison on the left and the signed changes on the right. Targeted query removal produces more negative changes than equally sized random query removal, supporting the usefulness of the selected samples as supervised queries beyond the effect of query-pool size alone. Original-view replacement yields changes closer to zero than targeted removal, indicating that retaining these samples as queries recovers part of their contribution. Its comparison with OrbitTune-full assesses the additional contribution of task transformations on the corresponding updates. These results provide evidence about query and task-view usefulness under the evaluated protocol; they do not identify a formal causal effect or establish that every transition sample is individually beneficial.

H Benchmark Details

This appendix provides additional details on the real tabular benchmarks used in our experiments, including benchmark sources, task types, evaluation protocol, and implementation details.

H.1 Benchmark Sources

We evaluate OrbitTune on real-world tabular prediction tasks collected from established tabular benchmark suites.

Maximum context length. For every dataset used in this study, we cap the context at 50,000 labeled training rows to accommodate the context-length limit of the LimiX-16M configuration used in our experiments. We apply the same cap across the evaluated TFM backbones and adaptation variants. This is an upper bound on the number of context rows, not a requirement that each dataset contain 50,000 examples; the actual context can be smaller when fewer eligible training rows are available. Query rows are excluded from their own labeled contexts.

Classification benchmarks. The classification benchmark suite is constructed from TabArena (Erickson et al., 2025) and TabZilla (McElfresh et al., 2023). The tasks include binary and multi-class classification problems with heterogeneous feature spaces. The experimental design follows the benchmark-specific train–validation–test splits and evaluation metrics. For each TabArena dataset, all compared methods use

the same training, validation, and test partitions and are evaluated on exactly the same held-out test samples. Unless otherwise specified, classification performance is reported using the standard metric adopted by each benchmark. In Sec. 3.5, we aggregate results across datasets to compare standard fine-tuning, OrbitTune-enhanced fine-tuning and other baselines.

Regression benchmarks. The regression benchmark suite is constructed from TabArena (Erickson et al., 2025) and TALENT (Liu et al., 2025). Although our training-dynamics analysis focuses on classification, regression tasks provide complementary evidence about downstream performance on continuous targets. TabArena regression uses the same training, validation, and test partitions across all compared methods. On both regression benchmarks, methods are evaluated on the same test rows within each dataset, and every reported RMSE is computed in that dataset's original target units. Any target standardization used during training is inverted before evaluation; method-specific feature preprocessing may still differ. TabArena regression uses RMSE-based Elo on 13 datasets and 10 methods; TALENT reports R^2, RMSE, and average ranks based on R^2. Further evaluation details are in Sec. F.1.

H.2 Evaluated Tabular Foundation Models

We evaluate OrbitTune on two representative tabular foundation models: LimiX-16M (Zhang et al., 2025) and TabPFN-2.5 (Grinsztajn et al., 2025) in Sec. 3.5. These two backbones allow us to assess OrbitTune beyond a single pretrained model, while broader architectural coverage remains future work.

For the classification comparisons, we evaluate zero-shot inference and the following four fine-tuning variants:

- **Standard fine-tuning**: full-parameter fine-tuning under the matched normalization and optimization setup, without OrbitTune task views.
- **FT+CTGAN**: fine-tuning with CTGAN augmentation (Xu et al., 2019).
- **FT+SMOTE**: fine-tuning with SMOTE augmentation (Chawla et al., 2002).
- **OrbitTune-enhanced fine-tuning**: the same backbone and fine-tuning setup, but with the proposed input-output task reparameterization applied during fine-tuning.

The controlled backbone comparisons keep architecture, initialization, and optimization fixed to isolate OrbitTune. Matching data partitions and test samples across TabArena methods does not imply matching all method-specific preprocessing or inference budgets; see Sec. H.4.

Other baselines. We include XGBoost (Chen & Guestrin, 2016), CatBoost (Prokhorenkova et al., 2018), AutoGluon (Erickson et al., 2020), and RealMLP (Holzmüller et al., 2024) as conventional baselines on TabZilla and TabArena. CausalMixFT (Bühler et al., 2025) is not included because a reliable reproduction was not available for our experiments.

H.3 Classification Aggregation and Elo Comparisons

TabZilla aggregation. For each metric, performance is averaged across datasets. Methods are ranked within each dataset, and those ranks are then averaged across datasets; lower average ranks indicate stronger performance. The main table reports mean scores.

TabArena classification Elo. We fit Bradley–Terry Elo ratings from dataset-wise pairwise outcomes on 29 classification datasets and 14 methods. Higher AUC wins on binary tasks, lower log loss wins on multiclass tasks, and exactly equal metric values count as ties. Binary and multiclass tasks contribute to one shared classification fit through their within-dataset outcomes, rather than by averaging different raw

metrics. Each dataset receives equal weight. Ratings are centered to mean 1000, with higher values indicating stronger relative performance within the compared method set. These local ratings are not official TabArena leaderboard scores and cannot be compared numerically with separately fitted ratings.

We obtain 95% percentile intervals from 200 dataset-level bootstrap refits. The available per-dataset records do not establish independent seed or fold repetitions, so these intervals quantify dataset-composition uncertainty rather than seed variability.

Relation to the official TabArena protocol. Our classification and regression analyses follow TabArena's task-specific metrics and Bradley–Terry framework, but use local fits over the methods and datasets reported here. The official leaderboard anchors its ratings to a default Random Forest at 1000, whereas we center ratings at a mean of 1000. TabArena's paper describes averaging outer-fold errors before comparing methods; the official implementation at the revision inspected here retains split-level pairwise outcomes, with equal total weight per dataset.[1] Our fits instead use one recorded metric value per method and dataset. These differences, together with the evaluated method and dataset subsets, preclude direct numerical comparison with official leaderboard scores (Erickson et al., 2025).

Additional classification comparisons. Both OrbitTune backbones exceed their zero-shot variants in the paired Elo comparisons. Base FT alone is not uniformly beneficial: LimiX-16M falls from 1012.58 Elo in zero-shot inference to 777.73 with Base FT. OrbitTune has higher point estimates than CTGAN and SMOTE on both backbones. However, the LimiX-minus-TabPFN OrbitTune difference is -47.04 Elo with interval $[-192.78, 83.05]$, which includes zero and does not establish a consistent advantage for either backbone. Both OrbitTune variants also have higher point estimates than the conventional learners; AutoGluon is the strongest conventional baseline at 968.52 Elo.

H.4 Implementation Details

For all experiments, OrbitTune uses the same downstream training data as standard fine-tuning and does not introduce additional labeled samples. The augmentation is applied only to the training split. Validation and test samples remain unchanged. This design ensures that performance differences reflect improved fine-tuning behavior rather than changes to the evaluation distribution.

For each TFM backbone, we keep the model-specific fine-tuning configuration fixed between the standard fine-tuning baseline and the OrbitTune-enhanced variant. Unless otherwise specified, we use the same optimizer, learning rate schedule, batch size, number of fine-tuning steps, early-stopping criterion, and random seed.

Non-ensembled evaluation. The controlled LimiX-16M and TabPFN-2.5 comparisons use single-model evaluation without post-hoc ensembling, checkpoint averaging, or test-time aggregation. This controlled setting does not establish equal inference budgets across all other baseline methods.

Reproducibility. All TabArena methods share the same data splits and test samples, as specified in Sec. H.1. The intended paired protocol also matches random seeds across compared methods. The main classification tables report cross-dataset means and ranks for TabZilla, and dataset-bootstrap Elo intervals for TabArena; independent repeated-run variability is not established by the TabArena exports. The complete list of datasets, task types, metrics, and hyperparameter configurations will be released together with the code.

[1] Official implementation inspected at TabArena commit `90fa5f66`; see `evaluation/subset_grid.py` and `bencheval/elo_utils.py`.