

Optimizing Pre-Training of Tabular Foundation Models by Shaping Geometry

Anonymous Authors¹

Abstract

Tabular foundation models (TFMs) are trained to solve new supervised learning problems in-context, despite not knowing the downstream table, feature distribution, or label rule during pre-training. This suggests that successful classifier TFMs might learn broad, well-spread intermediate representations that later layers can adapt to task-specific decision boundaries. We test this hypothesis in supervised TabPFN-style pretraining and find that it does not hold: while the model learns in-context prediction, its intermediate embeddings do not naturally become more isotropic or well dispersed. We therefore introduce row- and column-wise Sketched Isotropic Gaussian Regularization (SIGReg) for tabular transformer hidden states and use it to directly shape representation geometry. Early-layer SIGReg reorganizes the model across depth, producing better-distributed early embeddings while preserving later layers for task-specific decision structure. This improves convergence speed, classification performance, and seed-to-seed stability with little additional overhead.¹

1. Introduction

Tabular foundation models (TFMs), such as TabPFN and TabICL (Hollmann et al., 2023; Grinsztajn et al., 2026; Qu et al., 2025), solve new prediction problems in-context: after being trained on synthetic data, they generalize to completely unseen tables, learning the relationship between features at inference time. This leads to a question about representation geometry: what kind of embedding space should a model learn to perform in-context learning?

Because the isotropic Gaussian is theoretically optimal for downstream classification tasks (Balestrierio & LeCun,

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

¹All code is provided in the anonymous repository.

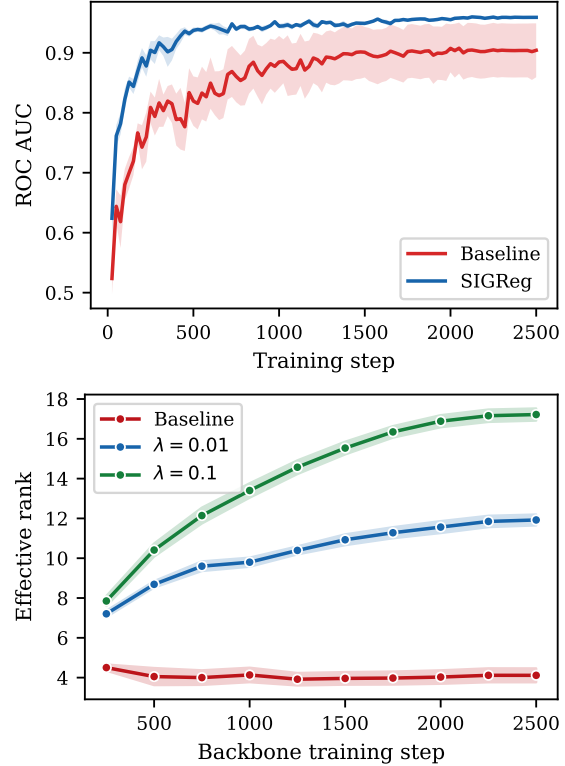


Figure 1. SIGReg improves downstream performance while shaping embedding geometry during pretraining.

2025), one might expect supervised TFM pretraining on many synthetic and randomly generated tables to naturally produce broad, well-spread intermediate embeddings (Shwartz-Ziv et al., 2024) that later layers can adapt to task-specific decision boundaries. We test this hypothesis and find that it does not hold: although the model learns in-context prediction, its embeddings do not naturally approach a well-spread distribution over the course of training.

However, isotropy is not theoretically desirable at every layer. Isotropic embeddings may conflict with the clustered, linearly separable structure needed for classification (Mickus et al., 2024). This suggests a natural depth-wise distinction for TFMs: early layers may benefit from isotropic, high-entropy representations, while later layers should remain free to form task-specific decision geometry.

We therefore use Sketched Isotropic Gaussian Regulariza-

tion (SIGReg) (Balestrieri & LeCun, 2025) as a direct intervention on TFM hidden-state geometry. Our main finding is that early-layer geometric regularization changes how the model uses the depth it has available. SIGReg improves classification performance, convergence speed, and seed-to-seed stability. Linear probing (Alain & Bengio, 2018) and principal component analysis (Hotelling, 1933) visualizations suggest that over the course of training, the regularization creates a better-distributed early representation while leaving later layers free to form task-specific decision structure.

These results suggest that representation geometry is not merely a byproduct of supervised TFM pretraining, but an important design axis for tabular foundation models. We make three contributions: (i) we show that supervised TFM pretraining does not naturally produce isotropic Gaussian intermediate embeddings; (ii) we introduce row- and column-wise SIGReg for tabular transformer hidden states; and (iii) we show that shaping early-layer geometry improves classification pretraining while reorganizing the learned representations across depth.

2. Method

2.1. Overview

We use nanoTabPFN (Pfefferle et al., 2025) which mimics the TabPFN v2 (Hollmann et al., 2025) architecture. The model uses a feature and target encoder; each transformer stack applies attention by features (row) and then attention between datapoints (column). We train on synthetic tabular datasets sampled from the TabPFN prior (Hollmann et al., 2025).

2.2. Gaussian regularizers

VICReg (Bardes et al., 2022) applies separate terms that encourage variance, covariance, and invariance among embeddings. We use VCREg (Zhu et al., 2024), which drops the invariance term. By discouraging covariance among embedding dimensions, VICReg encourages isotropy.

SIGReg (Balestrieri & LeCun, 2025) is the evolution of VICReg, instead using a sliced Epps–Pulley statistic that integrates a closed-form characteristic-function distance to the standard normal distribution over many random 1D projections. SIGReg offers several advantages over its predecessor, including linear time complexity. We therefore direct our focus on SIGReg.

2.3. Applying regularization

Let $\mathbf{H}^{(\ell)} \in \mathbb{R}^{B \times R \times C \times E}$ denote the hidden state at layer ℓ (B tables, R rows, C columns, E embedding dimension). Since there is no single canonical population of embeddings

to regularize, we define two complementary views and regularize each independently. The *column view* groups the R datapoint embeddings within each (b, c) pair; the *row view* groups the C feature embeddings within each (b, r) pair:

$$\mathbf{H}_{\text{col}}^{(\ell)} = \text{reshape}\left(\text{perm}(\mathbf{H}^{(\ell)}, (0, 2, 1, 3))\right) \in \mathbb{R}^{BC \times R \times E}$$

$$\mathbf{H}_{\text{row}}^{(\ell)} = \text{reshape}(\mathbf{H}^{(\ell)}) \in \mathbb{R}^{BR \times C \times E}$$

Each view is passed through its own projection head $g_v : \mathbb{R}^E \rightarrow \mathbb{R}^D$ ($v \in \{\text{c}, \text{r}\}$, two-layer MLP with BatchNorm (Ioffe & Szegedy, 2015)) so that the column and row regularizers can specialize independently. As empirically observed in **Appendix B.2.2**, these projectors are critical. SIGReg then evaluates per group and averages over the BC (resp. BR) groups.

2.4. Training objective

The full training loss is

$$\mathcal{L} = (1 - \lambda_c - \lambda_r) \mathcal{L}_{\text{sup}} + \sum_{v \in \{\text{c}, \text{r}\}} \lambda_v R\left(g_v(\mathbf{H}_v^{(\ell)})\right),$$

where $\mathcal{L}_{\text{sup}}$ is cross-entropy (classification) or bar-distribution NLL (regression). We use the $1 - \lambda$ formation to be consistent with the formulation in LeJEPa (Balestrieri & LeCun, 2025).

3. Results

We observe non-trivial variation across weight initialization and data ordering seeds; by performing 10 independent runs per experiment, we isolate the precise geometric effects of SIGReg. This approach follows established practices in other fields of machine learning, such as LLM research focusing on 1–8B parameter models before being applied at the trillion-parameter scale (Radford et al., 2019). Given the thousands of GPU hours for a single training run of TabPFN-v2, using it would have made it impractical. We recognize this as a limitation of the study. Our evaluation reports averages across multiple tabular datasets.

3.1. TFMs do not naturally become well dispersed

First, we test the hypothesis that the training objective of TFMs naturally encourages them towards the isotropic Gaussian distribution. We use three measures of normality: the Sketched Epps–Pulley (Epps & Pulley, 1983) score (SIGReg without any projector), the effective rank of the embeddings, and the mean per-dimension standard deviation. By all metrics, the baseline model’s embeddings become less Gaussian over the course of training, with **Figure 1** showing that the effective rank of the baseline model’s embeddings after the first layer stays nearly constant across training. The other metrics, across all layers, are presented in **Appendix B.2.1**.

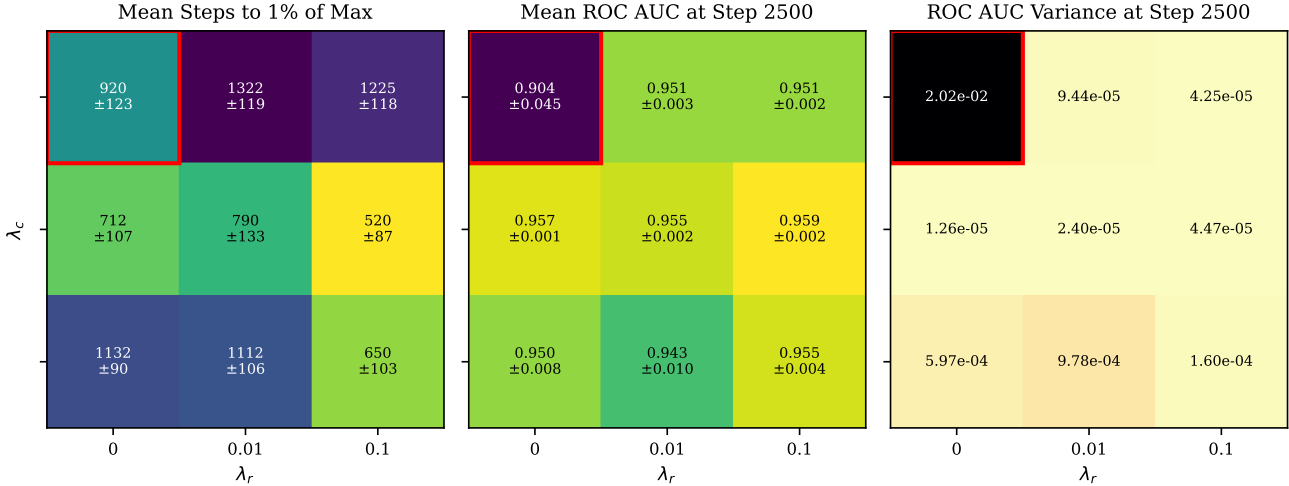


Figure 2. Evaluation performance metrics for SIGReg on classification tasks when doing a grid sweep over λ_c and λ_r .

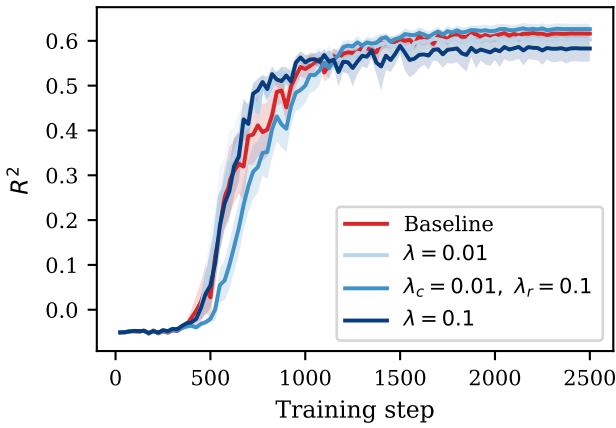


Figure 3. SIGReg on regression.

3.2. Cost of SIGReg

The cost of SIGReg approximately scales as $O(D)$, where D is the embedding dimension of the model. We evaluate the overhead per step (measured in wall clock time) of SIGReg on an Nvidia A40 GPU using the TabPFN-2.5 setup and model dimensions, sweeping each hyperparameter. We consider one and two hidden layer projectors of shape

$$E \rightarrow 4E \rightarrow E/4 \text{ and } E \rightarrow 4E \rightarrow 4E \rightarrow E/4$$

where E is the embedding dimension.

We empirically verify this using a model with the same size and architecture as TabPFN-2.5. As shown in **Table 1**, the SIGReg computation imposes a fixed cost that scales only with embedding dimension, not with model depth, FFN hidden dimension, or attention heads as seen in **Appendix B.1**. Concretely, SIGReg adds an approximately 10-15% overhead per step while reducing total steps to convergence by nearly half (**Figure 2**, left panel).

Embedding Dimension	SIGReg cost (norm.)
48	0.88
96	0.92
192	1
384	1.25
768	1.97

Table 1. Absolute wall clock cost of SIGReg (1-layer) per step scales with embedding dimension. See **Table 2** for complete results.

3.3. Applying SIGReg

Classification. We apply SIGReg on the first layer of the model for classification tasks and observe improvements in convergence speed, final performance, and seed-to-seed stability across a wide range of (λ_c, λ_r) pairs (**Figures 1** and **2**). Results for training with the VCRG loss (Mialon et al., 2024) are given in **Appendix B.2.3**. In contrast, applying SIGReg to the second layer is increasingly harmful for training, consistent with the idea that enforcing isotropic geometry too close to the classifier can interfere with linearly separable task structure (Mickus et al., 2024).

Regression. Regression provides a useful contrast to classification because the desired representation geometry is different. If SIGReg’s classification gains were primarily due to generic optimization benefits, such as stabilizing high-variance batches or improving conditioning, we would expect similar improvements on regression tasks as well. Instead, we observe only limited gains for regression (**Figure 3** and **Appendix B.3**).

This difference is consistent with the structure of the two task families. Classification benefits from representations that support separable decision regions, whereas regression requires estimating a continuous numerical quantity. Useful

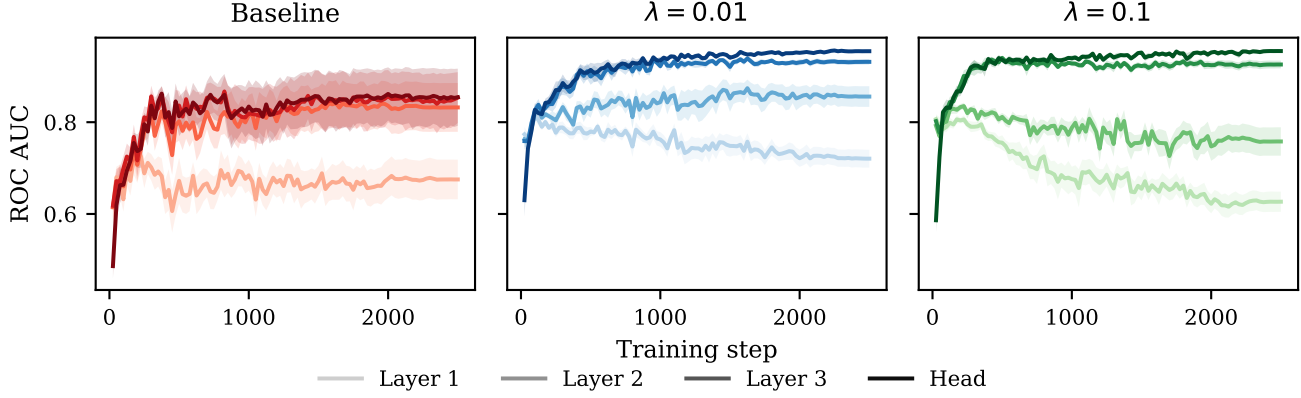


Figure 4. Linear probing results.

regression representations should preserve metric or ordinal relationships between examples: points with nearby target values should remain nearby, or at least similarly ranked, in feature space, reflecting the assumed continuity of the input-output mapping (Gong et al., 2022). Thus, the weak regression results support our interpretation that SIGReg helps classification by shaping a useful early-layer geometry, rather than merely improving optimization stability.

3.4. Embedding Geometry

Linear probing. We apply linear probes (Alain & Bengio, 2018) to the embeddings after each of the layers across training. Figure 4 shows three noteworthy patterns:

- The baseline model’s second layer’s performance is nearly as good as the final classifier head
- With SIGReg, the linear separability of embeddings outputted by earlier layers gets worse over the course of training
- With SIGReg, the linear separability of the embeddings after the third layer stops improving early into training but the head continues to improve in performance.

PCA visualizations. We apply PCA (Hotelling, 1933) to the embeddings after each layer and plot the first two principal components. Color indicates class, while marker type indicates train/test split (dot = train, X = test). Figure 5 shows the circles dataset averaged over all 10 seeds at step 2500 and additional datasets are shown in Appendix B.2.4. Across all three, we see a similar pattern: the first two layers offer little separation between the two classes in the baseline model, while the models with SIGReg scatter embeddings early.

Together, these results suggest that SIGReg redistributes the work across layers: early layers produce broadly distributed representations (even though they are not necessarily linearly separable), leaving later layers free to form

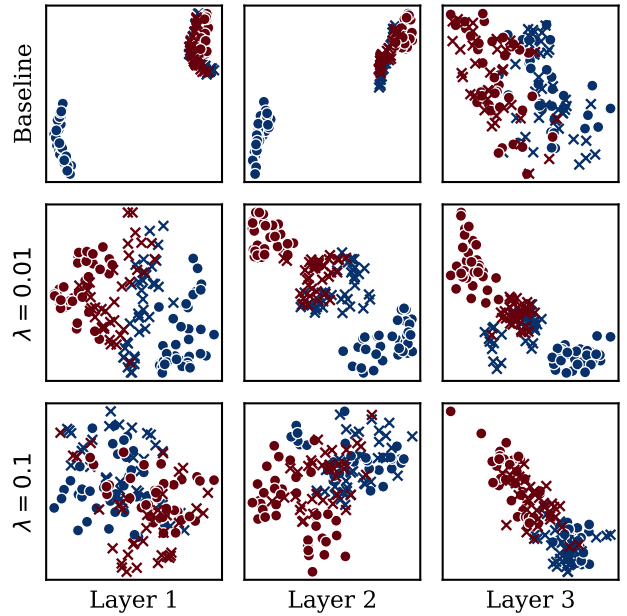


Figure 5. Embeddings projected into first two principal components.

task-specific decision structure, which is a more efficient use of model depth than the baseline achieves.

4. Conclusion

We demonstrate that tabular foundation models, despite being trained to generalize across tasks, do not naturally produce isotropic Gaussian embeddings. However, applying SIGReg at an early transformer layer during pretraining improves downstream classification performance with little added overhead. These findings suggest that explicitly shaping intermediate representation geometry during pre-training is a promising and theoretically grounded direction for improving tabular foundation models. These results also motivate further exploration of other embedding-space methods, e.g. JEPA (Thimonier et al., 2025; Chen et al., 2023).

References

- Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes, 2018. URL <https://arxiv.org/abs/1610.01644>.
- Balestrieri, R. and LeCun, Y. Lejpa: Provable and scalable self-supervised learning without the heuristics, 2025. URL <https://arxiv.org/abs/2511.08544>.
- Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.
- Chen, S., Wu, J., Hovakimyan, N., and Yao, H. Recontab: Regularized contrastive representation learning for tabular data, 2023. URL <https://arxiv.org/abs/2310.18541>.
- Epps, T. W. and Pulley, L. B. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 12 1983. ISSN 0006-3444. doi: 10.1093/biomet/70.3.723. URL <https://doi.org/10.1093/biomet/70.3.723>.
- Gong, Y., Mori, G., and Tung, F. Ranksim: Ranking similarity regularization for deep imbalanced regression, 2022. URL <https://arxiv.org/abs/2205.15236>.
- Grinsztajn, L., Flöge, K., Key, O., Birkel, F., Jund, P., Roof, B., Jäger, B., Safaric, D., Alessi, S., Hayler, A., Manium, M., Yu, R., Jablonski, F., Hoo, S. B., Garg, A., Robertson, J., Bühler, M., Moroshan, V., Purucker, L., Cornu, C., Wehrhahn, L. C., Bonetto, A., Schölkopf, B., Gambhir, S., Hollmann, N., and Hutter, F. TabPFN-2.5: Advancing the state of the art in tabular foundation models, 2026. URL <https://arxiv.org/abs/2511.08667>.
- Gupta, A., Sethi, A., and Kumar, D. TabPFN through the looking glass: An interpretability study of tabPFN and its internal representations, 2026. URL <https://arxiv.org/abs/2601.08181>.
- Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations 2023*, 2023.
- Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6. URL <https://www.nature.com/articles/s41586-024-08328-6>.
- Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Ed. Psych.*, 24:417–441, 1933.
- Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL <https://arxiv.org/abs/1502.03167>.
- Mialon, G., Balestrieri, R., and LeCun, Y. Variance covariance regularization enforces pairwise independence in self-supervised representations, 2024. URL <https://arxiv.org/abs/2209.14905>.
- Mickus, T., Grönroos, S.-A., and Attieh, J. Isotropy, clusters, and classifiers. In Ku, L.-W., Martins, A., and Sriku-mar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 75–84, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.7. URL <https://aclanthology.org/2024.acl-short.7/>.
- Pfefferle, A., Hog, J., Purucker, L., and Hutter, F. nano-tabPFN: A lightweight and educational reimplementation of tabPFN. In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025. URL <https://openreview.net/forum?id=kYUf6HwAsM>.
- Qu, J., Holzmüller, D., Varoquaux, G., and Morvan, M. L. Tabicl: A tabular foundation model for in-context learning on large data, 2025. URL <https://arxiv.org/abs/2502.05564>.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.
- Shwartz-Ziv, R., Balestrieri, R., Kawaguchi, K., Rudner, T. G. J., and LeCun, Y. An information-theoretic perspective on variance-invariance-covariance regularization, 2024. URL <https://arxiv.org/abs/2303.00633>.
- Thimonier, H., Costa, J. L. D. M., Popineau, F., Rimmel, A., and Doan, B.-L. T-jepa: Augmentation-free self-supervised learning for tabular data, 2025. URL <https://arxiv.org/abs/2410.05016>.
- Ye, H.-J., Liu, S.-Y., and Chao, W.-L. A closer look at tabPFN v2: Understanding its strengths and extending its capabilities, 2025. URL <https://arxiv.org/abs/2502.17361>.
- Zhu, J., Evtimova, K., Chen, Y., Shwartz-Ziv, R., and LeCun, Y. Variance-covariance regularization improves representation learning, 2024. URL <https://arxiv.org/abs/2306.13292>.

A. Related Work

Tabular foundation models and representations. TabPFN introduced prior-data fitted transformers that solve new tabular prediction tasks in-context after pretraining on synthetic tasks (Hollmann et al., 2023; 2025). Recent work has begun to analyze the internal representations and mechanisms of these models: (Gupta et al., 2026) probe TabPFN hidden states and find that linear coefficients, intermediate arithmetic quantities, and final-answer signals are decodable across layers, suggesting structured internal computation. (Ye et al., 2025) analyze how TabPFN v2 handles heterogeneous attribute spaces through randomized attribute tokens and in-context inference of attribute relationships, while also documenting limitations on high-dimensional, many-category, and large-scale tasks. Model reports for TabPFN-2.5 further emphasize architectural and scaling changes for larger tables (Grinsztajn et al., 2026). We extend this line of literature by directly regularizing the geometry of TFM hidden states during pretraining.

We encourage readers who are unfamiliar with TFMs to read the excellent explanation and code of nanoTabPFN (Pfefferle et al., 2025).

Representation regularization. Our work builds on regularizers that prevent collapse and encourage useful embedding geometry. The Sketched Isotropic Gaussian Regularization (SIGReg) loss term targets an isotropic Gaussian embedding distribution using a sketched characteristic-function objective, motivated by task-agnostic downstream prediction (Balestriero & LeCun, 2025). VICReg instead uses variance and covariance penalties to avoid collapse in self-supervised learning (Bardes et al., 2022); its supervised variant, often called VCReg, removes the invariance term and regularizes representations to maintain high variance and low covariance (Mialon et al., 2024; Zhu et al., 2024). We adapt these ideas to tabular transformers by applying row- and column-wise regularization to intermediate hidden states.

B. Additional Results

B.1. Additional wall-clock costs when applying SIGReg

The wall-clock time of computing the SIGReg loss when sweeping over all model hyperparameters is shown in **Table 2**. SIGReg has approximately the same absolute cost across all hyperparams except embedding dimension.

B.2. Classification

B.2.1. NORMALITY OF EMBEDDINGS

Figure 6 shows the results of measuring normality across all metrics and all layers. The layer index corresponds to the transformer layer which output the embeddings being measured; the SIGReg loss was calculated on the first layer during training (when not using the baseline model) for all plots.

B.2.2. TURNING OFF THE PROJECTOR

We test removing the projector and applying the SIGReg loss directly to the embeddings. We find that this greatly reduces performance (**Figures 7 and 8**).

B.2.3. USING THE VCReg LOSS

We test the VCReg loss terms and find that they perform equally as well as SIGReg, as shown in **Figures 9 and 10**. In order to compute the loss separately on row and column embeddings as we do with SIGReg, we first fix the hyperparameters from VICReg (Bardes et al., 2022) at $\mu = 25$ and $\nu = 1$, and then effectively remove the invariance term by setting its coefficient $\lambda = 0$. We additionally use a projector in the same manner as with SIGReg, though the projected embeddings for both views are reshaped to $(B \times R \times C, E)$ before computing the variance and covariance losses.

B.2.4. EMBEDDING GEOMETRY USING PRINCIPAL COMPONENT ANALYSIS

Decision boundaries for the embeddings outputted by each layer when sweeping across λ are shown for the linear and moons datasets in **Figures 11 and 12**, respectively.

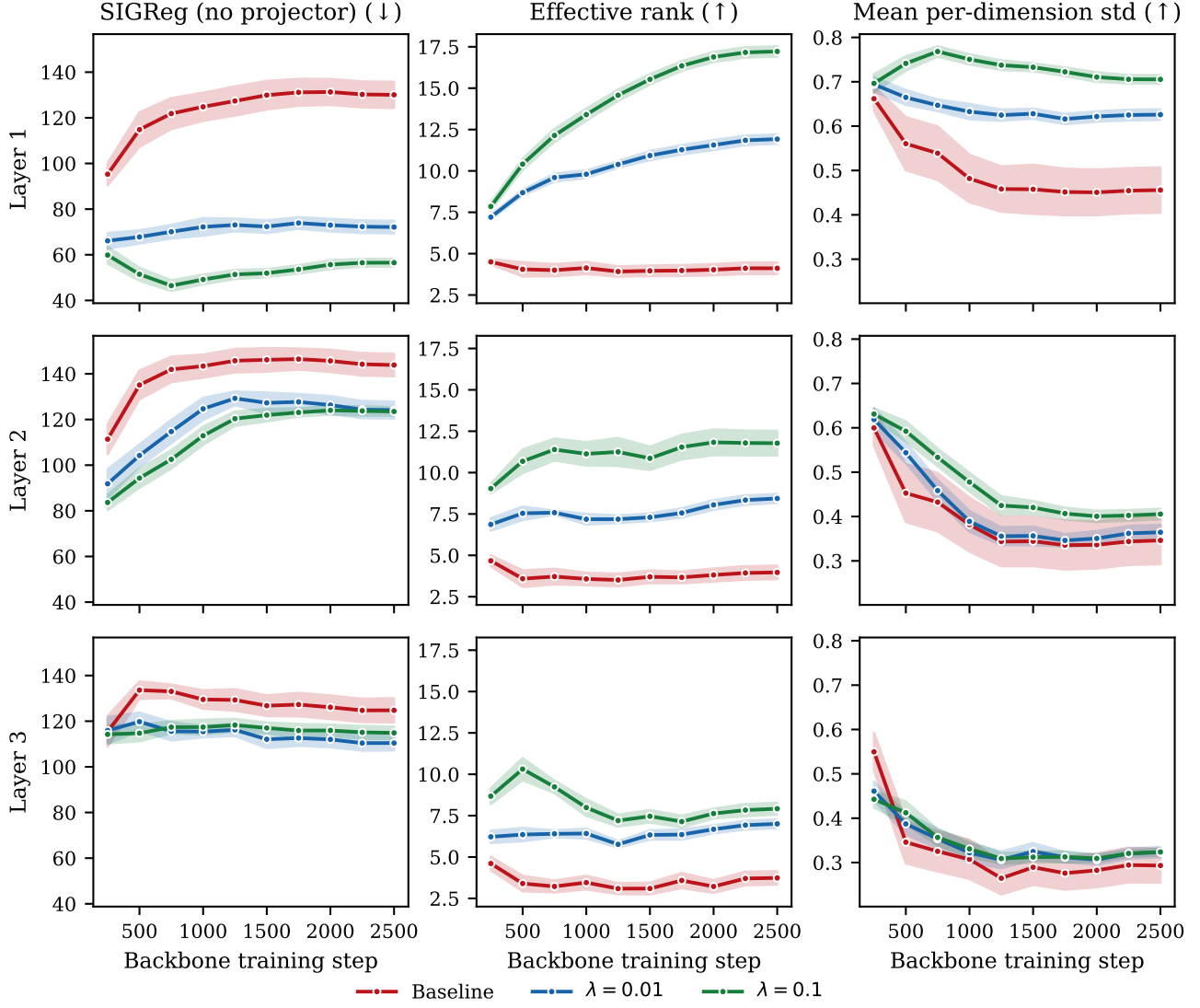


Figure 6. All normality metrics across all layers.

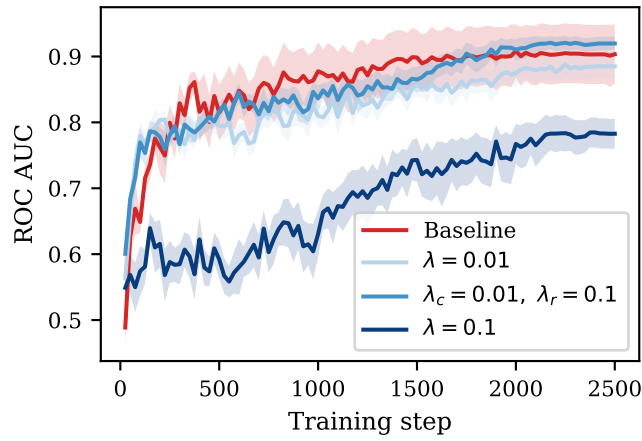


Figure 7. Training with SIGReg for classification without using a linear projector.

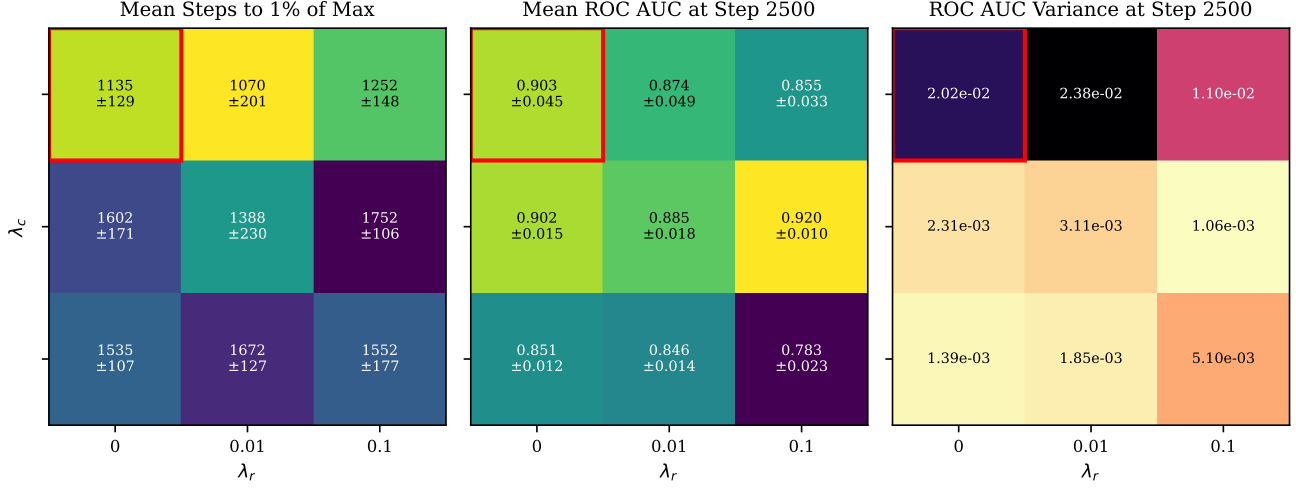


Figure 8. Heatmaps when training with SIGReg for classification without using a linear projector.

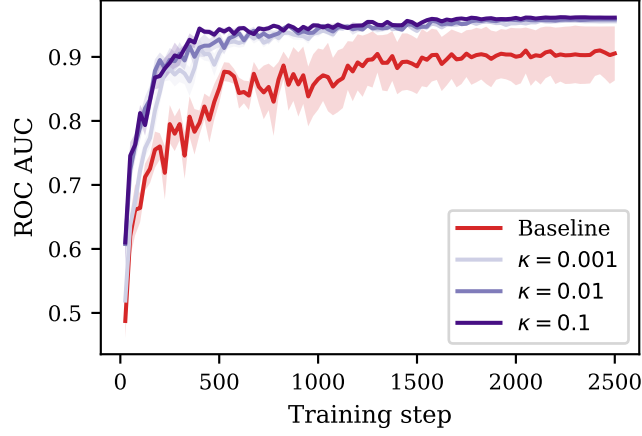


Figure 9. VCReg on classification.

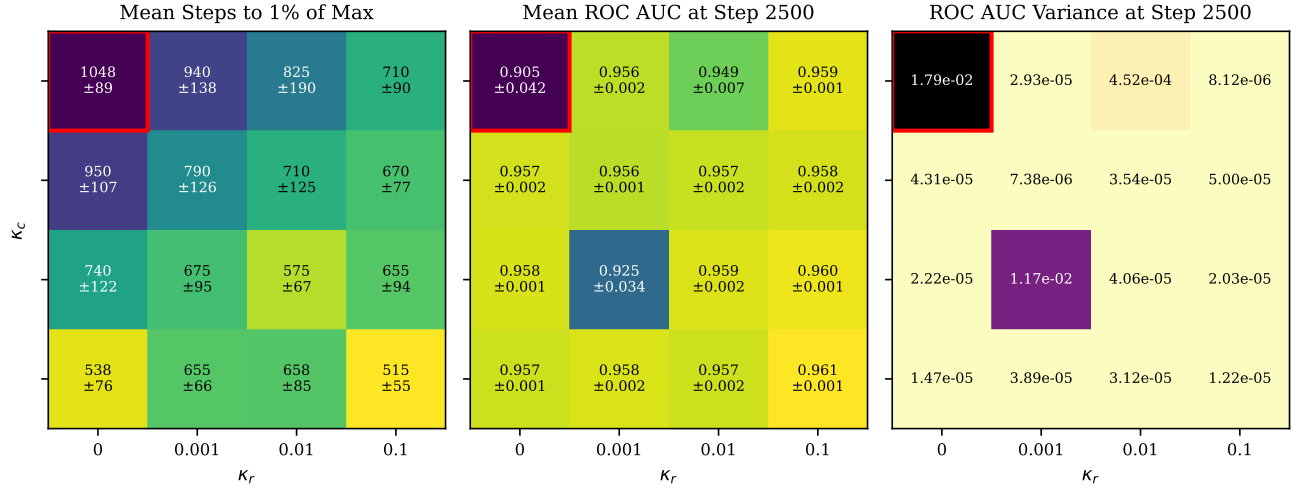


Figure 10. Heatmaps of evaluation metrics for VCReg on classification.

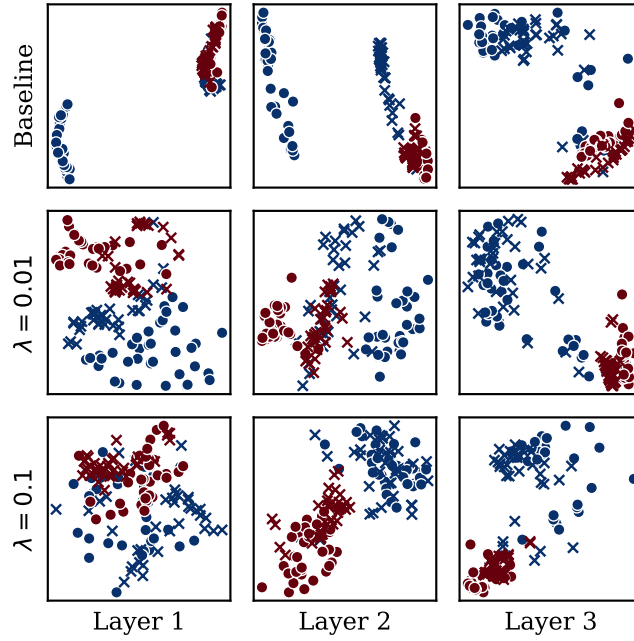


Figure 11. Decision boundaries for the linear dataset.

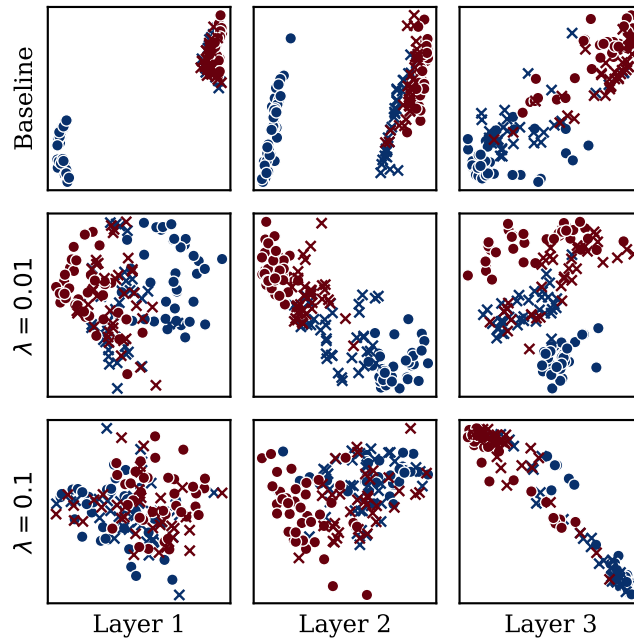


Figure 12. Decision boundaries for the moons dataset.

Optimizing Pre-Training of Tabular Foundation Models by Shaping Geometry

Layers	1-layer SIGReg cost (norm.)	2-layer SIGReg cost (norm.)	Total params	1-layer overhead	2-layer overhead
6	1	1	2.75M	47.3% $\pm$ 0.0%	58.3% $\pm$ 0.1%
12	1	0.99	5.42M	23.8% $\pm$ 0.1%	29.1% $\pm$ 0.0%
<u>24</u>	<u>1</u>	<u>1</u>	<u>10.77M</u>	<u>11.9% $\pm$ 0.0%</u>	<u>14.7% $\pm$ 0.0%</u>
36	0.98	0.99	16.12M	7.8% $\pm$ 0.1%	9.8% $\pm$ 0.0%
48	1	0.99	21.47M	6.0% $\pm$ 0.0%	7.3% $\pm$ 0.0%

(a) Sweep over model depth (other base hyperparameters fixed; default row underlined).

Embedding dim D	1-layer SIGReg cost (norm.)	2-layer SIGReg cost (norm.)	Total params	1-layer overhead	2-layer overhead
48	0.88	0.75	1.37M	20.6% $\pm$ 0.1%	21.6% $\pm$ 0.1%
96	0.92	0.82	3.62M	16.7% $\pm$ 0.0%	18.3% $\pm$ 0.0%
<u>192</u>	<u>1</u>	<u>1</u>	<u>10.77M</u>	<u>11.9% $\pm$ 0.0%</u>	<u>14.7% $\pm$ 0.0%</u>
384	1.25	1.63	35.69M	8.1% $\pm$ 0.0%	13.0% $\pm$ 0.0%
768	1.97	3.65	127.99M	5.5% $\pm$ 0.1%	12.7% $\pm$ 0.0%

(b) Sweep over embedding dimension D (other base hyperparameters fixed; default row underlined).

FFN hidden dim	1-layer SIGReg cost (norm.)	2-layer SIGReg cost (norm.)	Total params	1-layer overhead	2-layer overhead
96	1	1	8.05M	12.7% $\pm$ 0.0%	15.6% $\pm$ 0.0%
192	1	1	8.96M	12.3% $\pm$ 0.0%	15.1% $\pm$ 0.0%
<u>384</u>	<u>1</u>	<u>1</u>	<u>10.77M</u>	<u>11.9% $\pm$ 0.0%</u>	<u>14.6% $\pm$ 0.0%</u>
768	1.01	1	14.39M	11.2% $\pm$ 0.0%	13.7% $\pm$ 0.0%
1536	1.01	1	21.64M	9.7% $\pm$ 0.0%	11.8% $\pm$ 0.0%

(c) Sweep over FFN hidden dimension (other base hyperparameters fixed; default row underlined).

Heads	1-layer SIGReg cost (norm.)	2-layer SIGReg cost (norm.)	Total params	1-layer overhead	2-layer overhead
2	0.98	1	10.77M	14.1% $\pm$ 0.0%	17.6% $\pm$ 0.0%
3	1	1.01	10.77M	13.9% $\pm$ 0.0%	17.0% $\pm$ 0.0%
<u>6</u>	<u>1</u>	<u>1</u>	<u>10.77M</u>	<u>12.0% $\pm$ 0.1%</u>	<u>14.6% $\pm$ 0.0%</u>
12	1	1.01	10.77M	9.5% $\pm$ 0.0%	11.8% $\pm$ 0.1%
24	1	1.01	10.77M	6.7% $\pm$ 0.0%	8.3% $\pm$ 0.0%

(d) Sweep over number of heads (other base hyperparameters fixed; default row underlined).

Table 2. SIGReg wall-clock overhead vs. each architecture hyperparameter, holding the others at the TabPFN-2.5/2.6 base config ($D=192$, FFN=384, 24 layers, 6 heads). Errors are given by ± 1 SE over 10 timed steps.

B.3. Regression

Further evaluation metrics for SIGReg when training and testing on regression tasks are shown in **Figure 13**.

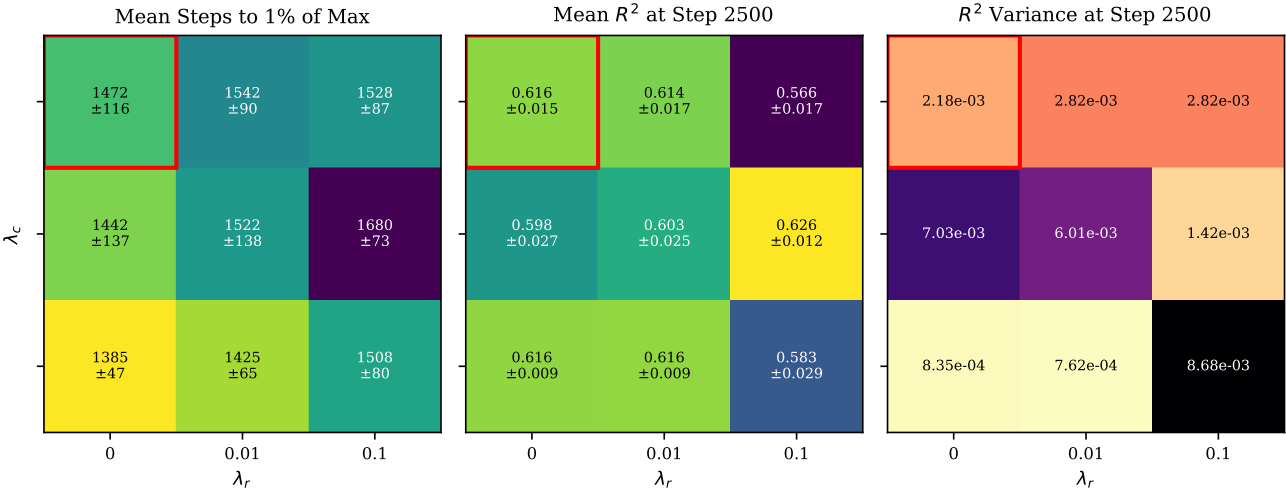


Figure 13. Heatmaps of evaluation metrics for SIGReg on regression.