GENEICL: A TABULAR FOUNDATION MODEL FOR BULK TRANSCRIPTOMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Gene expression is widely measured in biomedicine, yet clinical outcome prediction remains challenging due to high dimensionality, strong feature correlations, and limited labeled data. Large self-supervised transcriptomic foundation models often fail to outperform simple supervised baselines. Tabular foundation models offer an alternative through in-context learning, but are typically pretrained on generic synthetic data rather than transcriptomic structure. We ask whether transcriptomics-aware pretraining, rather than merely scale, is the missing ingredient. Towards this end, we introduce **GeneICL**, a 4.2M-parameter tabular foundation model that combines a semi-synthetic pretraining prior built from experimentally measured bulk expression profiles with a parameter-efficient weight-tied recurrent architecture. We further enable right-censored survival prediction through a training-free reduction to regression using Cox partial-likelihood residuals. We evaluate GeneICL on a benchmark of 80 clinical outcome-prediction tasks spanning classification, regression, and survival. Tabular foundation models consistently outperform self-supervised transcriptomic models, while GeneICL achieves the best overall rank among all evaluated foundation models and tuned baselines. GeneICL does so with up to 387× fewer parameters, no gradient updates at inference, and inference within seconds on a consumer laptop CPU.

1 INTRODUCTION

Gene expression provides a rich molecular readout of biological state and is routinely collected in clinical studies of disease progression and treatment response (Van den Berge et al., 2019). Predictive models built from transcriptomic measurements, therefore, have the potential to support clinically relevant tasks such as forecasting treatment response, estimating prognosis, detecting transplant rejection, and enabling non-invasive diagnostics (Byron et al., 2016). At the same time, these prediction problems are statistically challenging. A typical transcriptomic dataset contains expression measurements for tens of thousands of genes but only tens to hundreds of labeled samples, placing many applications in the high-dimensional, low-sample-size regime (Rezapour et al., 2026).

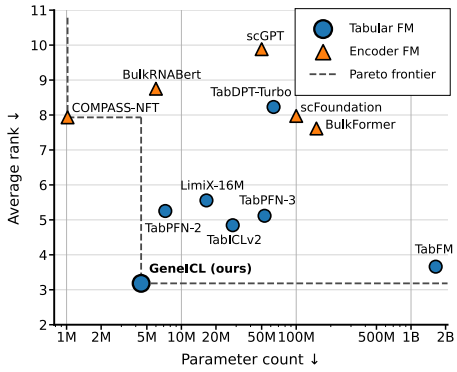

Figure 1: **Parameter–performance trade-off across foundation models.** Performance is the task-macro average rank across classification and regression tasks. The dashed line marks the Pareto frontier. Lower is better on both axes.

Recent transcriptomic foundation models use self-supervised pretraining on large collections of single-cell or bulk gene expression profiles (Theodoris et al., 2023; Cui et al., 2024; Kang et al., 2026). Inspired by the success of large-scale pretraining in other domains, these models aim to learn general-purpose representations that transfer to downstream tasks with limited labeled data. However, the extent to which self-supervised pretraining benefits supervised downstream prediction in transcriptomics remains debated. Independent evaluations have repeatedly found that simple supervised or statistical baselines can match or outperform substantially larger transcriptomic foundation models on downstream

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.

tasks (Boiarsky et al., 2024; Ahlmann-Eltze et al., 2025; Kedzierska et al., 2025; Hou et al., 2026). This raises a basic question: if the goal is prediction from a small labeled gene expression cohort, *is self-supervised representation learning the right pretraining objective?*

Tabular foundation models suggest a different approach. Models such as TabPFN, TabICL, TabDPT, LimiX, and TabFM are pretrained to solve supervised learning tasks through in-context learning (Hollmann et al., 2025; Qu et al., 2026; Ma et al., 2025; Wang et al., 2026; Kong & Das, 2026). At inference time, labeled training examples are provided as context, and predictions for new samples are produced without gradient-based fine-tuning. This paradigm is particularly attractive for transcriptomics, where each cohort naturally defines a small supervised learning problem. However, existing tabular foundation models are designed for generic tables rather than transcriptomic data, and therefore do not explicitly account for the statistical structure of gene expression. Transcriptomic datasets typically contain tens of thousands of strongly correlated features in the $n \ll p$ regime, with substantial redundancy arising from co-regulated gene expression programs. Recent methods such as TabPFN-Wide explicitly address the high-dimensional, low-sample-size setting, but remain designed for generic wide tables rather than transcriptomic data specifically (Kolberg et al., 2025).

We introduce GeneICL, a parameter-efficient tabular foundation model purpose-built for transcriptomic prediction. GeneICL operates in principal component (PC) space and uses a recurrent, weight-tied in-context learner to predict query samples from labeled support examples. A single pretrained model supports both regression and classification. To adapt in-context pretraining to transcriptomics, GeneICL is pretrained on a semi-synthetic prior. Its expression profiles are sampled from a variational autoencoder trained on real bulk RNA-seq data, while its labels are generated by random synthetic functions of these profiles. The inputs thus carry the gene–gene correlation structure of real transcriptomes, while the labels remain as plentiful as in any synthetic prior.

We evaluate GeneICL on a benchmark of prognostic and diagnostic outcomes from bulk transcriptomic cohorts, spanning patient cohorts and cancer cell-line drug screens, including treatment response, prognosis, kidney transplant outcomes, and non-invasive diagnostics. We compare against tuned classical machine-learning methods, transcriptomic foundation models, and general-purpose tabular foundation models, and analyze the contribution of GeneICL's transcriptomics-specific prior and architecture.

Our contributions are threefold:

(i) **A translational outcome-prediction benchmark for bulk transcriptomics.** We systematically compare generic tabular and domain-specialized foundation models on prognostic and diagnostic tasks with direct translational relevance, moving beyond representation benchmarks such as cell-type or tissue-of-origin prediction.

(ii) **GeneICL, a transcriptomics-specific tabular foundation model.** GeneICL pairs a semi-synthetic pretraining prior, whose covariates are generated from a model of over 500,000 experimental RNA-seq profiles, with a looped transformer that applies a single in-context block recurrently in place of a stack of distinct layers. With 4.2M parameters, GeneICL matches or outperforms general-purpose tabular foundation models with up to $387\times$ more parameters (Figure 1).

(iii) **A training-free extension of tabular in-context learning to survival prediction.** We show that regressing on Cox partial-likelihood residuals lets a regression-capable tabular foundation model rank patients by risk without architectural changes or survival-specific pretraining. Our adaptation achieves the best overall rank against existing censoring adaptations for tabular foundation models, SurvPFN, a dedicated survival architecture, and tuned conventional baselines.

2 Related Work

2.1 Transcriptomic foundation models

Large-scale pretraining has produced a growing family of transcriptomic foundation models. Single-cell models such as scGPT, scFoundation, Geneformer, and UCE train transformers with self-supervised objectives on large collections of cellular transcriptomes (Cui et al., 2024; Hao et al.,

2024; Theodoris et al., 2023; Rosen et al., 2026). This paradigm has since been extended to bulk RNA sequencing through models such as BulkRNABert and BulkFormer, as well as generative representation-learning approaches (Gélard et al., 2024; Kang et al., 2026; Pande et al., 2026).

A common feature of these methods is that each transcriptome is encoded independently, without conditioning on other samples or their labels. Supervised downstream prediction therefore typically requires task-specific fine-tuning or an additional predictor. Moreover, independent evaluations have found that pretrained transcriptomic representations do not consistently outperform simpler supervised or statistical baselines (Boiarsky et al., 2024; Ahlmann-Eltze et al., 2025; Kedzierska et al., 2025; Hou et al., 2026). GeneICL takes a different approach: labeled transcriptomes are processed jointly as an in-context learning problem, allowing predictions on a new dataset without task-specific parameter updates.

2.2 Tabular foundation models

Tabular foundation models are pretrained to solve supervised learning problems directly through in-context learning. TabPFN established this paradigm using millions of synthetic prediction tasks (Müller et al., 2021; Hollmann et al., 2025). Subsequent work has extended it through pretraining on real tables, more diverse synthetic priors, and larger-scale architectures (Ma et al., 2025; Zhang et al., 2025; Qu et al., 2026; Grinsztajn et al., 2026; Kong & Das, 2026). TabICL and TabICLv2, which GeneICL builds on, separate row encoding from in-context learning over rows and extend the framework to regression (Qu et al., 2026). Closest to our work, TabPFN-Wide adapts TabPFN-2 to high-dimensional data through continued pretraining on a high-dimensional prior (Kolberg et al., 2025).

The connection between tabular foundation models and transcriptomics is beginning to emerge. General-purpose TabPFN and TabICL models have shown strong performance for perturbation prediction (Palla et al., 2026), while BulkFormer representations combined with TabPFN have been applied to zero-shot prediction from bulk RNA-seq (Tingle et al., 2026). These approaches either use general-purpose tabular models directly or pair them with an independently pretrained transcriptomic encoder. GeneICL instead adapts the pretraining distribution and input representation of the in-context learner to bulk transcriptomic prediction, within a substantially more compact architecture.

General-purpose tabular foundation models natively support classification and regression, whereas many clinical datasets contain right-censored survival outcomes. Prior adaptations to survival analysis rely on specialized architectures and survival-specific priors (Böhm et al., 2026; Seletkov et al., 2026) or target imputation (Vieyra, 2026; Lyu et al., 2026). Instead of imputing censored event times, we regress Cox partial-likelihood residuals, allowing GeneICL to predict survival zero-shot without structural modifications or dedicated retraining.

2.3 Synthetic priors

Prior-data fitted tabular models depend strongly on the distribution of tasks encountered during pretraining. TabPFN generates synthetic tables from structural causal models (Hollmann et al., 2025), while later methods such as TabICLv2 and Mitra broaden the pretraining distribution using more diverse synthetic generators (Qu et al., 2026; Zhang et al., 2025). Recent work has further highlighted the importance of the synthetic prior for downstream behavior (Türkmen et al., 2026). In contrast, TabDPT demonstrates that tabular in-context models can also be pretrained on collections of real datasets (Ma et al., 2025).

2.4 Looped Transformers

In-context prediction can be viewed as learning an algorithm that maps labeled examples and a query to a prediction (Akyürek et al., 2022; Von Oswald et al., 2023). Since many learning algorithms are iterative, recurrent transformer architectures repeatedly apply shared layers across depth (Dehghani et al., 2018). Yang et al. (2024) applied this idea specifically to in-context data fitting, arguing that recurrence provides an inductive bias aligned with iterative learning algorithms.

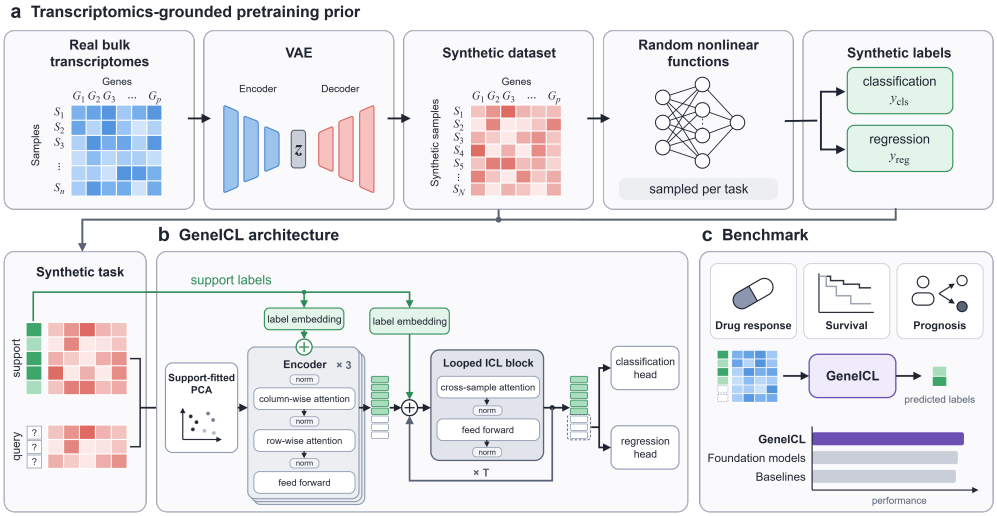

Figure 2: **Overview of GeneICL. (a)** Transcriptomics-grounded pretraining prior. A variational autoencoder (VAE) is trained on unlabeled real bulk transcriptomes. The decoder is used to generate synthetic expression datasets. A randomly sampled nonlinear function maps the individual profiles to a classification or regression label for every sample. Generated datasets are split into a labeled support set and a query set, forming one synthetic pretraining task. **(b)** GeneICL architecture. Transcriptomic profiles are projected with the support-fitted PCA and fed into an encoder. A label embedding of the support labels is added to the support tokens in the encoder and in each iteration of the weight-tied looped in-context-learning (ICL) transformer block. Classification and regression heads read out predictions from the query tokens. **(c)** Benchmark. GeneICL is compared against tabular and transcriptomic foundation models and conventional baselines (performance shown schematically, see Table 1).

More recently, Balef et al. (2026) found substantial redundancy across the depth of tabular in-context learners and showed that repeatedly applying a single transformer layer can retain comparable predictive performance with only a fraction of the original parameters.

3 GeneICL

3.1 Problem formulation

We consider supervised prediction from bulk gene expression in the in-context learning setting. Each task consists of a labeled support set

$$\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_s}, \qquad \mathbf{x}_i \in \mathbb{R}^G, \tag{1}$$

and an unlabeled query set

$$\mathcal{Q} = \{\mathbf{x}_j\}_{j=1}^{n_q}, \tag{2}$$

where $\mathbf{x}$ denotes a bulk gene expression profile over G genes. The target y is categorical for classification tasks and continuous for regression tasks.

Given the support examples, GeneICL predicts the labels of all query samples as

$$\hat{\mathbf{y}}_Q = f_\theta\left(\mathbf{X}_S, \mathbf{y}_S, \mathbf{X}_Q, \tau\right), \tag{3}$$

where $\tau \in \{\mathrm{clf}, \mathrm{reg}\}$ denotes the task type and θ are the pretrained model parameters. The same pretrained model is used for both classification and regression through separate output heads.

Crucially, θ is kept fixed for every downstream task. GeneICL, therefore, adapts to a new dataset solely through the labeled examples provided in context and requires no task-specific parameter updates.

3.2 ARCHITECTURE

GeneICL adopts the two-stage encoder–ICL architecture and default hyperparameter setup of TabICLv2 (Qu et al., 2026), with a few architectural modifications aimed at simplifying the model and improving efficiency. Specifically, we (a) replace full feature mixing with a row-attention bottleneck over `[CLS]` tokens, (b) replace the stack of distinct ICL layers with a single weight-tied block applied recurrently over fixed evidence, (c) omit positional encodings and feature grouping (Appendix H.1), and (d) use a unified model with a shared trunk for classification and regression. As shown in Section 4.3, these changes substantially reduce model size and computational cost without sacrificing predictive performance. Full architectural and training hyperparameters are provided in Appendix Table 5.

Model input. Each dataset is standardized and projected onto its PCs, with both transformations fitted on the support set alone. We use PC scores rather than raw gene features for two reasons:

1. **Rank in the $n \ll p$ regime.** Bulk transcriptomic cohorts contain far fewer samples than genes ($n_s \ll G \approx 20{,}000$). After centering, $\mathrm{rank}(\mathbf{X}_S) \leq n_s - 1$, so all support-set variance lies in a subspace of at most $n_s - 1$ dimensions. Projecting onto the PCs is therefore lossless on the support set, while reducing the number of feature tokens from G to at most $n_s - 1$.
2. **Inductive bias.** Linear models fitted on PC scores are strong baselines in several independent benchmarks of transcriptomic representations (Hou et al., 2026; Elmarakeby et al., 2025). Rather than requiring the transformer to learn this dimensionality reduction from noisy, highly correlated gene features, we build it into the input.

We keep the leading PCs that together explain 90% of the support-set variance, discarding low-variance directions without hurting performance (see Table 4), and embed each PC score into $\mathbb{R}^d$ ($d = 128$) with a linear projection. Support rows are then augmented with label embeddings, which are learned class tokens for classification and linear projections of the normal-score-transformed target for regression.

Encoder. As in TabICLv2, TabPFN-3, and TabFM (Qu et al., 2026; Grinsztajn et al., 2026; Kong & Das, 2026), the encoder alternates between column- and row-attention across three transformer blocks (see Figure 2b). Column attention mixes across samples within each feature using four inducing tokens that query only support rows. Row attention mixes information across features within each sample through four learned `[CLS]` tokens that act as an information bottleneck: features attend exclusively to and from these tokens, bypassing quadratic feature–feature attention without compromising performance (Section 4.3). After the final block, the four 128-dimensional `[CLS]` tokens per sample are normalized and concatenated into a 512-dimensional row representation. We denote the resulting support and query representations by $Z_{\text{supp}} \in \mathbb{R}^{n_s \times 512}$ and $Z_{\text{query}} \in \mathbb{R}^{n_q \times 512}$, respectively; support representations additionally receive an additive second label embedding.

Looped in-context learner. Inspired by work linking recurrence to in-context learning (Yang et al., 2024; Gatmiry et al., 2024), we replace the stack of distinct ICL layers in TabICLv2 with a single shared Transformer block applied recurrently. While prior tabular models have explored weight sharing across depth (Balef et al., 2026), we follow input-injected recurrent architectures (Geiping et al., 2025) and separate fixed task evidence from an evolving hidden state, which yields stable extrapolation beyond the training depth (Appendix H.2).

Let $\mathbf{E} = [\mathbf{0}; \mathbf{Z}_{\text{supp}}; \mathbf{Z}_{\text{query}}]$ denote the fixed evidence and initialize

$$\mathbf{H}^{(0)} = [\mathbf{T}_{\text{think}}; \mathbf{0}; \mathbf{0}], \tag{4}$$

where $\mathbf{T}_{\text{think}}$ contains eight learned thinking representations. We then iterate

$$\mathbf{H}^{(t+1)} = f_\theta\left(\mathbf{H}^{(t)} + \mathbf{E}\right), \qquad t = 0, \ldots, T-1. \tag{5}$$

Keys and values are restricted to thinking and support positions, so queries attend to the evolving reasoning state and labeled support set but never to one another. We use $T = 8$ steps; final query states are layer-normalized and passed to the corresponding classification or regression head.

3.3 TRANSCRIPTOMICS-GROUNDED PRETRAINING PRIOR

Tabular foundation models are commonly pretrained either on fully synthetic tasks sampled from hand-designed generative priors (Hollmann et al., 2025; Qu et al., 2026), or directly on collections of real-world tables (Ma et al., 2025). GeneICL takes an intermediate approach. We construct a semi-synthetic prior. Input features are sampled from a generative model of real expression profiles, and prediction targets are generated synthetically. This preserves the domain-specific covariate structure while allowing an effectively unlimited number of supervised tasks to be sampled during pretraining.

Transcriptomic data distribution. We derive the input distribution from ARCHS4 (Lachmann et al., 2018), a large-scale collection of uniformly processed public RNA-seq experiments from GEO and SRA. We use `human_gene_v2.latest` (upstream build of 19 May 2026), comprising 547,829 human RNA-seq profiles after preprocessing and removal of all profiles contributing to benchmark tasks, represented over $G = 20{,}021$ genes. The resulting corpus spans diverse tissues, cell types, diseases, and experimental conditions.

Rather than sampling these profiles directly during GeneICL pretraining, we first fit a variational autoencoder (VAE) (Kingma & Welling, 2013) to their expression distribution. See Appendix F.1 for VAE details. Let D_ϕ denote the learned decoder. Synthetic expression profiles are generated as

$$\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{x}_i = D_\phi(\mathbf{z}_i), \qquad i = 1, \ldots, n. \tag{6}$$

This allows us to sample an effectively unlimited number of expression datasets while retaining structure learned from real transcriptomic data, rather than repeatedly drawing from a finite set of observed profiles. The resulting matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ therefore defines a transcriptomics-grounded pretraining distribution rather than a generic synthetic tabular prior. As shown in Section 4.3, pretraining on VAE-generated profiles also yields slightly better mean performance than using real ARCHS4 profiles directly.

Synthetic task generation. For each generated expression dataset, we independently sample a supervised prediction problem. Targets are constructed in a principal-component representation of $\mathbf{X}$, matching the representation used by GeneICL downstream. Let $\mathbf{u}_i$ denote the whitened PCA representation of sample i. We sample a sparse subset $\mathcal{J}$ of principal components, with selection probabilities weighted by their explained variance, and generate latent task scores

$$s_{ik} = g_{\psi_k}(\mathbf{u}_{i,\mathcal{J}}), \qquad k = 1, \ldots, K, \tag{7}$$

where the functions g_{ψ_k} are independently sampled MLPs with randomized architectures, nonlinearities, and weight scales. Resampling both $\mathcal{J}$ and ψ_k for every task yields a broad distribution of prediction functions while preferentially placing task-relevant signal along prominent axes of transcriptomic variation.

Classification and regression targets are constructed from the same latent score functions. For binary classification, the first score s_{i1} is thresholded at a randomly sampled quantile, and for multiclass classification with K classes, the label is the index of the largest score,

$$y_i^{\text{clf}} = \arg \max_{k \in \{1, \ldots, K\}} s_{ik}. \tag{8}$$

For regression, the first latent score s_{i1} is used as the target and corrupted with task-dependent noise to obtain a randomly sampled target signal-to-noise ratio.

Every sampled dataset provides both categorical and continuous views, $\mathbf{y}^{\text{clf}}$ and $\mathbf{y}^{\text{reg}}$, of the same underlying synthetic task. Consequently, a single generated expression matrix can jointly supervise GeneICL's classification and regression capabilities. Further details of the task-function prior and sampling distributions are provided in Appendix F.2.

3.4 TRAINING OBJECTIVE

Each synthetic task provides paired categorical and continuous query labels, $\mathbf{y}_Q^{\text{clf}}$ and $\mathbf{y}_Q^{\text{reg}}$. During pretraining, the task type τ selects which label type is embedded in the support set, but both output heads are supervised on the query samples of every task. We minimize

$$\mathcal{L} = \text{CE}\big(\hat{\mathbf{P}}_Q, \mathbf{y}_Q^{\text{clf}}\big) + \mathcal{L}_{\text{rank}}\big(\hat{\mathbf{P}}_Q, \mathbf{y}_Q^{\text{clf}}\big) + \text{MSE}\big(\hat{\mathbf{y}}_Q^{\text{reg}}, \mathbf{y}_Q^{\text{reg}}\big), \tag{9}$$

where $\hat{\mathbf{P}}_Q$ are the class probabilities returned by the classification head, $\hat{\mathbf{y}}_Q^{\text{reg}}$ the predictions of the regression head, and CE and MSE the cross-entropy and mean squared error averaged over query samples. $\mathbf{y}_Q^{\text{reg}}$ is standardized with support-set statistics.

The ranking term $\mathcal{L}_{\text{rank}} = -\text{AUC}_{\text{binormal}}$ replaces the macro-averaged one-vs-rest AUROC by its binormal approximation, computed in closed form from the first two moments of positive and negative query scores, making it a parametric analogue of the histogram loss (Ustinova & Lempitsky, 2016). For each class c, we model the log-probabilities $\log \hat{p}_c$ of positive (label c) and negative (any other label) query samples as Gaussians with means μ_c^+, μ_c^- and standard deviations σ_c^+, σ_c^-, estimated per task, which gives the closed form

$$\text{AUC}_{\text{binormal}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \Phi\left(\frac{\mu_c^+ - \mu_c^-}{\sqrt{(\sigma_c^+)^2 + (\sigma_c^-)^2}} \right), \tag{10}$$

where Φ is the standard normal CDF and $\mathcal{C}$ the set of classes present in the query set.

3.5 Cox-based tabular survival prediction

For right-censored outcomes (t_i, δ_i), with observed time t_i and event indicator $\delta_i \in \{0, 1\}$, we regress on null-model martingale residuals (Therneau et al., 1990),

$$r_i = \delta_i - \hat{H}_0(t_i), \qquad \hat{H}_0(t) = \sum_{\tau_k \le t} \frac{d_k}{n_k}, \tag{11}$$

where d_k events occur among the n_k support samples at risk at event time τ_k. The residual r_i is the gradient of the Cox partial log-likelihood with respect to the log-risk f_i at $\mathbf{f} = \mathbf{0}$, so predicting it is a functional gradient step from the null model. As $\mathbf{r}_S$ depends only on the support labels, GeneICL uses it as an ordinary regression target, $\mathbf{f}_Q = f_\theta(\mathbf{X}_S, \mathbf{r}_S, \mathbf{X}_Q, \text{reg})$, and $\mathbf{f}_Q$ ranks query patients by risk. Survival curves follow from the Breslow estimator (Appendix G).

4 Results

4.1 Classification and regression benchmarks

We compare GeneICL against 17 methods spanning general-purpose tabular foundation models, self-supervised transcriptomic foundation models, and conventional machine-learning baselines. All methods use the same 5-fold cross-validation splits. Classification performance is evaluated by AUROC, AUPRC, and balanced accuracy, while regression is evaluated by Pearson's r, Spearman's ρ, and R^2. Conventional baselines are tuned separately for each task and fold using 50 iterations of random search with inner cross-validation, and foundation models are evaluated with fixed inference settings.

A clear separation emerges between model families (Table 1). Despite being pretrained directly on transcriptomic data, the self-supervised transcriptomic foundation models consistently trail the strongest general-purpose tabular foundation models and conventional baselines. In contrast, tabular foundation models are competitive with tuned supervised methods, suggesting that pretraining directly for supervised prediction is well matched to the high-dimensional, low-sample-size transcriptomic setting.

GeneICL further improves on this paradigm by combining the tabular in-context learning objective with a transcriptomics-grounded pretraining distribution. The standard GeneICL model achieves the second-best overall average rank across classification and regression, surpassed only by its inference-time ensemble. The ensemble achieves the best overall average rank, the highest AUROC and AUPRC, and the best result on all three regression metrics, while the tuned linear baseline remains strongest in balanced accuracy. Importantly, GeneICL contains only 4.24M parameters: approximately $387\times$ fewer than TabFM (1.65B), the strongest competing general-purpose tabular foundation model on regression. On a consumer laptop CPU, one cross-validation fold takes 7.5 s on average, with peak memory below 5 GB (Appendix E).

Table 1: **Performance across 27 classification and 25 regression tasks.** Best and second-best results per metric are **bold** and underlined, respectively. *Avg. rank* is the mean rank across all six metrics and tasks (lower is better). GeneICL (ensembled) averages predictions across PCA thresholds and support-set subsets (Appendix H.3). Parentheses for GeneICL denote the standard deviation over three seeds in units of the last decimal place.

Model	Params ↓	Avg. rank ↓	Classification ↑			Regression ↑		
			AUROC	AUPRC	Bal. Acc.	Pearson	Spearman	R^2
Tabular foundation models								
GeneICL (ensembled)	4.24M	**4.64**	**0.742**(2)	**0.695**(4)	0.592(6)	**0.528**(1)	**0.537**(1)	**0.274**(5)
GeneICL	4.24M	4.98	0.741(4)	0.689(2)	0.588(3)	0.521(1)	0.533(1)	0.271(4)
TabFM (ensembled)	1.65B	5.65	0.721	0.686	0.577	0.515	0.535	0.268
TabICLv2 (ensembled)	28.0M	7.24	0.712	0.671	0.584	0.495	0.519	0.243
TabPFN-3 (ensembled)	55.7M	7.45	0.726	0.685	0.575	0.488	0.508	0.248
TabPFN-2 (ensembled)	9M	7.77	0.720	0.680	0.566	0.492	0.504	0.241
LimiX-16M (ensembled)	16.5M	8.17	0.705	0.660	0.541	0.497	0.514	0.252
TabPFN-Wide	7M	8.25	0.731	0.692	0.556	–	–	–
TabDPT-Turbo (ensembled)	63.5M	12.43	0.692	0.650	0.537	0.394	0.430	0.100
Self-supervised transcriptomics models								
BulkFormer	147M	11.86	0.683	0.636	0.565	0.437	0.442	0.188
COMPASS-NFT	1.02M	11.96	0.684	0.639	0.564	0.392	0.387	0.154
scFoundation	100M	12.35	0.669	0.630	0.568	0.424	0.426	0.164
BulkRNABert	6.01M	13.48	0.670	0.622	0.542	0.382	0.390	0.146
scGPT	51.3M	15.31	0.613	0.572	0.516	0.366	0.363	0.111
Conventional baselines								
Linear (elastic-net)	–	5.41	0.735	0.694	**0.621**	0.500	0.516	0.262
CatBoost	–	10.74	0.705	0.658	0.547	0.448	0.460	0.207
LightGBM	–	12.20	0.673	0.633	0.520	0.439	0.455	0.202
MLP	–	12.23	0.684	0.640	0.566	0.414	0.431	0.016
Random Forest	–	12.73	0.690	0.638	0.510	0.432	0.444	0.186

4.2 GeneICL can predict survival outcomes

Clinical survival cohorts come with right-censored time-to-event outcomes, which GeneICL handles through the Cox partial-likelihood residual adaptation (Section 3.5). To isolate the effect of this adaptation, we first compare existing strategies for turning tabular foundation models into survival models with GeneICL as fixed backbone (Vieyra, 2026; Kim et al., 2026; Lyu et al., 2026). We also compare GeneICL-Cox against SurvPFN, a tabular foundation model with survival-specific architecture and pretraining (Böhm et al., 2026), and tuned survival baselines. In both settings, GeneICL-Cox achieves the highest aggregate C-index and best D-calibration, while remaining within 0.001 of the best integrated Brier score (Figure 3), without any survival-specific pretraining.

4.3 Ablation study

We first examine whether GeneICL's architectural simplifications come at a cost in predictive performance. Replacing the single recurrent ICL block with eight untied blocks increases the parameter count from 4.24M to 19.25M, yet does not improve any of the evaluated metrics (Table 2). Similarly, replacing the bottleneck feature attention with full self-attention in the encoder increases the feature-axis complexity from $\mathcal{O}(n)$ to $\mathcal{O}(n^2)$ with matched or lower performance. We further analyze the recurrent computation in Appendix H.2, where we vary the number of recurrence steps.

The pretraining distribution mainly affects regression. Replacing the VAE-generated covariates with either the generic TabICLv2 prior or the real ARCHS4 profiles underlying the VAE leaves classification ranking metrics largely unchanged but lowers all regression metrics (R^2 0.238 and 0.248 vs. 0.271, see Table 2). Generating covariates with the VAE therefore costs no performance relative to real data, while providing an unlimited, leakage-free stream of novel expression profiles.

Finally, we investigate inference-time ensembling strategies in Appendix H.3. Across several alternative views of the same prediction problem, generic strategies based on PCA thresholds, support subsets, gene subsampling, and input transformations provide modest gains on several metrics, whereas transcriptomics-specific ensembles based on highly variable genes or Hallmark pathways do not consistently improve performance.

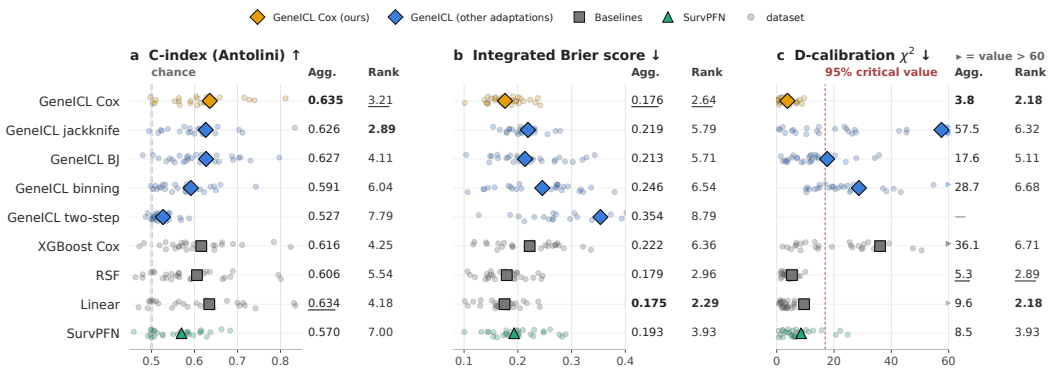

Figure 3: **Survival prediction across 28 SurvBoard datasets.** We compare GeneICL with Cox residual fitting against alternative tabular survival adaptations, SurvPFN, and tuned baselines. Panels show Antolini's C-index (**a**), integrated Brier score (**b**), and D-calibration χ^2 (**c**). Small points denote datasets; larger markers show study-weighted aggregates across TARGET, TCGA, METABRIC, and ICGC. *Agg.* reports this aggregate and *Rank* the mean within-dataset rank; best and second-best values are **bold** and underlined. Dashed lines mark chance-level C-index (0.5) and the D-calibration threshold ($\chi^2 = 16.9$, $p = 0.05$).

Table 2: **Ablation of the main GeneICL design choices**. *Attn.* denotes the asymptotic cost of the feature-axis attention. Parentheses report the standard deviation over three seeds in units of the last decimal place. Best and second-best results are bold and underlined, respectively.

Variant	Params ↓	Attn.	Avg. rank ↓	Classification ↑			Regression ↑		
				AUROC	AUPRC	Bal. Acc.	Pearson	Spearman	R^2
GeneICL	4.24M	$\mathcal{O}(n)$	**2.56**	**0.741**(4)	0.689(2)	0.588(3)	**0.521**(1)	**0.533**(1)	0.271(4)
Untied (8 blocks)	19.25M	$\mathcal{O}(n)$	2.70	0.740(2)	0.689(3)	0.589(1)	0.517(1)	0.531(2)	0.263(6)
Full attention	4.05M	$\mathcal{O}(n^2)$	2.85	0.739(0)	0.688(2)	0.570(6)	0.515(3)	0.530(1)	**0.273**(3)
TabICLv2 prior pretraining	4.24M	$\mathcal{O}(n)$	3.55	0.737(1)	**0.690**(1)	0.576(3)	0.496(6)	0.514(5)	0.238(7)
Real expression data pretraining	4.24M	$\mathcal{O}(n)$	3.33	0.735(3)	0.682(3)	**0.593**(11)	0.505(3)	0.517(4)	0.248(4)

5 Discussion and Conclusion

Transcriptomic foundation model evaluations have largely centered on representation benchmarks such as cell-type or tissue-of-origin prediction, with clinical outcomes covered by only a few tasks. We therefore built a benchmark of 80 translational prognostic and diagnostic tasks, to ask whether self-supervised pretraining helps where it matters clinically. Our results suggest it does not. GeneICL, our 4.24M-parameter tabular foundation model, pretrained directly for supervised prediction, outperforms larger transcriptomic foundation models and matches general-purpose tabular foundation models with up to 387× more parameters (Table 1). Through Cox partial-likelihood residual regression, the same frozen model also predicts survival without retraining. Across all three outcome types, the pretraining objective, rather than model scale, appears to be what limits current models.

Supervised pretraining benefits from tasks that resemble transcriptomic cohorts. Replacing our prior with the generic TabICLv2 prior, or VAE samples with real profiles, lowers regression performance (Table 2), so realistic covariates matter more than fidelity to individual samples. Architecture matters less than data. One looped block matches eight untied blocks with under a quarter of their parameters and the resulting model runs in seconds on a laptop CPU.

Limitations. Our prior is derived from bulk RNA-seq, so single-cell RNA-seq and other omics modalities remain untested. Every task is also evaluated within a single cohort, which leaves robustness to batch and platform shifts between support and query samples open. Finally, whether the small margin over linear baselines reflects a limit of what tens to hundreds of expression profiles can reveal, or a limit of current priors and architectures, is the question we consider most important to answer next.

AI USE STATEMENT

In this work, we used generative AI tools for feedback on experimental design and result interpretation, as well as for manuscript editing, literature assistance, and code and figure refinement. All AI-assisted outputs were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The GeneICL architecture is fully specified in Section 3.2, its training objective is in Section 3.4, and all model and pretraining hyperparameters in Table 5. The pretraining prior is described in full, including the VAE with its training data and hyperparameters (Appendix F.1, Table 6) and the synthetic label-generating process with all of its sampling distributions (Appendix F.2, Table 7). All data used in this work are publicly available: ARCHS4, which serves as the VAE training corpus, is available publicly, and the benchmark tasks can be downloaded from the Gene Expression Omnibus (GEO) under `https://www.ncbi.nlm.nih.gov/geo/` or the publications of available cohorts listed in Table 3, while the survival cohorts come from the public SurvBoard benchmark Wissel et al. (2025).

REFERENCES

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22(8): 1657–1661, 2025.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in medicine*, 24(24):3927–3944, 2005.

Amir Rezaei Balef, Mykhailo Koshil, and Katharina Eggensperger. Is one layer enough? understanding inference dynamics in tabular foundation models. *arXiv preprint arXiv:2605.06510*, 2026.

Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A. Margolin, Sungjoon Kim, Christopher J. Wilson, Joseph Lehár, Gregory V. Kryukov, Dmitriy Sonkin, Anupama Reddy, Manway Liu, Lauren Murray, Michael F. Berger, John E. Monahan, Paula Morais, Jodi Meltzer, Adam Korejwa, Judit Jané-Valbuena, Felipa A. Mapa, Joseph Thibault, Eva Bric-Furlong, Pichai Raman, Aaron Shipway, Ingo H. Engels, Jill Cheng, Guoying K. Yu, Jianjun Yu, Peter Aspesi, Melanie De Silva, Kalpana Jagtap, Michael D. Jones, Li Wang, Charles Hatton, Emanuele Palescandolo, Supriya Gupta, Scott Mahan, Carrie Sougnez, Robert C. Onofrio, Ted Liefeld, Laura MacConaill, Wendy Winckler, Michael Reich, Nanxin Li, Jill P. Mesirov, Stacey B. Gabriel, Gad Getz, Kristin Ardlie, Vivien Chan, Vic E. Myer, Barbara L. Weber, Jeff Porter, Markus Warmuth, Peter Finan, Jennifer L. Harris, Matthew Meyerson, Todd R. Golub, Michael P. Morrissey, William R. Sellers, Robert Schlegel, and Levi A. Garraway. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483 (7391):603–607, March 2012. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature11003. URL `https://www.nature.com/articles/nature11003`.

Samuel Böhm, Lennart Purucker, Frank Hutter, and Pascal Schlosser. Survpfn: Towards foundation models for survival predictions. *arXiv preprint arXiv:2606.04564*, 2026.

Rebecca Boiarsky, Nalini M Singh, Alejandro Buendia, Ava P Amini, Gad Getz, and David Sontag. Deeper evaluation of a single-cell foundation model. *Nature Machine Intelligence*, 6(12):1443–1446, 2024.

Hamid Bolouri, Jason E Farrar, Timothy Triche Jr, Rhonda E Ries, Emilia L Lim, Todd A Alonzo, Yussanne Ma, Richard Moore, Andrew J Mungall, Marco A Marra, et al. The molecular landscape of pediatric acute myeloid leukemia reveals recurrent structural alterations and age-specific mutational interactions. *Nature medicine*, 24(1):103–112, 2018.

Benjamin M Bolstad, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

Philip Brennecke, Simon Anders, Jong Kyoung Kim, Aleksandra A Kołodziejczyk, Xiuwei Zhang, Valentina Proserpio, Bianka Baying, Vladimir Benes, Sarah A Teichmann, John C Marioni, et al. Accounting for technical noise in single-cell rna-seq experiments. *Nature methods*, 10(11):1093–1095, 2013.

Sara A Byron, Van Keuren-Jensen, R Kendall, David M Engelthaler, John D Carpten, and David W Craig. Translating rna sequencing into clinical diagnostics: opportunities and challenges. *Nature Reviews Genetics*, 17(5):257–271, 2016.

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. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Haitham Elmarakeby, Ahmed Roman, Shreya Johri, and Eliezer M Van Allen. Empirical evaluation of single-cell foundation models for predicting cancer outcomes. *bioRxiv*, 2025.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. doi: 10.18637/jss.v033.i01.

Khashayar Gatmiry, Nikunj Saunshi, Sashank J Reddi, Stefanie Jegelka, and Sanjiv Kumar. Can looped transformers learn to implement multi-step gradient descent for in-context learning? *arXiv preprint arXiv:2410.08292*, 2024.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *Advances in Neural Information Processing Systems*, 38:41340–41391, 2025.

Maxence Gélard, Guillaume Richard, Thomas Pierrot, and Paul-Henry Cournède. Bulkrnabert: cancer prognosis from bulk rna-seq based language models. *bioRxiv*, pp. 2024–06, 2024.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *Journal of Machine Learning Research*, 21(85):1–63, 2020.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

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.

Siyu Hou, Penghui Yang, Wenjing Ma, Jade Xiaoqing Wang, and Xiang Zhou. A unified framework enables accessible deployment and comprehensive benchmarking of single-cell foundation models. *bioRxiv*, 2026.

Jeffrey S Hyams, Sonia Davis Thomas, Nathan Gotman, Yael Haberman, Rebekah Karns, Melanie Schirmer, Angela Mo, David R Mack, Brendan Boyle, Anne M Griffiths, et al. Clinical and biological predictors of response to standardised paediatric colitis therapy (protect): a multicentre inception cohort study. *The Lancet*, 393(10182):1708–1720, 2019.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, 2008. ISSN 19326157. URL http://www.jstor.org/stable/30245111.

Boming Kang, Rui Fan, Meizheng Yi, Chunmei Cui, and Qinghua Cui. Bulkformer: A large-scale foundation model for bulk transcriptomes. *Cell Systems*, 2026.

Kasia Z Kedzierska, Lorin Crawford, Ava P Amini, and Alex X Lu. Zero-shot evaluation reveals limitations of single-cell foundation models. *Genome Biology*, 26(1):101, 2025.

Da In Kim, Wei Siang Lai, and Kelly W Zhang. Tabular foundation models can do survival analysis. *arXiv preprint arXiv:2601.22259*, 2026.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Christopher Kolberg, Jules Kreuer, Jonas Huurdeman, Sofiane Ouaari, Katharina Eggensperger, and Nico Pfeifer. Tabpfn-wide: Continued pre-training for extreme feature counts. *arXiv preprint arXiv:2510.06162*, 2025.

Weihao Kong and Abhimanyu Das. Introducing TabFM: A zero-shot foundation model for tabular data. Google Research Blog, June 2026. URL https://research.google/blog/introducing-tabfm-a-zero-shot-foundation-model-for-tabular-data/. Accessed: 2026-09-14.

Subra Kugathasan, Lee A Denson, Thomas D Walters, Mi-Ok Kim, Urko M Marigorta, Melanie Schirmer, Kajari Mondal, Chunyan Liu, Anne Griffiths, Joshua D Noe, et al. Prediction of complicated disease course for children newly diagnosed with crohn's disease: a multicentre inception cohort study. *The Lancet*, 389(10080):1710–1718, 2017.

Alexander Lachmann, Denis Torre, Alexandra B Keenan, Kathleen M Jagodnik, Hoyjin J Lee, Lily Wang, Moshe C Silverstein, and Avi Ma'ayan. Massive mining of publicly available rna-seq data from human and mouse. *Nature communications*, 9(1):1366, 2018.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. The molecular signatures database hallmark gene set collection. *Cell systems*, 1(6): 417–425, 2015.

Yue Lyu, Steven H Lin, Xuelin Huang, and Ziyi Li. A censoring-aware target interface for tabular foundation models in survival prediction. *arXiv preprint arXiv:2607.09577*, 2026.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L Caterini, and Maks Volkovs. Tabdpt: Scaling tabular foundation models on real data. *Advances in Neural Information Processing Systems*, 38:172692–172722, 2025.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

Cancer Genome Atlas Research Network. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *New England Journal of Medicine*, 368(22):2059–2074, 2013.

Giovanni Palla, Alexander Hillsley, Yang-Joon Kim, and Loic A Royer. Tabular foundation models are competitive cellular perturbation predictors across biological scales. *bioRxiv*, pp. 2026–06, 2026.

Amit Pande, Bora Uyar, and Altuna Akalin. An atlas-scale generative model for unified representation learning of bulk rna-seq data. *bioRxiv*, pp. 2026–06, 2026.

James D Pearce, Sara E Simmonds, Gita Mahmoudabadi, Lakshmi Krishnan, Giovanni Palla, Ana-Maria Istrate, Alexander Tarashansky, Benjamin Nelson, Omar Valenzuela, Donghui Li, et al. Transcriptformer: A generative cell atlas across 1.5 billion years of evolution. *Science*, 393 (6806):aec8514, 2026.

Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

Shi-ang Qi, Weijie Sun, and Russell Greiner. Survivaleval: A comprehensive open-source python package for evaluating individual survival distributions. In *Proceedings of the AAAI Symposium Series*, volume 2, pp. 453–457, 2023.

Jingang Qu, David Holzmüller, GaÃl Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

Matthew G Rees, Brinton Seashore-Ludlow, Jaime H Cheah, Drew J Adams, Edmund V Price, Shubhroz Gill, Sarah Javaid, Matthew E Coletti, Victor L Jones, Nicole E Bodycombe, et al. Correlating chemical sensitivity and basal gene expression reveals mechanism of action. *Nature chemical biology*, 12(2):109–116, 2016.

Mostafa Rezapour, Stephanie V Trefry, Lorreta A Opoku, and Aarthi Narayanan. Artificial intelligence in bulk rna-seq: Challenges and potential solutions. *Computational and Structural Biotechnology Journal*, 35(1):0039, 2026.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorčan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embedding provides a foundation model for cell biology. *Nature*, pp. 1–9, 2026.

Marc-Andre Schulz and Kerstin Ritter. Measurement noise limits the advantage of nonlinear models over linear models in biomedical prediction. *arXiv preprint arXiv:2606.18420*, 2026.

Dmitrii Seletkov, Paul Hager, Georgios Kaissis, Rickmer Braren, Daniel Rueckert, and Raphael Rehms. Survival in-context: Amortized bayesian survival analysis via prior-fitted networks. *arXiv preprint arXiv:2603.29475*, 2026.

Wanxiang Shen, Intae Moon, Thinh H Nguyen, Michelle M Li, Yepeng Huang, Nitya Nair, Daniel Marbach, and Marinka Zitnik. Generalizable ai predicts immunotherapy outcomes across cancers and treatments. *Nature Medicine*, pp. 1–13, 2026.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011. doi: 10.18637/jss.v039.i05.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Terry M Therneau, Patricia M Grambsch, and Thomas R Fleming. Martingale-based residuals for survival models. *Biometrika*, 77(1):147–160, 1990.

Samuel J Tingle, Georgios Kourounis, Sofia Kazerouni, Harry VM Spiers, Miguel Larraz, Maithili Mehta, Serena MacMillan, Sarah A Hosgood, Michael L Nicholson, Neil S Sheerin, et al. Combining bulkformer and tabpfn to predict post-transplant function from kidney biopsies during normothermic machine perfusion or cold storage. *Scientific Reports*, 2026.

Zeynep Türkmen, Kürşat Kaya, Alexander Pfefferle, and Frank Hutter. Towards evaluating data priors for tabular foundation models. *arXiv preprint arXiv:2606.29241*, 2026.

Jeffrey W Tyner, Cristina E Tognon, Daniel Bottomly, Beth Wilmot, Stephen E Kurtz, Samantha L Savage, Nicola Long, Anna Reister Schultz, Elie Traer, Melissa Abel, et al. Functional genomic landscape of acute myeloid leukaemia. *Nature*, 562(7728):526–531, 2018.

Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. *Advances in neural information processing systems*, 29, 2016.

Koen Van den Berge, Katharina M Hembach, Charlotte Soneson, Simone Tiberi, Lieven Clement, Michael I Love, Rob Patro, and Mark D Robinson. Rna sequencing data: hitchhiker's guide to expression analysis. *Annual Review of Biomedical Data Science*, 2(1):139–173, 2019.

Dieudonne van der Meer, Syd Barthorpe, Wanjuan Yang, Howard Lightfoot, Caitlin Hall, James Gilbert, Hayley E Francies, and Mathew J Garnett. Cell model passports—a hub for clinical, genetic and functional datasets of preclinical cancer models. *Nucleic acids research*, 47(D1): D923–D929, 2019.

Mariana Vargas Vieyra. Staying alive: Uncensored survival analysis with tabular foundation models. *arXiv preprint arXiv:2606.03689*, 2026.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Yuanrui Wang, Xingxuan Zhang, Han Yu, Mingchao Hao, Gang Ren, Hao Yuan, Li Mao, Yunjia Zhang, Chun Yuan, and Peng Cui. Limix-2m: Mitigating low-rank collapse and attention bottlenecks in tabular foundation models. *arXiv preprint arXiv:2606.04485*, 2026.

David Wissel, Nikita Janakarajan, Aayush Grover, Enrico Toniato, Maria Rodríguez Martínez, and Valentina Boeva. Survboard: standardized benchmarking for multi-omics cancer survival models. *Briefings in Bioinformatics*, 26(5):bbaf521, 2025.

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature machine intelligence*, 4(10):852–866, 2022.

Liu Yang, Kangwook Lee, Robert Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *International conference on learning representations*, volume 2024, pp. 42195–42214, 2024.

Wanjuan Yang, Jorge Soares, Patricia Greninger, Elena J. Edelman, Howard Lightfoot, Simon Forbes, Nidhi Bindal, Dave Beare, James A. Smith, I. Richard Thompson, Sridhar Ramaswamy, P. Andrew Futreal, Daniel A. Haber, Michael R. Stratton, Cyril Benes, Ultan McDermott, and Mathew J. Garnett. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Research*, 41(D1):D955–D961, November 2012. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gks1111. URL http://academic.oup.com/nar/article/41/D1/D955/1059448/Genomics-of-Drug-Sensitivity-in-Cancer-GDSC-a.

Hatairat Yingtaweesittikul, Jiaxi Wu, Aanchal Mongia, Rafael Peres, Karrie Ko, Niranjan Nagarajan, and Chayaporn Suphavilai. Creammist: an integrative probabilistic database for cancer drug response prediction. *Nucleic acids research*, 51(D1):D1242–D1248, 2023.

Xiyuan Zhang, Danielle Maddix Robinson, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael Mahoney, et al. Mitra: Mixed synthetic priors for enhancing tabular foundation models. *Advances in neural information processing systems*, 38:15795–15840, 2025.

A EVALUATION DATASETS

Scope. Transcriptomic foundation models are commonly validated on tasks that probe the quality of integrated embeddings, such as cell-type annotation (Yang et al., 2022; Rosen et al., 2026) or tissue-of-origin prediction (Pearce et al., 2026; Pande et al., 2026). These tasks are informative about representation quality but have limited bearing on clinical decisions. Our benchmark instead consists of prognostic and diagnostic prediction tasks on bulk transcriptomic cohorts. It comprises 27 classification and 25 regression tasks (52 in total) and, separately, 28 survival tasks. Table 3 lists every task with its source, target, sample size, and task type.

Task categories. The classification and regression tasks cover (i) response to drugs, vaccines, or other therapies, including ex-vivo drug sensitivity of patient-derived cells and cell lines, (ii) prognosis, such as recurrence or relapse, and (iii) non-invasive diagnosis from blood transcriptomes, together with a few tumor and tissue phenotypes (e.g. Ki67 index or telomere length). Survival prediction forms a fourth group and is evaluated under a separate protocol described below.

Data sources. Classification and regression tasks are drawn from three kinds of sources. The first are large clinical cohorts: TARGET (Bolouri et al., 2018), TCGA (Network, 2013), BeatAML (Tyner et al., 2018), and the pediatric inflammatory bowel disease cohorts PROTECT (ulcerative colitis) and RISK (Crohn's disease) (Hyams et al., 2019; Kugathasan et al., 2017). The second are preclinical pharmacogenomic screens including cell-line drug-response panels from GDSC, CCLE, and CTRP (Yang et al., 2012; Barretina et al., 2012; Rees et al., 2016) obtained in harmonized form from Cell Model Passports (van der Meer et al., 2019) and CREAMMIST (Yingtaweesittikul et al., 2023). The third are smaller clinical cohorts deposited in the Gene Expression Omnibus (GEO), identified by mining GEO sample annotations for clinical outcome keywords (e.g., *clinical*, *therapy*, *response*, *diagnosis*, etc.). We only retained study designs in which expression was profiled at baseline, before the outcome was measured, and excluded cross-sectional studies. Most candidate studies were excluded because of their size ($n \leq 30$ patients, $n < 5$ samples per class), because the sample was taken at the same time as the outcome was recorded rather than at baseline before it, or because their target was unpredictable (AUROC ≤ 0.55 or Spearman $\rho \leq 0.1$ with TabPFN-3). We retained only longitudinal designs in which expression precedes the outcome, so that the prediction task is prospective.

Survival tasks. For survival prediction, we use SurvBoard (Wissel et al., 2025), a benchmark that standardizes preprocessing, data splits, and evaluation for cancer survival models. It comprises 28 cancer datasets from four programs: TCGA, ICGC, TARGET, and METABRIC. SurvBoard contains clinical and multi-omic data, but we exclusively use the gene-expression modality.

Table 3: **Datasets used in the benchmark.** n is the number of usable (non-NaN) rows (samples) for the associated task. For datasets with several tasks of the same type (i.e. BeatAML, CMP), the range over tasks is given. Type: cls = classification, reg = regression.

Dataset	n	Type	Description
GSE107422	80	cls	Systemic recurrence from colorectal cancer resection tissue
GSE116324	44	cls	Bortezomib-induction response (response sustained $\geq$1 yr) in multiple myeloma patients
GSE120622	80	cls	Non-Small cell lung cancer (surgical tumor) relapse
GSE154261	73	cls	24-month recurrence of early stage non-muscle-invasive bladder cancer
GSE156699	88	cls	platinum chemotherapy response (PFS $\geq$6 mo) of high-grade serous ovarian cancer patients
GSE157657	59	cls	Anti-tuberculosis therapy response (4-class) from whole blood

continued on next page

Table 3 – continued

Dataset	n	Type	Description
GSE176178	40	cls	Durable vs. non-durable Bacillus Calmette-Guerin (immunotherapy) response in early stage bladder cancer patients
GSE198520	46	cls	EULAR response to anti-TNF therapy (3-class) from rheumatoid arthritis synovium
GSE213346	40	cls	Recalcitrant vs. stable disease status at 1 year from oral lichen planus biopsy
GSE278476	46	cls	Immune responder status from peripheral blood mononuclear cells in a MUC1-vaccine prevention trial
GSE294705	60	cls	High-frequency recurrence index in non-muscle-invasive bladder cancer patients post Bacillus Calmette-Guerin therapy
GSE316750	41	cls	Biochemical recurrence from radical-prostatectomy tumor
GSE54460	99	cls	Biochemical recurrence from radical-prostatectomy tissue
GSE109142	198	cls	Week-4 clinical remission in pediatric ulcerative colitis
GSE112927	235	cls	Death-censored kidney allograft loss from pre-transplant blood
GSE163882	217	cls	pathological complete response to neoadjuvant chemotherapy from breast cancer pre-treatment biopsy
GSE185263	345	cls	In-hospital mortality from early-sepsis whole blood
GSE243375	242	cls	Pathological complete response to HER2+ breast neoadjuvant T-DM1+pertuzumab
GSE93624	245	cls	3-year complicated progression from pediatric Crohn's disease ileal biopsy
GSE175718	234	cls	Kidney-transplant rejection type (4 classes) from blood
GSE107995	414	cls	Future tuberculosis status (4 classes) from whole blood
GSE156902	796	cls	GBM/brain-metastasis/MS/control classification (4-class) from tumor-educated platelets (peripheral blood)
GSE234297	144	cls	ALS vs. healthy diagnosis from whole blood
GSE279480	255	cls	CMV serostatus (positive/negative) from whole blood
GSE68086	246	cls	Pan-cancer type vs. healthy classification (7-class) from tumor-educated platelets (peripheral blood)
TARGET	159	cls	INSS stage from pediatric neuroblastoma tumors
TARGET	2056	cls	relapse from pediatric tumors
GSE115525	73	reg	Prednisolone LC50 drug sensitivity of B-lineage ALL primary cells
GSE157103	66	reg	Hospital-free days (45-day window) from leukocytes of COVID-19 intensive care unit patients
BeatAML	277–492	reg	AML patient-derived ex-vivo drug sensitivity (AUC) for 10 drugs
GTEx	3940	reg	Telomere length from multi-tissue samples
GSE81538	405	reg	Ki67 proliferation index from breast cancer
CMP[1]	1177–1202	reg	IC50 drug sensitivity of pan-cancer cell lines for 10 drugs
TARGET	453	reg	Minimal residual disease level from pediatric tumors

[1] Bulk gene expression data from Cell Model Passports (CMP). Corresponding labels are from CREAMMIST, which integrates drug-response curves from GDSC, CCLE, CTRP, and PRISM.

B BENCHMARK DETAILS

Splitting and cross-validation. Regression splits are random, but classification splits are stratified by the target. In cohorts with replicates or several samples per patient, splits are additionally grouped by patient, so that no patient contributes to both the support and the query set. In Table 1, classification and regression tasks are evaluated with 5-fold cross-validation and results are averaged across folds, except R^2, which is computed from pooled out-of-fold predictions because fold-wise R^2 is unstable for small folds. For methods with distinct classification and regression checkpoints, we report the mean parameter count across the two in Table 1.

Survival tasks instead use the first five of the 25 fixed splits distributed with SurvBoard (Wissel et al., 2025). In every split, all data-dependent preprocessing, including standardization and the principal-component projection, is fitted on the support set only.

Model inputs. Expression profiles are $\log(1 + \mathrm{CPM})$ values over the $G = 20{,}021$ reference genes. GeneICL takes these profiles directly and does its own standardization and 90%-variance PCA. Every other tabular model, including the conventional baselines, gets the same two steps applied before the model. The only exception is TabPFN-Wide, which is designed for inputs with thousands of features and receives $\log(1 + \mathrm{CPM})$ values as input.

Transcriptomic foundation models. Transcriptomic foundation models are used as frozen feature extractors, without fine-tuning. Each model receives our profiles converted to its expected input format, with genes missing from its vocabulary set to zero, and returns one embedding per sample. These embeddings replace the expression profiles in the pipeline above (standardization and PCA). An elastic-net regularized linear probe is then fitted on the support set of each fold.

COMPASS (Shen et al., 2026), one of the transcriptomic foundation models we benchmark, expects a cancer-type token for every sample, but it has no token for non-cancer or unknown samples. Because many of our cohorts are not cancer cohorts, every sample receives a neutral token equal to the mean of the 33 learned cancer-type embeddings. We use the self-supervised pretrained checkpoint, not the variants fine-tuned on immunotherapy response, which would add external supervision.

C HOLDOUT TASKS FOR MODEL DEVELOPMENT

To avoid overfitting design choices to the benchmark, we ran preliminary experiments on a separate set of datasets, which also served as the validation set monitored during pretraining. The set contains five tasks from five GEO series that are not part of the benchmark and, unlike the benchmark, consists mostly of biological rather than clinically relevant endpoints, such as brain region in post-mortem brain (six classes, GSE102556), age at death from post-mortem brain (GSE80655), age at blood draw (GSE124284), major depressive disorder versus healthy controls (GSE251778), and histological remission in inflammatory bowel disease biopsies (GSE193677, current status, not a prospective label).

Although these tasks differ from those of our benchmark, conclusions drawn on the holdout tasks transferred to it. Changes that did not affect performance on the holdout tasks (Table 4) did not affect performance on the full benchmark (Table 2) and alternative pretraining distributions lowered regression performance on both sets.

The holdout tasks also show that GeneICL does not benefit from additional input components. Giving the pretrained model all principal components with non-zero variance at inference, rather than only those explaining 90% of the support-set variance, left classification mostly unchanged (AUROC 0.847 vs. 0.846) but lowered regression performance (Spearman 0.691 vs. 0.753, pooled R^2 0.506 vs. 0.629, see Table 4). This supports the 90% threshold used throughout.

Table 4: **Experiments evaluated on our holdout datasets (three classification and two regression)**. In the last experiment, GeneICL was given all principal components at inference instead of those explaining 90% of the support-set variance. *Attn.* denotes the asymptotic cost of the feature-axis attention. Parentheses report the standard deviation over three seeds in units of the last decimal place. Best and second-best results are bold and underlined, respectively.

Variant	Params	Attn.	Classification			Regression		
			AUROC	AUPRC	Bal. Acc.	Pearson	Spearman	R^2
GeneICL	4.24M	$\mathcal{O}(n)$	<u>0.846</u>(2)	**0.812**(1)	**0.787**(10)	**0.806**(7)	<u>0.753</u>(8)	**0.629**(4)
Untied (8 blocks)	19.25M	$\mathcal{O}(n)$	0.845(1)	**0.812**(3)	<u>0.784</u>(6)	<u>0.805</u>(5)	**0.758**(4)	<u>0.622</u>(8)
Full attention	4.05M	$\mathcal{O}(n^2)$	0.843(1)	0.808(2)	0.778(11)	0.784(13)	0.737(20)	0.581(28)
TabICLv2 prior pretraining	4.24M	$\mathcal{O}(n)$	0.836(4)	0.801(5)	0.769(18)	0.744(11)	0.688(5)	0.518(20)
Real expression data pretraining	4.24M	$\mathcal{O}(n)$	0.843(3)	0.800(14)	0.731(59)	0.788(9)	0.736(8)	0.457(35)
Without ranking loss	4.24M	$\mathcal{O}(n)$	0.729(5)	0.609(10)	0.554(11)	0.755(15)	0.716(13)	0.514(32)
GeneICL, all PCs at inference	4.24M	$\mathcal{O}(n)$	**0.847**(2)	<u>0.811</u>(5)	0.772(18)	0.739(12)	0.691(9)	0.506(28)

D HYPERPARAMETERS

Table 5: **Hyperparameter selection for GeneICL**. Bolded values indicate identical configurations to TabICLv2.

Parameter	Value
Architecture	
Embedding dim D	**128**
Attention heads	**8**
Encoder depth	**3**
CLS columns	**4**
Inducing rows (col. attn. bottleneck)	4
Thinking rows (scratchpad tokens)[1]	8
Recurrence depth (ICL loops)[2]	8
Norm	RMSNorm
Activation function	**GELU**
Dropout	**0**
Max classes	**10**
Training	
Optimizer steps	20,000
Warmup	**5,000 steps**
Tasks per step	6
Segments[3]	2×4 steps
Optimizer (FFN matrices)	**Muon**, lr 6×10^{-4}, momentum 0.95
Optimizer (rest)	AdamW ScheduleFree, lr 3×10^{-4}
Weight decay	0.01
Gradient clip	**1.0**
Seeds	0, 1, 2

[1] Learned reasoning workspace prepended to the ICL context sequence, adapted from TabPFN-2.5.
[2] Product of `segment_iters` × `halt_max_segments`; fixed recurrence unrolling used during inference.
[3] Hidden state representation H is detached across segment boundaries, with evidence features re-encoded using current parameters at each segment.

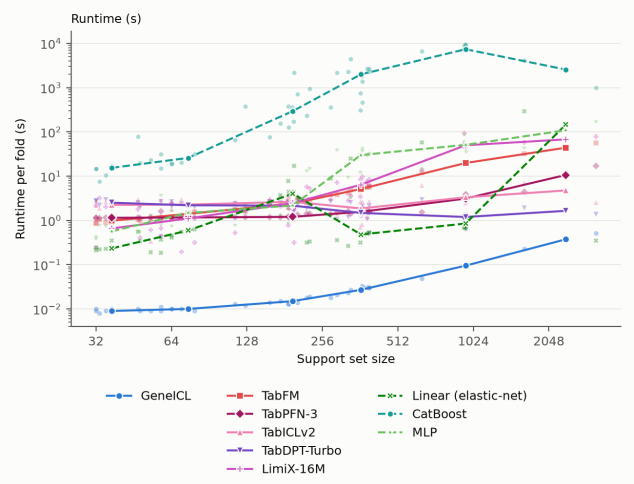

Figure 4: **Runtime vs. support set size.** Wall-clock runtime per cross-validation fold (log scale) against the number of support (training) samples (log scale), for GeneICL (blue), the other tabular foundation models (red to purple) and the tuned baselines (green, dashed). Each faint marker is one of the 52 benchmark tasks. Lines connect the medians over tasks within support-size bins (edges 32, 64, 128, 256, 512, 1024, 4096). Classification and regression tasks are pooled. Runtime covers fit and predict: for the in-context models that is the forward pass. All models ran on an NVIDIA GeForce RTX 4090 GPU and, except CatBoost and the linear model, which ran on AMD EPYC 7763 CPUs.

E COMPUTATIONAL RESOURCE REQUIREMENTS

We recorded the wall-clock runtime per cross-validation fold of all tabular foundation models and tuned baselines during the benchmark (Figure 4). For the tuned baselines, this includes the full hyperparameter search (50 candidates, each evaluated with inner 5-fold cross-validation); for the in-context models, it is a single forward pass. Across all support-set sizes, GeneICL is consistently the fastest model, requiring about 0.01–0.4 s per fold, one to two orders of magnitude less than the other tabular foundation models. Note that CatBoost and the elastic net ran on CPUs, whereas all other models ran on NVIDIA GeForce RTX 4090 GPUs.

In addition, we tested GeneICL without accelerators and ran the classification and regression benchmark (32 datasets) on a laptop CPU (Ryzen 7 7730U, 8 cores, 16 threads) with 16 GB of RAM. We used the same 5-fold cross-validation protocol as in the main experiments and applied no test-time ensembling. Each fold is one in-context forward pass on the raw gene-expression profiles. Averaged over folds and then over tasks, one fold took **7.5 seconds** of wall-clock time and the peak memory use was **4.9 GB**. For comparison, training and inference of an elastic-net baseline took 10.9 seconds per fold on the same laptop, where the average is dominated by the logistic-regression solver on large classification cohorts. GeneICL can therefore be applied to typical bulk transcriptomic cohorts on a standard laptop.

F PRETRAINING

F.1 VARIATIONAL AUTOENCODER

Training data. The VAE used for data generation during GeneICL pretraining is trained on human bulk RNA-seq profiles from ARCHS4 (Lachmann et al., 2018) (`human_gene_v2.latest`, upstream build of 19 May 2026). After excluding every GEO series with curated labels and every sample that appears in a benchmark task, we keep samples that are non-single-cell (ARCHS4 single-cell probability below 0.5), have a library size between 5×10^5 and 10^8 counts, leaving 547,829 profiles for VAE training. Counts are converted to $\log(1 + \mathrm{CPM})$ using the full-library sum and mapped onto a reference axis of $G = 20{,}021$ protein-coding genes (HGNC symbols intersected with GENCODE v44), summing duplicate source symbols and zero-filling reference genes without a match.

Architecture. We use a single-latent Gaussian VAE with one hidden layer of width 5,000 in both the encoder and the decoder, GELU activations, and a 256-dimensional latent space (see Table 6). The encoder outputs the mean and log-variance of a diagonal Gaussian posterior. The decoder output is linear and the likelihood is Gaussian with unit variance.

Training. Because GeneICL consumes whole datasets rather than individual profiles, the VAE is trained on locally coherent groups of samples. Each step draws $B = 8$ anchor profiles uniformly at random and forms a group from each anchor and its $n-1$ nearest neighbors, with $n \sim \log\mathcal{U}[64, 512]$ and neighbors found by exact Euclidean search in a 500-component PCA space fitted on a random subsample of 100,000 profiles and applied to the whole corpus. Within each group, every gene is standardized across samples. Every row $\mathbf{x}_{bi} \in \mathbb{R}^G$ is then encoded independently to a posterior $q_\psi(\mathbf{z} \mid \mathbf{x}_{bi}) = \mathcal{N}(\boldsymbol{\mu}_{bi}, \operatorname{diag} \boldsymbol{\sigma}^2_{bi})$ and decoded to $\hat{\mathbf{x}}_{bi} = D_\phi(\mathbf{z}_{bi})$ with $\mathbf{z}_{bi} \sim q_\psi$. In parallel, $m = \min(n, 256)$ rows per group are decoded from the prior, $\tilde{\mathbf{x}}_{bj} = D_\phi(\tilde{\mathbf{z}}_{bj})$ with $\tilde{\mathbf{z}}_{bj} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Writing $\mathbf{X}$ and $\tilde{\mathbf{X}}$ for the real and the prior-decoded batch, the loss at step t is

$$\mathcal{L}_{\mathrm{VAE}} = \mathcal{L}_{\mathrm{rec}} + \beta_t \, \mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{cov}} \, \mathcal{L}_{\mathrm{cov}} + \lambda_{\mathrm{mm}} \, \mathcal{L}_{\mathrm{mm}}, \tag{12}$$

where $\beta_t = \min(1,\, t/1{,}500)$ implements the KL warm-up and $\lambda_{\mathrm{cov}} = 20$, $\lambda_{\mathrm{mm}} = 10$. The first two terms are the usual ELBO under a unit-variance Gaussian likelihood,

$$\mathcal{L}_{\mathrm{rec}} = \frac{1}{Bn} \sum_{b,i} \tfrac{1}{2} \big\| \hat{\mathbf{x}}_{bi} - \mathbf{x}_{bi} \big\|_2^2, \qquad \mathcal{L}_{\mathrm{KL}} = \frac{1}{Bn} \sum_{b,i} D_{\mathrm{KL}}\big(q_\psi(\mathbf{z} \mid \mathbf{x}_{bi}) \,\big\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big). \tag{13}$$

The two auxiliary terms see only prior samples and constrain dataset-level structure that the ELBO leaves free,

$$\mathcal{L}_{\mathrm{cov}} = 1 - \operatorname{corr}\Big(\operatorname{offdiag} \hat{\boldsymbol{\Sigma}}_{\mathcal{S}}(\tilde{\mathbf{X}}),\ \operatorname{offdiag} \hat{\boldsymbol{\Sigma}}_{\mathcal{S}}(\mathbf{X})\Big), \qquad \mathcal{L}_{\mathrm{mm}} = \Big(\log \bar{v}(\tilde{\mathbf{X}}) - \log \bar{v}(\mathbf{X})\Big)^2. \tag{14}$$

Here $\mathcal{S}$ is a fresh random subset of 128 genes at every step, $\hat{\boldsymbol{\Sigma}}_{\mathcal{S}}$ is the gene–gene covariance over $\mathcal{S}$ obtained by centering each group on its own mean and pooling the groups, corr is the Pearson correlation over the upper-triangular entries, and $\bar{v}(\cdot)$ is the within-group variance of a gene, averaged over genes and groups. We train for 10,000 steps with AdamW on a single RTX 4090 (about 70 minutes) and use the final checkpoint. At the end of training, generated and real gene–gene covariances correlate at $r = 0.87$, measured on 32 evaluation neighborhoods over 256 randomly chosen genes, and generated samples retain 53% of the real within-group variance.

Sampling. During GeneICL pretraining, every row of a synthetic dataset is generated independently by drawing $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and taking the decoder mean $D_\phi(\mathbf{z})$, without observation noise or temperature scaling.

F.2 PRETRAINING PRIOR

Table 7 lists the sampling distributions of the semi-synthetic prior. We write $\log \mathcal{U}[a, b]$ for a log-uniform distribution and $\log \mathcal{U}\{a, \ldots, b\}$ for its integer counterpart $\lfloor \exp(\mathcal{U}[\log a, \log(b+1)]) \rfloor$, which places probability proportional to $\log \frac{k+1}{k}$ on each integer k.

Table 6: **VAE configuration and hyperparameters.**

Setting	Value
Input dimension	20,021 genes
Hidden width (encoder / decoder)	5,000 / 5,000
Latent dimension	256
Activation	GELU
Parameters	204M
KL weight	1, linear warm-up over 1,500 steps
Auxiliary loss weights (covariance / moment matching)	20 / 10
Groups per step $\times$ group size	$8 \times n$, $n \sim \mathrm{logU}[64, 512]$
Optimizer	AdamW, lr 2×10^{-4}, weight decay 10^{-4}
Gradient clipping	1.0
Training steps	10,000

Task and label space. Every training batch contains six synthetic datasets that share a sampled number of rows n and support fraction. Labels are constructed in a per-dataset principal-component representation of the generated matrix $\mathbf{X}$: we retain the leading components explaining 90% of the variance, whiten each score to unit variance, and clip it to $[-10, 10]$. Each score function then reads a subset $\mathcal{J}$ of k of these components, drawn without replacement with probability proportional to their explained-variance ratio, so that task signal concentrates on prominent axes of transcriptomic variation while occasionally involving minor ones.

Score functions. Each latent score s_{ik} is the average of an ensemble of one to three independently sampled MLPs, each standardized over samples before averaging. Every weight row is generated by applying a softmax to $-\alpha \log j + \beta\, \varepsilon_j$ ($j = 1, \ldots, d_{\text{in}}$, $\varepsilon_j \sim \mathcal{N}(0, 1)$), randomly permuting and sign-flipping the entries, and rescaling to norm $\sqrt{d_{\text{in}}}$. Small α and β yield dense, evenly spread weights, whereas large values concentrate the weight on a few inputs. Hidden pre-activations are standardized, scaled by a random pre-activation scaling, passed through a randomly chosen nonlinearity, and standardized again. In small-scale experiments, adding non-MLP score functions such as random decision-tree ensembles did not improve downstream performance. Richer function classes nonetheless remain a promising direction for future work.

Labels. Each latent score is mixed with independent Gaussian noise to reach a sampled signal fraction t (the fraction of target variance explained by the score). Binary labels threshold the first score at a random quantile, multiclass labels take the argmax over K scores, and the regression target is the first score. With probability 0.2, the regression target is additionally floored and/or capped at random quantiles, mimicking detection limits and saturated clinical measurements. Regression targets are standardized with support-set statistics.

Table 7: **Sampling distributions of the pretraining prior.** Quantities are resampled for every synthetic dataset unless marked as shared across the batch.

Quantity	Distribution
Task and label space	
Support fraction (shared)	$\mathcal{U}[0.5, 0.9]$
Samples per task	$\log \mathcal{U}[32, 1024]$
Number of selected PCs k	$\log \mathcal{U}\{2, \ldots, \min(64, k_{\text{tot}})\}$
Labels	
Number of classes K	$\log \mathcal{U}\{2, \ldots, 10\}$
Binary threshold quantile	$\mathcal{U}[0.2, 0.8]$
Random MLP functions	
Ensemble size	$\mathcal{U}\{1, 2, 3\}$
Number of layers	1 / 2 / 3 with probability 0.2 / 0.5 / 0.3
Hidden width	$\log \mathcal{U}\{4, \ldots, 64\}$
Weight concentration exponent α	$\log \mathcal{U}[0.1/\log(1 + d_{\text{in}}), 6]$
Weight noise β	$\log \mathcal{U}[10^{-4}, 10]$
Bias	$\log \mathcal{U}[0.01, 1] \cdot \mathcal{N}(0, 1)$
Pre-activation scaling	$\log \mathcal{U}[0.5, 3]$

G SURVIVAL PREDICTION

This section details GeneICL-Cox and the three groups of methods it is compared against: four alternative censoring adaptations applied to the same frozen GeneICL backbone, three tuned conventional survival models, and SurvPFN.

GeneICL-Cox. Let (t_i, δ_i) denote the observed time and event indicator of support sample i. We first transform the support labels into the martingale residuals r_i of Eq. 11, which requires no model, and then regress them with GeneICL in a single pass. Support scores are predicted out-of-fold over five random folds and query scores from the full support set: the out-of-fold scores are what the Breslow baseline hazard $\hat{H}_0$ is subsequently estimated at, and in-sample scores would be over-separated and bias $\hat{H}_0$ downwards, decalibrating the survival curves $S(t \mid \mathbf{x}) = \exp(-\hat{H}_0(t)\, e^{f(\mathbf{x})})$.

Alternative censoring adaptations. The following four methods use the same frozen GeneICL backbone as GeneICL-Cox and differ only in how right-censored outcomes are turned into targets. The first three are the adaptations evaluated by Vieyra (2026), who introduced the Buckley–James and jackknife variants and adopted the discrete-time variant from Kim et al. (2026). Two-step imputation is a simple imputation baseline.

- **Buckley–James estimator**: This method iteratively imputes censored log-times. In each of 10 iterations, GeneICL regresses log-time using all events and a random half of the censored samples as context. Each censored log-time is then re-imputed by sampling from a log-normal residual distribution truncated below at the censoring time, and the scale σ is re-estimated by censored maximum likelihood. Imputations and σ are initialized from the jackknife variant below.
- **Jackknife pseudo-observations**: Censored targets are replaced by leave-one-out Kaplan–Meier pseudo-observations of survival at the median event time, which are converted into survival-time pseudo-targets. GeneICL then regresses these targets in a single pass.
- **Discrete-time hazards**: The time axis is partitioned into 10 intervals at percentiles of the training event times. For each interval, GeneICL's classification head predicts the conditional event probability h_k among samples still at risk. The survival function is then $S(t) = \prod_{k:\, \tau_k \le t}(1 - h_k)$.
- **Two-step imputation**: GeneICL first predicts event times for the censored samples using only the uncensored samples as context. It then predicts query event times using all samples, with the imputed times, as context. The resulting survival function is a step function at the predicted event time.

Baseline methods.

- **Elastic net**: Cox PH elastic net was implemented using `glmnet` (Friedman et al., 2010; Simon et al., 2011) in `R`, using a fixed α mixing component of 0.1 (`alpha=0.1`), five-fold inner cross-validation stratified on the event indicator, and considering 100 regularization hyperparameters (`nlambda=100`), among which the hyperparameter yielding minimum cross-validation error was chosen (`lambda.min`). All other parameters were left at their respective defaults. Survival functions were extracted using `pec` at all unique test timepoints.
- **Random survival forests**: Random survival forests (Ishwaran et al., 2008) were implemented using `scikit-survival` (Pölsterl, 2020) in `Python`, using inner five-fold cross-validation stratified on the event indicator to choose the set of hyperparameters maximizing mean Harrell's concordance based on the ensemble risk score prediction for each set of hyperparameters. Mirroring Kim et al. (2026), we tuned `n_estimators` $\in \{50, 100, 200\}$, and `min_samples_split` $\in \{2, 5, 10, 20\}$. All other parameters were left at their respective defaults. Survival functions were extracted using the `predict_survival_function` at all unique test timepoints, clipped to the training time domain.
- **XGBoost**: A Cox PH gradient boosting model was implemented using `xgboost` (Chen & Guestrin, 2016) in `Python` (`survival:cox`). Hyperparameters were selected via inner five-fold cross-validation stratified on the event indicator, using `GridSearchCV`. Following Kim et al. (2026), we tuned `n_estimators` $\in \{100, 200\}$, `learning_rate` $\in \{0.1, 0.3\}$, and `max_depth` $\in \{3, 6\}$. For model selection, the cross-validation score was the mean negative Cox partial log-likelihood returned directly by XGBoost (`cox-nloglik`), obtained by calling the fitted booster evaluation API (`Booster.eval_set`) on each validation fold. All other parameters were left at their respective defaults. Survival functions were obtained by fitting a Breslow baseline hazard from `scikit-survival` (Pölsterl, 2020) to the predicted linear predictors and then evaluating the resulting survival step functions at all unique test timepoints, with evaluation times clipped to the training-time domain.

SurvPFN. In addition to tuned baselines, we compare GeneICL to SurvPFN (Böhm et al., 2026), a tabular foundation model pretrained on synthetic right-censored survival data that predicts a discretized distribution over log survival times. We used the released checkpoint with the full training split as context and the same input as GeneICL, namely the principal components explaining 90% of the variance, fitted on the training split. Risk scores were the negative predicted mean log survival time, and survival functions were evaluated at all unique test timepoints.

Metrics. Discriminative performance is measured using Antolini's time-dependent concordance index (Antolini et al., 2005), an overall score incorporating both discrimination and calibration via the integrated Brier score (IBS) using 100 points (Graf et al., 1999), and distributional calibration through the D-calibration test statistic using 10 bins (Haider et al., 2020). All metrics are calculated using `SurvivalEVAL` (Qi et al., 2023) on the first five SurvBoard test splits and averaged per dataset.

Predicted survival curves were post-processed to account for numerical issues in selected models (primarily enforcing monotonic non-increasing survival over time via carrying forward the survival probability at the previous timepoint where applicable).

D-calibration is undefined for the `two-step` variant and is thus not reported. XGBoost sometimes resulted in survival functions on which D-calibration calculation did not succeed, in which case these splits were reported as NA and subsequently imputed using the mean D-calibration across splits for the same model and the same dataset.

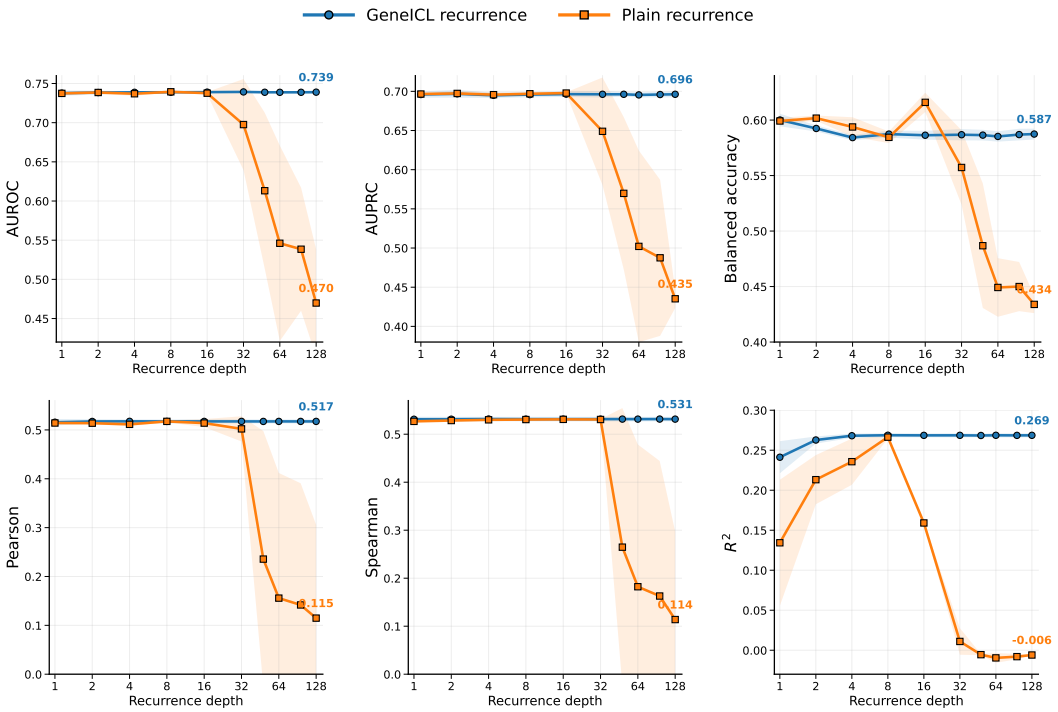

Figure 5: **Depth extrapolation of the recurrent in-context learner.** GeneICL and a plain recurrent baseline are trained with $T = 8$ recurrence steps and evaluated with up to 128 steps at inference time. The plain baseline repeatedly applies a post-norm Transformer block, whereas GeneICL uses the fixed-evidence, evolving-hidden-state recurrence described in Section 3.2. Panels report **(a)** AUROC, **(b)** AUPRC, **(c)** balanced accuracy, **(d)** Pearson correlation, **(e)** Spearman correlation, and **(f)** R^2. Lines and markers show the mean over three independently trained seeds and shaded regions indicate ± 1 standard deviation. GeneICL remains stable far beyond the training depth, whereas plain recurrence degrades under continued iteration.

H Extended ablations

H.1 Positional encodings and feature grouping

TabICL and TabICLv2 use rotary positional encodings and circular feature grouping to break the symmetry between features with similar distributions, without which samples that differ only by a permutation of their feature values can collapse to the same representation (Qu et al., 2026). GeneICL omits both because principal components have no order or sign that is shared across datasets, continuous PC scores rarely coincide under a permutation, and the support labels added in the encoder already distinguish features by their relation to the target.

H.2 Looping

GeneICL is trained with $T = 8$ recurrence steps. To test whether its recurrent computation remains stable beyond the depth encountered during training, we evaluate pretrained checkpoints at $T \in \{1, 2, 4, 8, 16, 32, 48, 64, 96, 128\}$. We compare the GeneICL recurrence described in Section 3.2, which separates fixed task evidence from an evolving hidden state, against a plain recurrent baseline that repeatedly applies a post-norm Transformer block without these recurrent design choices. Both models are trained with the same recurrence depth of $T = 8$. Results are averaged over three independently trained seeds.

Performance generally reaches its maximum, or a near-maximum, within the first few recurrence steps (Figure 5). Beyond this point, additional iterations do not systematically improve GeneICL.

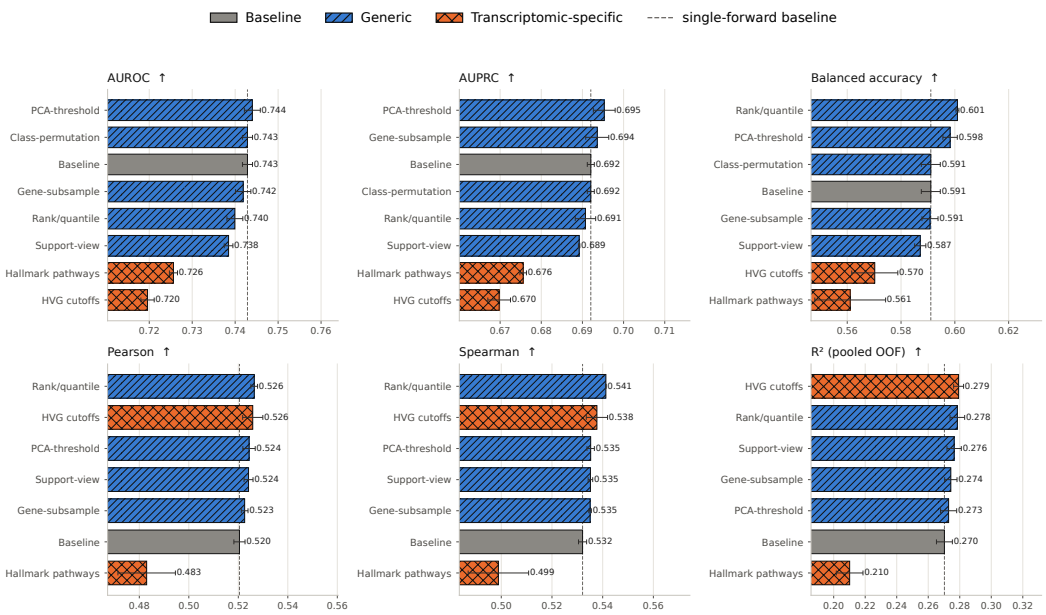

Figure 6: **Inference-time ensembling strategies for GeneICL.** Each ensembling method is applied independently to a frozen GeneICL checkpoint and compared against a single-forward-pass baseline. We report AUROC, AUPRC, and balanced accuracy for classification, and Pearson correlation, Spearman correlation, and R^2 for regression. Bars show the mean performance across three independently trained seeds, with error bars indicating one standard deviation. The dashed vertical line marks the single-forward baseline. Generic ensembling strategies are compared with transcriptomic-specific strategies based on highly variable genes and Hallmark pathways. Higher is better for all metrics.

Crucially, however, its predictions remain stable as recurrence depth is increased far beyond the training horizon. Across all six classification and regression metrics, performance at $T = 128$ remains essentially unchanged from that around the training depth. Plain recurrence behaves differently. While it is competitive at small depths, performance deteriorates once the block is repeated substantially beyond the depth seen during training, with particularly large declines in Pearson correlation, Spearman correlation, and R^2.

Thus, weight sharing by itself is not sufficient for stable recurrent depth extrapolation. GeneICL's recurrent formulation supports substantially deeper iteration without destabilizing predictive performance, although additional recurrence does not provide a performance benefit on the present benchmark. One possible reason is that current biomedical prediction tasks may be limited more by the information contained in the measurements than by available model computation. Recent work argues that measurement noise can preferentially attenuate nonlinear predictive structure, thereby reducing the advantage of more flexible predictors (Schulz & Ritter, 2026). Whether additional recurrent computation becomes beneficial as transcriptomic measurements become more precise is therefore an interesting direction for future work.

H.3 Inference-time ensemble

We investigate whether GeneICL can benefit from inference-time ensembling without retraining or updating model parameters. Each strategy constructs multiple views of the same prediction problem using a single frozen GeneICL checkpoint and averages the resulting class probabilities for classification or point predictions for regression. We evaluate all strategies using three independently trained checkpoints and compare them against a single-forward baseline using the default PCA threshold of 0.90. The goal is to test whether robustness can be improved by varying the in-

put representation or support set, and whether transcriptomics-specific structure provides additional benefits over generic ensembling strategies.

We consider five generic ensembling strategies. *PCA-threshold* ensembling evaluates the model using PCA variance-retention thresholds of $\{0.85, 0.90, 0.95, 0.98\}$ and averages the resulting predictions, thereby combining multiple low-dimensional representations of the same expression profiles. *Support-view* ensembling averages predictions over eight class-stratified subsets containing 90% of the available support examples. *Gene-subsample* ensembling similarly averages over eight random subsets containing 50% of the input genes, providing a feature-bagging analogue for transcriptomic data. *Rank/quantile* (Bolstad et al., 2003) ensembling combines three different distributional representations of the expression matrix: the original values, a per-sample rank transformation, and quantile normalization to the mean sorted expression profile. Finally, *class-permutation* ensembling, which applies only to classification, averages predictions over eight permutations of the mapping between class labels and class indices after mapping the outputs back to the original label space.

We additionally evaluate two transcriptomics-specific alternatives. *HVG-cutoff* ensembling restricts the input to the top 500, $1{,}000$, $2{,}000$, or $5{,}000$ highly variable genes (Brennecke et al., 2013), with variability estimated from the support set, and averages predictions across the four resulting views. *Hallmark-pathway* ensembling instead evaluates the model separately on each of the 50 MSigDB Hallmark gene sets (Liberzon et al., 2015) and averages the corresponding predictions. These two approaches test whether ensembles based on biologically motivated feature subsets provide an advantage over generic perturbations of the input representation.

As shown in Figure 6, the generic ensembling strategies are generally competitive with the single-forward baseline and provide modest improvements for several metrics. PCA-threshold, support-view, gene-subsample, and rank/quantile ensembling all preserve or improve performance on at least part of the benchmark, with the strongest strategy depending on the metric. In contrast, the transcriptomics-specific approaches do not provide consistent gains. HVG-based ensembling substantially reduces classification performance, while Hallmark-pathway ensembling performs worse across all reported metrics. These results indicate that GeneICL benefits more from generic variation in the representation or support set than from restricting inference to predefined transcriptomic feature subsets.