Generative ranking enables scalable pretraining on noisy biological multisets

Glen Taggart[1], Adam Green[1,*]

[1]Markov Biosciences

*Corresponding author: adam@markov.bio

Abstract

No virtual cell model has demonstrated that scaling self-supervised pretraining consistently improves downstream biological prediction. We show that the obstacle is the training objective: all existing gene-space objectives are ultimately count-based, and count-based objectives spend capacity fitting technical artifacts of the measurement process rather than biology. We introduce ranking-based pretraining via Geometric Plackett–Luce (GPL; Henderson 2022), a generative likelihood that models gene expression as a ranking rather than a vector of counts. Under controlled ablation (same Transformer architecture, data, and compute), GPL outperforms all count-based alternatives at every scale tested. When finetuned, the pretrained model achieves state-of-the-art perturbation prediction across the full transcriptome, with downstream performance improving monotonically with pretraining scale. Zero-shot, the model recovers transcription factor–DNA binding—including repressor geometry, protein complex composition, and sequence motif enrichment. Linear probing reveals the model encodes the spatial topology of the cell from nucleus to secreted proteins, extending to transmembrane domain architecture, the organization of the druggable genome by therapeutic modality, and the endocytic biology that determines antibody-drug conjugate efficacy—correlating with clinical outcomes for approved therapeutics. These results point toward a scalable paradigm for training cellular world models from the noisy multisets that dominate modern experimental biology.

1 Introduction

The promise of so-called 'virtual cell' models—general-purpose simulators of cellular behavior learned from molecular data—has animated much of computational biology in recent years (Bunne et al., 2024). The data to train such models is now abundant: hundreds of millions of gene expression profiles are publicly available, with billions expected in the next year. Yet for all the excitement, scalable self-supervised pretraining—the approach most likely to exploit this abundance—has eluded the field: no published single-cell foundation model has demonstrated that self-supervised pretraining scalably improves downstream biological task performance. The field has conflated this failure with a limitation of observational data itself, investing heavily in the generation of large-scale perturbation datasets and task-specific architectures, on the assumption that learning cellular dynamics requires paired interventional examples. Yet if scalable self-supervised learning has succeeded across domains as diverse as language, vision, and protein sequences—in each case, when a simple, scalable training objective met a scalable architecture—the failure of analogous approaches on single-cell data is most parsimoniously explained not by poverty of the data but by a mismatch between the training objective and the data-generating process.

The challenges begin with the canonical unit of biological organization, the cell, and the nature of signal sampled from it. A cell is, quite literally, a bag of molecules. When we measure cellular state, we do so by

drawing samples from this bag. In the case of single-cell gene expression profiling, this involves capturing and sequencing transcripts of the 20,000 or so species of protein-coding mRNA floating around in the cell at varying concentrations. This yields a vector of integer counts per gene—five of gene$_A$, zero of gene$_B$, and so on; the resulting mathematical object, a multiset, is identical to the "bag of words" that dominated early natural language processing (Harris, 1954; Blei et al., 2003).

Unfortunately, it is far easier to do a census of words in a document than of molecules in a cell, and therefore the task of counting molecules is far more fraught. Observed counts arise from Poisson sampling of the cell's true gene counts (Sarkar and Stephens, 2021), and due to the technical limitations of transcript capture and sequencing depth, this sampling is shallow and only retrieves a small subset of the $O(10^5)$ mRNA molecules in a typical mammalian cell. This measurement noise, and in particular sequencing depth, drives not only the observed sparsity of these data—upwards of 95% of genes may register an observed count of zero in any given cell (Choi et al., 2020)—but the entire observed integer count distribution, rendering true individual count magnitudes ambiguous (an effect that, to our knowledge, we are the first to systematically characterize; Supplement B). Our biological multiset is therefore a noisy, sparse one.

And unlike the token sequences of language, these multisets admit of no natural self-supervised training objective. In the absence of one, the field has defaulted to count-based reconstruction objectives—mean squared error, negative binomial count likelihoods, cross entropy over gene proportions—that are fundamentally mismatched to the data. The most immediate problem is the aforementioned sparsity: any count-based objective must spend much of its capacity learning to predict zeroes that are fundamentally ambiguous in nature. And among detected genes, things are not much better, with the majority of counts crowding in the 1–3 range, a regime where measurement noise rivals the signal itself. Variance-stabilizing transforms do not resolve this: Warton (2018) proved that no monotonic transformation can stabilize variance when mean counts fall below one, as they do for the majority of genes in typical single-cell data. More principled approaches such as analytic Pearson residuals (Hafemeister and Satija, 2019) model the mean-variance relationship explicitly but remain transformations of count data subject to the same fundamental limitations.

Across datasets, the problem compounds. Because library chemistry and sequencing depth determine the observed count distribution, the same biological state produces non-comparable count vectors in different experiments. A model trained on counts must therefore learn the noise profile of every dataset it encounters—and as training data scales to encompass thousands of experiments, dozens of library chemistries, and a decade of evolving protocols, this technical heterogeneity threatens to obscure the very signal that scaling was meant to surface. Count-level objectives do not abstract away from these artifacts; they absorb them.

The answer to both problems lies in a long-established distinction in measurement theory: ordinal structure—the relative ordering among measurements—is approximately preserved under noise that obscures cardinal magnitudes, a consequence first formalized by Thurstone (1927) and systematized by Stevens (1946). Gene expression counts are nominally cardinal, but Poisson sampling noise degrades them to the point where ordinal structure—the ranking of genes by observed count—is the most informative level of measurement that reliably survives the measurement process. This ranking is naturally more comparable across datasets with different sequencing depths and library chemistries than the count magnitudes themselves. Empirical evidence confirms this. Binarized representations—the coarsest form of ordinal structure, reducing each gene to expressed or not expressed—perform comparably to full count data for clustering and cell type identification; binary dropout patterns are more consistent across sequencing technologies than count-based expression profiles (Qiu, 2020). Cross-dataset integration *even* improves under binarization (Bouland et al., 2023). The standard interpretation of this evidence is that binary representations are surprisingly informative. We take the converse reading: count magnitudes are surprisingly uninformative. But binarization overcorrects, discarding the relative ordering among expressed genes. Rankings preserve this ordering, modeling the signal further upstream of measurement noise rather than fitting the noise itself.

Operationalizing this insight as a training objective requires a generative likelihood over rankings, one that

handles ties (thousands of genes share identical counts, particularly at zero) and vocabularies of tens of thousands of genes. The Plackett–Luce model (Luce, 1959; Plackett, 1975) provides such a ranking likelihood but assigns zero probability to ties. The Geometric Plackett–Luce model (GPL; Henderson 2022)—which has found most recent application in accommodating ties in chocolate pudding preference ratings (Davidson, 1970)—extends Plackett–Luce to full rankings with ties via discrete geometric random variables, making tie equivariance a property of the distribution itself. The resulting likelihood factorizes to $O(V+S\log S)$ in vocabulary size V and expressed genes S, making ranking-based pretraining on full gene vocabularies computationally tractable at billion-parameter scale.

We demonstrate that this ranking objective unlocks scalable self-supervised pretraining on single-cell data. Under controlled ablation—identical architecture, data, and compute—GPL outperforms count-based and proportion-based alternatives at every scale. Finetuned on perturbation data, GPL-pretrained models achieve state-of-the-art perturbation prediction across the full transcriptome, improving monotonically with pretraining scale. Without any finetuning, gene embeddings recover transcription factor–DNA binding validated against ChIP-seq—distinguishing activators from repressors, resolving protein complex composition, and discriminating sequence motifs recognized by distinct regulatory complexes sharing a subunit. With only a linear probe, they encode subcellular protein topology along interpretable axes validated against the Human Protein Atlas—extending to transmembrane domain architecture and the spatial organization of the druggable genome by therapeutic modality, with cosine neighborhoods recovering signaling mechanisms, endocytic routes, and trafficking biology that correlate with clinical outcomes for approved antibody-drug conjugates. All from mRNA expression rankings alone.

2 Results

Observed counts arise from Poisson sampling of true gene proportions. Under the maximum-entropy (Exponential) prior on proportions, the marginal count distribution across genes within a single cell is Geometric: $n(k;D) = V(1-p)\cdot p^k$ where $p = D/(\alpha + D)$. One parameter (sequencing depth D) controls the entire distribution—not just the zero rate, but every count frequency. This is (to our knowledge) the first analytical characterization of integer count distributions as a function of sequencing depth.

GPL decomposes the cell's expression vector into two terms: a binary term (which genes are detected?) and a ranking term (in what order among expressed genes?). In a typical cell at 10,000 UMI, the majority of genes register zero counts—these collapse into a single tied group under the binary term. The ranking term operates only over the expressed genes, modeling their relative ordering. Count-based objectives have no such decomposition—they process every gene's count individually, spending equal effort on zeros and expressed genes.

Consider a toy cell with six genes and observed counts [0, 0, 0, 3, 3, 7]. GPL asks two questions: which three genes are expressed, and in what order? The three zeros collapse into a single tied group—one evaluation, not three. A count-based objective asks a different question: for each of the six genes individually, how probable is its exact count? It spends half its evaluations learning to predict zero. Scale this to a real cell—several thousand expressed genes among a full transcriptome of ~20,000 or more—and the asymmetry becomes the dominant feature of training.

This decomposition predicts two specific failure modes for count-based objectives at scale: (1) **Zero coupling**—objectives that normalize across all genes (MSE, proportional likelihood) waste capacity suppressing the massive zero group every training step, creating a ceiling independent of scale. (2) **Noise memorization**—objectives that fit count magnitudes (Geometric, Negative Binomial) memorize depth-dependent artifacts that compound across heterogeneous datasets, causing collapse at scale. The following ablation tests both predictions.

2.1 A ranking objective outscales all count-based alternatives

All models use a standard Transformer decoder architecture with muP hyperparameter transfer (Yang et al., 2022), pretrained on human single-cell RNA-seq data from scBaseCount (Youngblut et al., 2025). The training

objective varies across experiments; architecture, data, and compute are held constant. Full architectural details and training procedures are provided in Methods (§4).

2.1.1 GPL: a ranking-based pretraining objective

The GPL training objective models each cell's gene expression as a ranking rather than a vector of counts. For each cell, genes are ordered by observed expression, with undetected genes tied at the lowest rank. The model is trained to maximize the likelihood of this ranking under a Geometric Plackett–Luce distribution (Henderson, 2022), which handles ties natively. The resulting objective never sees a count—only the relative ordering among genes. Full details of the GPL formulation are provided in Methods (§4.2) and Supplement C.

2.1.2 Ablation design

To isolate the effect of the training objective, we trained four models at each of three scales (250M, 1B, and 2B parameters) under identical conditions—same transformer architecture, same training data, same compute budget, same hyperparameter scaling via width-only muP—varying only the loss function. The four objectives were chosen to decompose the ranking-versus-count distinction into two independent variables.

The first variable is **zero coupling**. Proportional (PROP) places a softmax over all ~36,000 genes, coupling expressed and unexpressed genes through a shared normalizer—every zero-count gene (varying from 50–95% of genes in any given cell) pulls probability mass from every expressed gene at every training step. Geometric Count (GC) avoids this by modeling each gene independently via the Geometric distribution: $P(\text{count} = k \mid \theta) = \theta(1-\theta)^k$. Each gene's count is scored against its own parameter θ with no shared denominator, so zeros do not compete with expressed genes for capacity.

The second variable is **ranking versus counts**. GPL and GC share the same Geometric parameterization—same output head, same per-gene θ, same UMI-depth shift—and differ only in what the loss function does with θ. GC scores each gene's observed count against the Geometric PMF directly; GPL scores the probability of the observed expression ranking under a race model where each gene draws a Geometric waiting time. GPL never sees a count. It sees only that gene A is more expressed than gene B, and that unexpressed genes share the lowest rank. This is the cleanest possible ablation: same distribution, same architecture, same data, one variable changed.

MSE on log-normalized counts (the most common baseline in the field) completes the comparison as a standard reference. We additionally tested Negative Binomial (NB), which generalizes GC by adding a learned per-gene dispersion parameter—the standard distributional assumption in scRNA-seq analysis (Lopez et al., 2018). NB represents the most flexible count-based objective in our comparison; it collapsed catastrophically mid-training as dispersion parameters absorbed technical variance (Supplement A).

We evaluate via masked prediction: the model sees a fraction of each cell's genes as context (50% in all reported results) and predicts expression values for the full vocabulary (~36,000 genes). Each model produces a per-gene scalar—a log-rate for GPL and GC, a logit for PROP, a predicted value for MSE—which is ranked by magnitude to produce a predicted gene ordering. Ground-truth rankings are derived from observed counts, with tied counts (including the undetected genes (typically 50–80% of the vocabulary)) assigned fractional ranks via the standard midrank convention. Spearman rank correlation between predicted and ground-truth orderings is computed using the same pipeline regardless of training objective; no objective-specific transformation is applied, and no objective is privileged by the use of a rank-based evaluation metric (Methods; Supplement D). All results are computed on cells from tissue-disease groups held out entirely from pretraining—the model has never seen a healthy lung cell or any cell from these evaluation groups during training.

We report two complementary metrics. Full-transcriptome Spearman correlation measures how well the model recovers the complete gene expression ranking including zero-expressed genes. Binary correctness measures whether the model correctly predicts which genes are expressed at all—a set-overlap measure asking what fraction of truly expressed genes appear in the model's top predictions. This is the evaluation most favorable to GC, which explicitly models $P(\text{count} = 0)$ via a per-gene Bernoulli component; GPL never sees zeros directly.

2.1.3 The fan-out

At 250M parameters, the four objectives perform comparably on held-out Spearman correlation, with GPL leading GC by +0.005 (**Fig. 1A**). At 1B parameters, the gap widens to +0.012. At 2B parameters, it widens further to +0.039—GPL reaches 0.6681 while GC declines to 0.6291 (**Fig. 1A**). The fan-out is monotonic: +0.005 → +0.012 → +0.039 (gaps measured against GC's peak performance before cooldown collapse; plotted values are post-collapse finals).

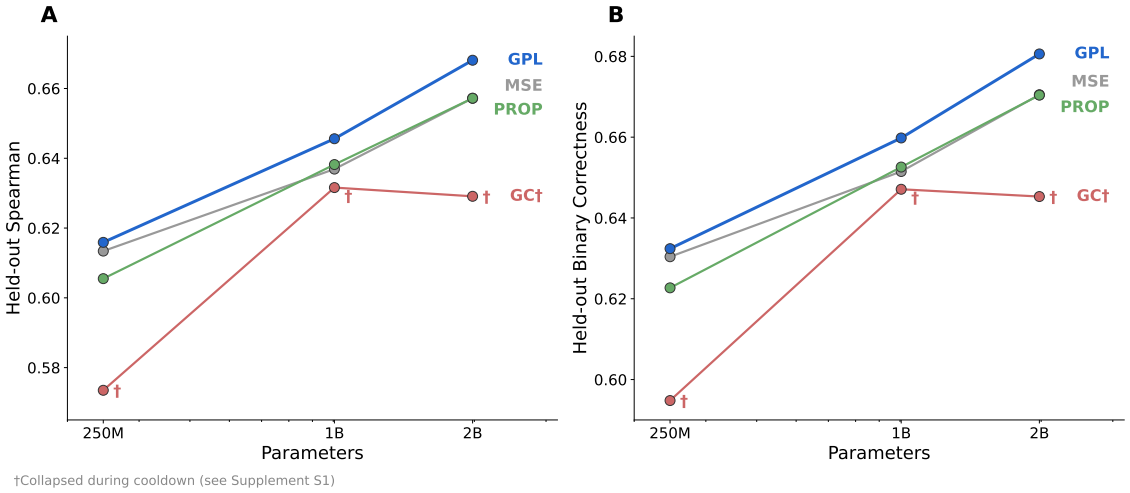

Figure 1. Pretraining objective scaling comparison. **(A)** Full-transcriptome Spearman correlation on held-out cells as a function of model scale for four training objectives (GPL, GC, PROP, MSE). GPL improves monotonically; GC declines at 2B parameters after training collapse. GC values shown are post-collapse finals (indicated by daggers); the GPL–GC gaps reported in text (+0.005, +0.012, +0.039) are measured against GC's peak pre-collapse performance to isolate the scaling effect from the collapse artifact. **(B)** Binary correctness (fraction of truly expressed genes predicted as expressed). GPL surpasses GC at 250M+ parameters despite never being trained on binary detection.

The 2B results confirm both failure modes independently. MSE and PROP—two different count-based objectives that both couple zeros through their loss—converge to identical performance (both 0.6572), as if the zero-coupling bottleneck imposes a shared ceiling. GC avoids zero coupling entirely (independent per-gene likelihood) but falls *below* both (0.6291, after training collapse; see below), because at 2B parameters the noise memorization failure mode is more damaging than zero coupling. GPL avoids both failure modes: no zero coupling (unexpressed genes form a single tied bucket) and no count fitting (the loss sees only the ranking). It reaches 0.6681, +0.011 above the zero-coupling ceiling and +0.039 above the collapsed count model.

The pre-cooldown trajectory confirms this is not an artifact of learning rate dynamics: at 50% of training, before cooldown begins at any scale, the GPL–GC gap already widens monotonically: +0.000 at 250M → +0.002 at 1B → +0.009 at 2B. Cooldown amplifies the advantage (from +0.002 to +0.012 at 1B, a 6× amplification) but does not create it.

2.1.4 Distributional flexibility memorizes noise at scale

GC's held-out Spearman declined from peak to final value at every scale tested: from 0.6107 (peak) → 0.5735 (final) at 250M, from 0.6334 → 0.6316 at 1B, and from 0.6539 → 0.6291 at 2B (**Fig. 1A**). At 2B, the decline begins mid-training—well before learning rate cooldown—and cooldown accelerates the collapse rather than causing it (Supplement A). At 1B, the decline also begins before cooldown, but cooldown partially rescues the model. GPL improved stably through training and cooldown at all scales. The pattern is consistent with the capacity argument: count-based models with more parameters have more capacity to memorize the training distribution's technical noise, and this memorization begins as soon as the model has sufficient capacity, independent of learning rate schedule. Count-based objectives converge toward noise-specific optima. GPL is more robust: its training signal—

the ranking—is more invariant to sequencing depth variation than count magnitudes, so larger models have less room to memorize technical noise.

Notably, adding distributional flexibility makes the problem worse, not better. Negative Binomial (NB)—the standard distributional assumption for scRNA-seq count data and the most flexible objective in our comparison—exhibited catastrophic overfitting: unregularized NB saw Spearman correlation collapse from 0.45 to 0.047 mid-training as the learned per-gene dispersion parameters absorbed technical variance rather than biological signal. Adding L2 regularization to the dispersion parameters stabilized training but did not close the gap—regularized NB remained below GC at convergence. This collapse occurs despite metadata conditioning via AdaLN, which explicitly provides the model with batch-level information—the dispersion parameters absorb technical variance that metadata conditioning was designed to handle. The failure is intrinsic to count-based pretraining, not an artifact of distributional misspecification: more parameters for the count distribution means more capacity to overfit sequencing noise (Supplement A).

2.1.5 Rankings subsume counts: a ranking model predicts binary detection better than models trained on it

Binary correctness is not a ranking metric—it is the evaluation most favorable to GC. Yet at 250M+ parameters, GPL predicts which genes are expressed more accurately than GC (binary correctness: 0.6598 vs 0.6486 at 1B; 0.6806 vs 0.6453 at 2B). A model trained only on rankings recovers the binary signal as an emergent property—and does so better than the model explicitly supervised on it.

At 250M, switching from MSE to GPL yields +0.005 in held-out Spearman. At 1B, the same switch yields +0.009—equivalent to approximately 40% of the gain from the ∼4× parameter increase from 250M to 1B under MSE training. The right objective is worth a fraction of a scale step, and the fraction grows with scale.

Having established that the ranking objective scales where count-based alternatives plateau, we next test whether these pretrained representations transfer to downstream biological tasks—beginning with the most direct test: predicting the effects of genetic perturbations the model has never seen.

2.2 Observational pretraining drives perturbation prediction without perturbation data

2.2.1 The field has not converged on how to evaluate perturbation prediction

The promise of virtual cell models rests on a concrete capability: given a cell and a perturbation it has never encountered, predict the resulting change in gene expression. Yet despite significant investment, the field has not converged on how to measure this capability, what constitutes a fair test of it, or whether existing models have achieved it.

Perturbation screens—the ground truth for evaluation—apply hundreds of genetic perturbations to populations of cells and read out gene expression via single-cell sequencing, a destructive assay. The perturbed and control cells are therefore not pre- and post-perturbation snapshots of the same cell; they are different cells from the same condition, pseudobulked by averaging across single cells within each treatment group.

The field has witnessed an efflorescence of evaluation metrics—MAE, PDS, Pearson, fold change correlation, DE overlap—that yield contradictory rankings across models and bear unclear relationships to experimental utility. We focus on mean absolute error (MAE)—the most interpretable metric and the only one with a straightforward relationship to the quantity it measures. For each perturbation, MAE is the average absolute difference between predicted and observed pseudobulked expression across all genes. The natural baseline is the control mean—the average expression of unperturbed cells. A model that scores below the mean baseline on MAE has, by definition, learned perturbation-specific effects; a model that does not has learned nothing the control average does not already provide. The Virtual Cell Challenge confirmed the difficulty of this metric: over 1,200 teams competed, and nearly all performed worse than mean prediction on MAE (Arc Institute, 2025).

MAE can be calculated against the full transcriptome or a pre-selected subset; most published evaluations report only the top 2,000 highly variable genes, pre-selected for maximal variance. The full transcriptome is a substantially harder test. That said, both trained model and mean baseline MAEs tend to be higher for the highly variable gene subset because these genes are, as the name suggests, highly variable.

2.2.2 "Perturbation prediction" conflates two distinct tasks

Most consequentially, the field elides two distinct generalization tasks under the term "perturbation prediction." Terms like "zero-shot" and "in-context learning" have been misappropriated from their standard machine learning definitions in this context; we provide a more descriptive taxonomy.

Cross-cell-type transfer. A perturbation is observed during training—in other cell types or as prompt cells at inference—and the model transfers its effect to a new cellular context. The perturbation itself is not novel. This is the easier task: simply selecting the nearest cell type from the training set is itself a strong baseline (Dong et al., 2026). Prompt-conditioned architectures (Dong et al., 2026) require examples of the perturbation effect at inference. Cross-cell-type transfer is a genuinely useful capability, but it is not virtual screening.

Held-out perturbation prediction. A gene is never perturbed during training in any context, and no examples of the perturbation's effect in any cell type are available at inference. The model must rely entirely on pretrained representations of gene function to predict how perturbing that gene will alter the transcriptome. This is the harder task, and the capability required for virtual screening—the perturbations of interest are, by definition, ones for which no experimental data exists. Prompt-conditioned architectures cannot attempt this task by construction.

2.2.3 Evaluation setup

We evaluate cross-cell-type transfer on the Nadig–Replogle dataset: 1,677 CRISPR knockouts across 4 cell lines (HepG2, Jurkat, K562, RPE1), with leave-one-cell-type-out (LOOV) cross-validation. We chose this dataset for its high UMI counts per cell (ensuring signal quality), breadth of perturbations, and natural four-fold cross-validation structure. The model observes all 1,677 perturbations, including the 380 held-out, in 3 of 4 cell lines during finetuning, then predicts their effects in the held-out 4th cell line.

Our models are finetuned on the full transcriptome (~9,600 genes), not the top 2,000 HVG subset used by all comparison models. This is strictly harder: the model must predict expression for thousands of genes with minimal variance, where the signal-to-noise ratio is lowest. We report MAE on both—evaluating on the HVG subset for direct comparison with prior work, and on the full transcriptome as the more comprehensive test. To our knowledge, no prior model reports full-transcriptome evaluation.

We compare against published results from Wang et al. (2026) (Table B7), who evaluate four models on the same split. X-Cell (Wang et al., 2026) is a bespoke diffusion language model architecture pretrained on 25.6 million perturbation profiles with a seven-component loss function and six injected knowledge sources via cross-attention (gene ontology, pathway databases, protein-protein interactions, transcription factor targets, drug-target interactions, and disease associations). STATE (Adduri et al., 2025), retrained by Wang et al. on matching data, follows a similar paradigm with perturbation-specific pretraining. Both represent the field's most sophisticated attempts at perturbation prediction through perturbation data. scGPT (Cui et al., 2024) and Cell2Sentence (C2S; Levine et al. 2023) complete the comparison. X-Cell finetunes models up to 3.1B parameters but reports that downstream performance plateaus beyond 1.6B. X-Cell is the current state of the art on this benchmark; STATE, despite architectural similarities, scores substantially worse (0.0864 vs 0.0682 on HVG MAE). We report results on the single fold reported by Wang et al. (2026) for all these models, HepG2, matching their exact 380 held-out perturbation split. Results on the harder, truly held-out setting—where the perturbed gene is never observed in any training context (Virtual Cell Challenge)—are forthcoming.

2.2.4 Cross-cell-type transfer: ranking-pretrained models achieve state-of-the-art perturbation prediction with monotonic scaling

Finetuning setup. The pretrained ranking head is replaced with a fresh prediction head outputting per-gene logits, and the model is finetuned with a proportional normlog loss on perturbed cell profiles. Both embedding and encoder weights are updated. The 2B parameter model was excluded from finetuning evaluation due to compute scheduling constraints. Details of the finetuning protocol, including the optional directional loss scaling (DSCALE), are provided in Methods (§4.6).

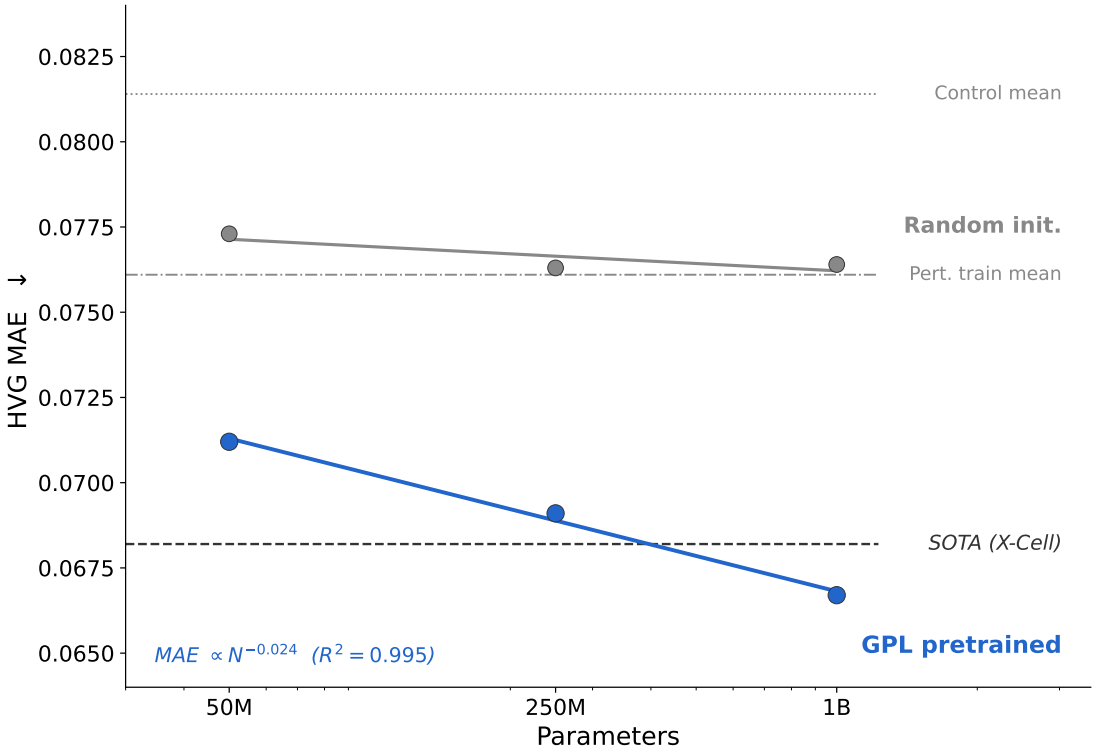

Figure 2. Perturbation prediction scales with pretraining. HVG MAE on the Nadig–Replogle HepG2 held-out fold as a function of pretrained model size. GPL-pretrained models (blue) improve monotonically: 0.0712 → 0.0691 → 0.0667. A log-log regression yields MAE $\propto N^{-0.024}$ ($R^2 = 0.995$). Randomly initialized models of equivalent capacity (gray) show modest initial improvement but saturate near the perturbation train mean, confirming that downstream transfer is driven by pretrained representations, not model capacity alone. Dashed lines indicate published baselines.

GPL-pretrained models improve monotonically with scale: 0.0712 → 0.0691 → 0.0667 (**Fig. 2**). At 1B parameters, the model achieves HVG MAE of 0.0667, compared to X-Cell's 0.0682 (**Table 1**). On the full transcriptome, our 1B-parameter model achieves MAE of 0.0551, beating the control mean baseline (0.0621)—a comparison no other model reports.

Randomly initialized models of equivalent capacity saturate near the perturbation train mean regardless of scale (0.0773, 0.0763, 0.0764), confirming that downstream transfer is driven by pretrained representations rather than model capacity. The pretraining advantage widens monotonically with scale ($\Delta = 0.006$ at 50M, 0.007 at 250M, 0.010 at 1B). Downstream MAE follows a power law in parameter count for pretrained models: a log-log regression yields MAE $\propto N^{-0.024}$ ($\alpha = 0.024$, $R^2 = 0.995$, three scales). X-Cell reports a MAE scaling exponent of $\alpha = 0.006$ across five scales (Wang et al., 2026)—four-fold flatter, and their best model plateaus at larger scales while ours continues to improve monotonically.

We use Wang et al.'s reproduction of published scores on this split, as publicly available evaluation results

Table 1. Perturbation prediction on Nadig–Replogle HepG2 held-out fold. Published results from Wang et al. (2026), Table B7. Best scores in bold.

Model	Params	HVG MAE	Full MAE
C2S	2B	0.2845	—
STATE	600M	0.0864	—
scGPT	53M	0.0804	—
X-Cell	≤1.6B	0.0682	—
GPL d=1024	1B	**0.0667**	**0.0551**
GPL d=512	250M	0.0691	0.0565
GPL d=256	50M	0.0712	0.0576
Control mean	—	0.0814	0.0621
Pert. train mean	—	0.0761	0.0594

show inconsistencies across model versions and evaluation settings.

On secondary metrics, our model ranks second on centroid accuracy (0.7290 vs X-Cell's 0.7876)—a perturbation discrimination metric complementary to MAE. Full results across all six metrics from Wang et al. (2026) Table B7 are reported in Supplement E, alongside additional discrimination and distance metrics and full-transcriptome results—which only our model can provide, as all benchmark models were trained exclusively on 2,000 HVGs.

2.2.5 Observational pretraining scales; perturbation pretraining does not

These results are from a single cross-cell-type transfer fold, and the margin over state-of-the-art on HVG MAE is narrow (0.0667 vs 0.0682). But the scaling dynamics are not narrow. A model pretrained exclusively on observational data with a single ranking objective—without perturbation data, injected knowledge, or task-specific architectural modifications—scales four-fold more steeply than models pretrained on tens of millions of perturbation profiles, whose performance plateaus at larger scales. The model that scales is the one that learned gene function from observation, not the one trained on the outcomes it is asked to predict.

2.3 Expression rankings encode regulatory grammar without binding data

2.3.1 Benchmark and method

Gene regulation is controlled by transcription factors (TFs)—proteins that bind specific DNA sequences in gene promoters to activate or repress transcription. To test whether our pretrained representations encode these regulatory relationships, we evaluated zero-shot GRN inference on ENCODE ENETS2 (Gerstein et al., 2012), a benchmark of ~26,000 TF–target edges derived from chromatin immunoprecipitation sequencing (ChIP-seq), an assay that maps where TF proteins physically bind to DNA. The benchmark covers five ENCODE cell lines (K562, GM12878, HeLa-S3, HepG2, H1-hESC), with peaks aggregated across cell types to capture cell-type-agnostic regulatory grammar. From the larger set of profiled factors, we restrict evaluation to sequence-specific TFs, excluding chromatin regulators and general transcriptional machinery, yielding 95 evaluable TFs (Supplement F).

For each TF, we rank all other genes by cosine similarity to the TF in the pretrained embedding space, without any task-specific training, learned probes, or regulatory labels. Genes closer to the TF in embedding space are, we find, more likely to be true regulatory targets. AUROC measures how well this ranking separates true ChIP-seq targets from non-targets—a binary classification over ~26,000 true edges among ~700,000 evaluated pairs (~3.7% positive rate), evaluated per-TF and averaged.

GRN methodological choices. We evaluate under the most stringent conditions available, with each methodological choice designed to minimize post-hoc degrees of freedom:

1. **Static embeddings only.** We use the embedding-layer representations (nn.Embedding) rather than hidden-state activations from intermediate transformer layers. Unlike hidden-state approaches, which require selecting a layer and attention head—choices that can be optimized post-hoc—the embedding matrix is a single, fixed representation with no selection degrees of freedom. At larger model scales, regulatory knowledge migrates from the static embedding into transformer layers, recoverable by hidden activation probes (Supplement F; cf. Tenney et al. 2019); we report the static embedding results as the conservative baseline.

2. **Input-independent evaluation.** Embedding-layer representations do not depend on the input context, eliminating a major confound: other approaches show large performance variation depending on which cell-type data is used as input (Bravo González-Blas et al., 2023). Our evaluation is deterministic and reproducible regardless of input.

3. **Fully zero-shot, no supervised baselines.** We use no regulatory labels, learned probes, or task-specific training. Supervised approaches, even with strict train-test splits, permit subtle forms of data leakage through class imbalance in TF–target labels across the genome (Supplement F). Zero-shot evaluation eliminates this risk entirely.

4. **Per-TF and overall evaluation.** We report both per-TF mean AUROC (averaging across TFs, giving equal weight to each regulatory program) and AUPR (which better captures performance under the heavy class imbalance of ~3.7% positive rate). Both metrics are reported in full in Supplement F.

2.3.2 Zero-shot gene embeddings recover transcription factor–DNA binding

Under these conditions, on the common-gene subset shared across all models, our 250M-parameter model achieves per-TF mean AUROC = 0.573 across 94 evaluable sequence-specific TFs, surpassing Tahoe-3B (0.562; 12× larger; Gandhi et al. 2025), scGPT (Cui et al., 2024) (0.559), Geneformer-316M (Theodoris et al., 2023) (0.549), and all other baselines (**Fig. 3B**; Supplement F). AUPR, which better captures performance on this heavily imbalanced benchmark, is comparable between our model and Tahoe-3B (Supplement F). Unlike Tahoe, which shows monotonic improvement with scale under masked language modeling, embedding cosine performance does not scale monotonically under ranking-based pretraining; regulatory knowledge migrates from the static embedding matrix into transformer layers at scale, recoverable by hidden activation probes (Supplement F). Binding data, peak thresholds, and threshold sensitivity analyses are detailed in Supplement F.

Among the top 25 genes ranked nearest to STAT2 by embedding cosine similarity, 40% have a STAT2 ChIP-seq peak at their promoters—a 19.3× enrichment over background. Enrichment is monotonically graded: 30% at top 50 (14.5×), 20% at top 100 (10×), 8.6% at top 500 (4.2×).

2.3.3 The model learns the sign of regulation: repressor targets are pushed away in embedding space

If the model learned only unsigned co-expression magnitude—the strength of co-variation regardless of direction—all TF targets would cluster near the TF in embedding space, regardless of whether the TF activates or represses them. Per-TF analysis reveals that the model has instead learned the *direction* of transcriptional regulation. The bottom of the per-TF AUROC ranking is dominated by established transcriptional repressors: REST (0.27), ZEB1 (0.35), MXI1 (0.40), and PRDM1 (0.43). An AUROC below 0.5 means the model ranks a TF's true targets *lower* in cosine similarity than non-targets—the targets are pushed *away* from the TF in embedding space. Four of the five lowest-scoring TFs are well-characterized repressors. Among 94 sequence-specific TFs, 17 show this negative-direction pattern, and the lowest-scoring among them are enriched for TFs with documented repressor function.

REST provides the clearest illustration. Under the standard cosine metric, REST achieves AUROC = 0.27—well below chance. This is not model failure. REST's neuronal gene targets are expressed in cell types where REST is low and silenced where REST is high. The model learned this anti-correlation as a geometric property: REST targets point *away* from REST. Under the inverted metric (−cosine similarity), REST's AUROC rises to

0.73.

The model recovered the sign of transcriptional regulation—activator targets pulled toward, repressor targets pushed away—without ever seeing binding data, perturbation readouts, or regulatory annotations. We note that signed Pearson correlation also produces a repressor geometry (REST achieves Pearson AUROC of 0.173, confirming the anti-correlation is detectable by simple correlation across cellular contexts); the unsigned null is the appropriate comparison for whether the *embedding* encodes direction (§2.3.4). The question is what else the embedding recovers beyond this shared signal.

2.3.4 Co-expression baseline

To quantify how much regulatory information the embedding captures beyond co-expression, we computed a Pearson co-expression baseline. For each TF, we correlated its expression with every target gene's expression across 154 HPA cell types—one correlation coefficient per gene, measuring how similarly two genes vary across cellular contexts. Genes were ranked by this correlation and evaluated against the same ENCODE ChIP-seq targets using the same per-TF AUROC pipeline, on the same gene set (7,315 genes at the intersection of ENETS2, embedding vocabulary, and HPA expression data; 7,285 targets). The co-expression baseline achieves mean per-TF AUROC of 0.550, compared to the embedding's 0.570—a modest but consistent advantage (embedding wins 57 of 95 TFs). We note that the Pearson co-expression baseline approaches or exceeds several published foundation model results on this benchmark (Supplement F); the full 95-TF comparison is in Supplement F.

The scatter (**Fig. 3A**) reveals a systematic biological pattern in where the two methods diverge. Pearson dominates for TFs whose regulatory programs define cell-type identity—HNF4A (liver, 0.740 vs. 0.678), E2F4 (cell cycle, 0.737 vs. 0.508), BCL11A (erythroid, 0.696 vs. 0.604), PAX5 (B cell, 0.601 vs. 0.516)—where co-expression across cell types *is* the regulatory relationship. The embedding dominates for TFs requiring compositional resolution—STAT2 (0.848 vs. 0.562), whose targets are regulated by a specific heterodimer that co-expression cannot distinguish from general pathway membership; ELK4 (0.575 vs. 0.352), which forms ternary complexes with SRF; and RFX5 (0.720 vs. 0.581), which regulates MHC class II genes as part of the RFX/RFX-ANK/RFXAP complex—and for TFs operating in rare cellular contexts, such as the pluripotency factors NANOG (0.725 vs. 0.557) and POU5F1 (0.647 vs. 0.423).

For rare-context TFs, the embedding's advantage may partly reflect training data composition—scBaseCount contains millions of stem cells, whereas the 154 HPA bulk cell types include few pluripotent contexts. But this explanation fails for STAT2: interferon signaling is active across the majority of the 154 cell types, giving Pearson abundant data, yet STAT2's co-expression AUROC (0.562) is unremarkable. The 29-point gap reflects representational capacity, not data availability—the ability to encode which specific combination of transcription factors drives a target gene, a compositional relationship that pairwise correlation across cell types cannot resolve regardless of sample size. We examined STAT2 in detail.

2.3.5 The STAT2 case study: from co-regulation to complex composition

STAT2 achieves the highest per-TF AUROC (0.851)—far above the next TF (NFKB1, 0.765). We examined it in detail to test whether the continuous cosine similarity between STAT2 and target gene embeddings—not just the binary target/non-target classification—predicts actual protein–DNA binding strength. We assembled four independent lines of evidence, each progressively further removed from the model's training data.

Genes embedded nearest to STAT2 are enriched for STAT2 ChIP-seq binding at their promoters. Across 10,518 genes with both model-derived cosine similarity scores (from the pretrained embedding matrix) and promoter annotations (TSS ±2.5 kb from hg19 refGene), we tested whether genes ranked most similar to STAT2 are more likely to have STAT2 ChIP-seq binding at their promoters (ENCODE IDR peaks, K562). They are: 40% of the top 25 genes have a STAT2 peak (19.3× over background), monotonically graded to 20% at top 100 (10×) and 8.4% at top 500 (4.2×). These embeddings are general-purpose—trained on pan-tissue expression data with no access to K562 binding data. That general regulatory grammar, learned from expression rankings across di-

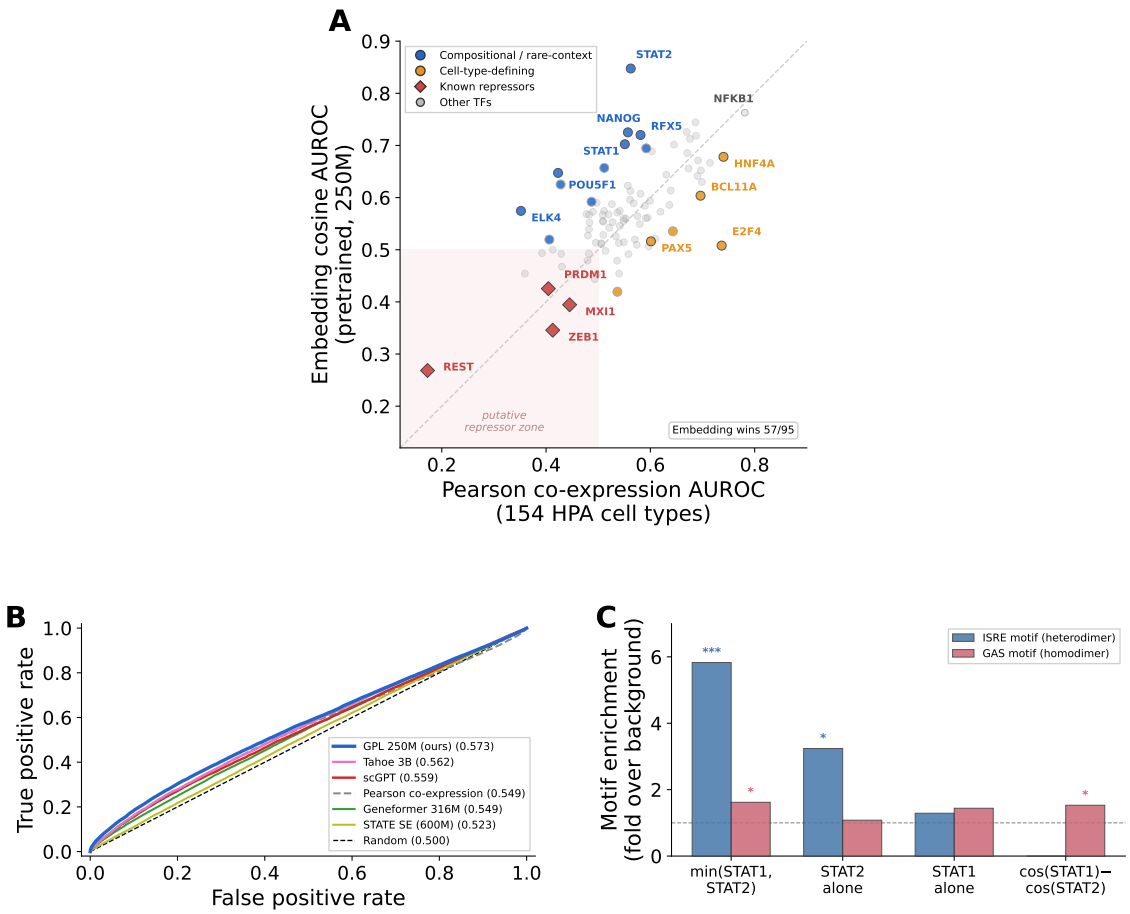

Figure 3. Zero-shot GRN inference from pretrained gene embeddings. **(A)** Per-TF AUROC for 94 sequence-specific transcription factors: embedding cosine similarity (y-axis) vs. Pearson co-expression across 154 cell types (x-axis). Blue: compositional regulators and rare-context TFs where the embedding outperforms co-expression (STAT2, RFX5, ELK4, NANOG, POU5F1). Orange: cell-type-defining TFs where co-expression captures the regulatory program (HNF4A, E2F4, BCL11A). Red diamonds: known repressors whose targets are pushed away in embedding space. Shaded region: putative repressor zone. **(B)** ROC curves for TF–target classification on ENCODE ENETS2 (common gene set, 94 seq-specific TFs). Per-TF mean AUROC in legend. **(C)** Promoter motif enrichment among the 50 genes nearest to each embedding-based ranking. Blue: ISRE motif (ISGF3 heterodimer binding site); red: GAS motif (GAF homodimer binding site). The min(STAT1, STAT2) ranking enriches 5.8× for ISRE while the cos(STAT1) − cos(STAT2) subtraction enriches for GAS and completely depletes ISRE—the embedding geometrically separates two regulatory complexes sharing a subunit. Stars: Fisher's exact test ($\star$ $p < 0.05$, $\star\star$ $p < 0.01$, $\star\star\star$ $p < 0.001$).

verse cellular contexts, correctly predicts cell-type-specific protein–DNA binding in a specific cell line. STAT1, which heterodimerizes with STAT2 in the Interferon-Stimulated Gene Factor 3 (ISGF3) complex, shows concordant enrichment (9.82×, $p = 6.6 \times 10^{-6}$). GATA1, a hematopoietic TF unrelated to the interferon pathway with abundant K562 ChIP-seq peaks, serves as a negative control and shows no significant enrichment ($p = 0.21$)—confirming pathway specificity. The genes with STAT2 ChIP-seq peaks among the top-ranked neighbors are canonical interferon-stimulated genes: STAT1, IRF2, IRF7, OAS2/3, IFI16, PML.

Enrichment is confirmed by DNA sequence motif independent of any binding assay. ChIP-seq measures where a protein binds DNA, but depends on peak-calling thresholds. To test the prediction independently of any binding assay, we scanned all 10,518 promoter sequences in the human reference genome (hg19) for the JASPAR STAT1::STAT2 binding motif (MA0517.1)—a 15-base-pair position weight matrix encoding the sequence preference of the STAT1::STAT2 heterodimer. Of 10,518 promoters, 325 (3.1%) carry the motif above threshold. Among genes whose embeddings have high cosine similarity to STAT2, the motif is significantly enriched (2.4-fold

at top 500, $p = 6.6\times10^{-7}$; full enrichment sweep in Supplement F). This rules out experimental artifacts but not co-expression—interferon pathway genes might carry the motif simply because they are interferon-responsive. The same interferon-stimulated genes appear in both ChIP-seq-positive and motif-positive sets (IRF7, OAS2, IFIT2, HERC6), confirming convergent evidence from protein binding and DNA sequence.

The model encodes heterodimer complex composition. STAT1 and STAT2 do not bind DNA independently—they heterodimerize as part of the ISGF3 complex (with IRF9; Darnell et al. 1994) and bind cooperatively at interferon-stimulated response elements. The JASPAR motif (MA0517.1) represents this heterodimer's binding site, not either factor alone. If the model has learned the complex composition, then requiring proximity to both subunits should produce stronger motif enrichment than proximity to either alone.

We tested this by ranking genes by proximity to different combinations of complex members (**Table 3**). The enrichment is monotonically graded across both ranking stringency and complex composition. At top 100: STAT1 cosine similarity alone shows no significant enrichment for the heterodimer motif (1.29×, $p = 0.37$), while STAT2 alone achieves 2.59× ($p = 0.01$). Requiring proximity to both—min(STAT1, STAT2)—increases enrichment to 5.18× ($p = 6.1\times10^{-8}$). Adding IRF9, the third component, pushes enrichment further: min(STAT1, STAT2, IRF9) reaches 9.06× at top 50 ($p = 2 \times 10^{-10}$). Each added complex member increases enrichment monotonically.

At the most stringent thresholds, the three complex members form a tighter core than even canonical downstream targets: min(STAT1, STAT2, IRF9) achieves 9.06× at top 50, compared to 6.47× for min(OAS1, STAT2)—where OAS1 is a prototypical ISRE target gene, not a complex member. This distinction diminishes at relaxed thresholds as the regulatory module broadens. The embedding encodes the ISGF3 program as a graded geometric structure: complex members at the core, canonical targets in the immediate neighborhood.

ChIP-seq binding validates this compositional structure independently of DNA sequence. Among the triple-min top 50 genes, 50% have STAT2 ChIP-seq peaks at their promoters—compared to 28% for STAT2-alone top 50, against a background rate of 2.1%. Requiring proximity to all three complex members in embedding space selects for genes that are physically bound in chromatin nearly twice as effectively as proximity to STAT2 alone.

GAS vs. ISRE—the model distinguishes two complexes sharing a subunit. STAT1 participates in two functionally distinct regulatory complexes. The GAF homodimer (STAT1:STAT1), activated primarily by IFN-γ, binds GAS (gamma-activated sequence) elements to drive macrophage activation and inflammatory gene programs (Eilers et al., 1995). The ISGF3 complex (STAT1:STAT2:IRF9), activated by IFN-α/β, binds ISRE (interferon-stimulated response element) sequences to drive antiviral effector programs (Li et al., 1996; Darnell et al., 1994). These complexes share STAT1 as a subunit but recognize different DNA motifs and regulate largely distinct target gene sets—the cell uses STAT1 in two different machines for two different jobs.

The embedding separates them geometrically (**Fig. 3C**). To test this, we ranked genes by the difference (cosine STAT1 minus cosine STAT2), selecting genes near STAT1 but far from STAT2—the putative GAF homodimer zone. These genes are enriched for the GAS homodimer motif (JASPAR MA0071.1; 1.39×, $p = 0.026$) and depleted for the ISRE heterodimer motif (JASPAR MA0517.1; 0.32×, $p = 0.04$ for depletion). At the tightest thresholds (top 25 and top 50), zero genes in the homodimer zone carry the ISRE motif—complete geometric exclusion. The converse holds: genes near both STAT1 and STAT2—min(STAT1, STAT2)—are strongly enriched for ISRE (5.18×, $p = 6.1 \times 10^{-8}$) but only weakly for GAS (1.53×). STAT2 alone enriches for ISRE (2.59×, $p = 0.012$) and not GAS (0.99×), consistent with STAT2 participating only in the heterodimer (**Table 2**).

2.3.6 The motif enrichment test: embeddings succeed where co-expression fails

The co-expression baseline demonstrates that cross-cell-type co-variation explains a substantial fraction of the embedding's regulatory signal—as expected, since co-expression is real regulatory biology. But the motif enrichment tests reveal a sharp boundary. Pearson-ranked STAT2 neighbors show no significant enrichment for the STAT1::STAT2 heterodimer motif at any threshold (1.62× at top 100, $p = 0.20$; 1.29× at top 500, $p = 0.14$). Every embedding ranking—from STAT2 alone to the full three-component minimum—is significant at $p < 10^{-5}$.

Table 2. GAS vs. ISRE motif enrichment across embedding-based gene rankings (10,518 gene universe). "S1−S2" denotes genes ranked by cosine(STAT1) − cosine(STAT2), selecting the putative GAF homodimer zone. At top 25/50, zero genes in the homodimer zone carry the ISRE motif while GAS is enriched—complete geometric separation. Stars denote significance: ⋆ $p < 0.05$, ⋆⋆ $p < 0.01$, ⋆⋆⋆ $p < 0.001$.

	ISRE (heterodimer motif, MA0517.1; background 3.1%)			
Ranking	Top 25	Top 50	Top 100	Top 500
min(S1, S2)	5.18× ⋆⋆	5.83× ⋆⋆⋆	5.18× ⋆⋆⋆	2.46× ⋆⋆⋆
STAT2 alone	3.88× ⋆	3.24× ⋆	2.59× ⋆	2.40× ⋆⋆⋆
STAT1 alone	2.59×	1.29×	1.29×	1.68× ⋆⋆
S1−S2 (homodimer zone)	**0.00**×	**0.00**×	**0.32**×	1.17×
Pearson STAT2	1.29×	2.59×	1.62×	1.29×
GATA1 (neg. ctrl)	2.59×	1.29×	1.29×	1.10×
	GAS (homodimer motif, MA0071.1; background 22.2%)			
Ranking	Top 25	Top 50	Top 100	Top 500
min(S1, S2)	1.62×	1.62× ⋆	1.53× ⋆⋆	1.20× ⋆
STAT2 alone	1.08×	1.08×	0.99×	1.11×
STAT1 alone	1.62×	1.44×	1.21×	1.21× ⋆⋆
S1−S2 (homodimer zone)	1.62×	**1.53**× ⋆	**1.39**× ⋆	1.01×

Table 3. ISRE motif enrichment sweep across ranking methods and thresholds (10,518 gene universe; background motif rate 3.1%). Enrichment increases monotonically with complex completeness at every threshold. Pearson is non-significant at every threshold.

Ranking method	Top 50	Top 100	Top 500
Emb: min(S1, S2, IRF9)	**9.06**× ($p = 2 \times 10^{-10}$)	**5.50**× ($p = 9 \times 10^{-9}$)	**2.59**× ($p = 3 \times 10^{-8}$)
Emb: min(OAS1, S2)	6.47× ($p = 2 \times 10^{-6}$)	5.18× ($p = 6 \times 10^{-8}$)	2.46× ($p = 2 \times 10^{-7}$)
Emb: min(S1, S2)	5.82× ($p = 2 \times 10^{-5}$)	5.18× ($p = 6 \times 10^{-8}$)	2.46× ($p = 2 \times 10^{-7}$)
Emb: STAT2 alone	3.24× ($p = 0.02$)	2.59× ($p = 0.01$)	2.40× ($p = 6.6 \times 10^{-7}$)
Emb: STAT1 alone	1.29× ($p = 0.46$)	1.29× ($p = 0.37$)	1.68× ($p = 0.007$)
GATA1 (neg. ctrl)	1.29× ($p = 0.46$)	1.29× ($p = 0.37$)	1.10× (NS)
Pearson STAT2	2.59× ($p = 0.07$)	1.62× ($p = 0.20$)	1.29× (NS)

Genes that co-vary with STAT2 across cell types are not the genes with STAT1::STAT2 binding motifs in their promoters. The embedding captures co-regulatory specificity that expression correlation cannot access.

2.3.7 Conditional co-regulation beyond pairwise correlation

The model learned conditional co-regulation—which genes are co-ranked with STAT1 when STAT2 is co-active versus absent—structure that pairwise correlation, which averages over all cellular contexts equally, cannot resolve regardless of sample size. A single-cell co-expression analysis conditioned on interferon-activated cells might recover comparable structure, but would require prior knowledge of which cells are in the relevant state, repeated for each regulatory program; the embedding encodes all such programs simultaneously from a single static representation. This parallels the co-expression ablation in §2.5, where co-expression rankings for SYK's signaling partners degrade with increasing B-cell purity while embedding rankings remain stable—in both cases, co-expression captures cell-type identity while the embedding captures functional relationships that operate within a cell type.

These results use only the static embedding matrix—no forward pass, no attention, no cellular context. Elementary algebraic operations in embedding space—cosine similarity, minimum across subunits, subtraction between partners—recover which complex a transcription factor acts through, which DNA motif that complex recognizes, and which genes it binds. No training, no labeled data, no task-specific architecture. The same operations applied to other multi-subunit complexes should yield comparable structure—testable predictions for any TF with known

binding partners and DNA motifs. Generative models trained on per-cell observations can only improve on this as they scale, and cell-type-conditional representations extracted from transformer hidden states should extend these capabilities further (Supplement F).

Having established that pretrained gene embeddings encode regulatory relationships—including the direction of regulation, the composition of multi-protein complexes, and the discrimination of distinct regulatory programs sharing a subunit—from expression rankings alone, we next asked whether the same representations encode the spatial organization of the cell.

2.4 Pretrained representations encode subcellular protein topology without structural data

The preceding section demonstrated that pretrained representations encode regulatory relationships—upstream structure. We now ask whether they also encode the spatial organization of the cell—downstream structure. This is a cross-modality test: the training data is transcriptomic, the ground truth is proteomic. There is no obvious reason why gene expression rankings should encode where proteins end up inside the cell, because subcellular localization is determined by molecular sorting signals—signal peptides, transmembrane helices, nuclear localization sequences—that are properties of protein sequence, not expression level.

We applied linear discriminant analysis (LDA) as a probe to gene embeddings, following the established approach of using linear projections to test what biological structure foundation model representations encode (Pearce et al., 2025). LDA finds directions in embedding space maximally separating labeled categories and cannot create structure that is not already present in the representation. We evaluated on 12,500 genes with subcellular localization annotations from the Human Protein Atlas (HPA), a proteomics resource based on immunofluorescence imaging that assigns each gene to one of 21 compartments (Thul et al., 2017). Five-fold cross-validation balanced accuracy is 12.3%—more than double chance (4.8%) across 21 imbalanced categories. Shuffled embeddings yield chance-level performance (4.8%), confirming that the spatial signal depends on learned representations rather than the probe's flexibility. A co-expression baseline achieves 9.6% (comparison with protein and DNA sequence models in Supplement G). The classification number is modest; the critical distinction is qualitative—the embedding's first discriminant axis produces the monotonic inside-to-outside gradient demonstrated below.

2.4.1 An inside-to-outside spatial gradient emerges from expression data alone

LDA receives only 21 coarse category labels—yet the first discriminant axis encodes a continuous spatial gradient that recapitulates the topology of the eukaryotic cell:

Nuclear speckles → Nucleoplasm → Nucleoli → Cytosol → Plasma membrane → Golgi → ER → Secreted

The separation between extremes is Cohen's $d = 1.97$ (nuclear speckles vs endoplasmic reticulum; **Fig. 4A**). That the resulting axis recapitulates the cell's spatial organization—with compartments arranged from nuclear to secretory—reflects structure present in the embeddings themselves, not the supervision.

The remaining axes capture additional organizational principles: LD2 separates host-encoded from mitochondrial genes (an evolutionary axis), LD3 separates structural/cytoskeletal proteins from secretory pathway residents ($d = 1.57$), and LD4 separates signal-receiving proteins from biosynthetic machinery ($d = 1.74$). These four axes are orthogonal—the representation simultaneously encodes spatial position, evolutionary origin, structural role, and functional mode. Control analyses confirm that the spatial signal persists after controlling for functional similarity (Supplement G).

2.4.2 Spatial signal is independent of function: the ribosome test

A natural concern is that the model learned gene function, and function correlates with location. The ribosome lifecycle test addresses this directly.

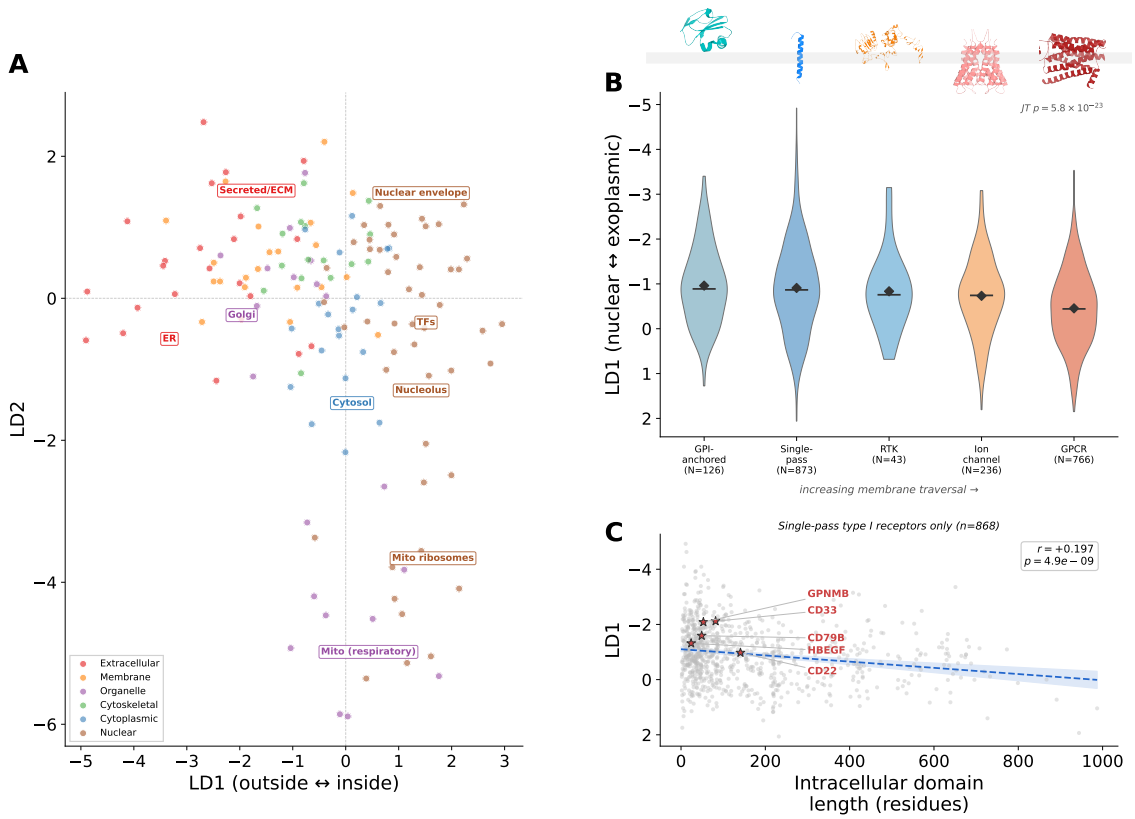

Figure 4. Pretrained representations encode subcellular protein topology. **(A)** Gene embeddings projected onto the first two LDA discriminant axes, trained on 21 HPA subcellular compartment labels. LD1 encodes a continuous inside-to-outside gradient from nucleus to secreted/ER proteins (Cohen's $d = 1.97$ between extremes). Colors indicate functional groups. **(B)** LD1 distributions for five receptor classes ordered by membrane traversal count. GPI-anchored (0 traversals, most exoplasmic) through GPCRs (7 traversals, least exoplasmic). Ordering is monotonic (Jonckheere–Terpstra $p = 5.8 \times 10^{-23}$). Receptor structure renders from PDB (top) illustrate increasing membrane traversal left to right; gray band indicates the lipid bilayer. **(C)** Within single-pass type I receptors ($n = 868$), intracellular domain length drives LD1 position ($r = +0.197$, $p = 4.9 \times 10^{-9}$). Red stars: approved ADC targets discussed in §2.5. Six outliers with ICD > 1,000 residues (ferlin family vesicle fusion proteins) excluded.

Nucleolar ribosome assembly factors (NCL, FBL, NOP56, NOP58, and 6 others; Methods) and cytoplasmic ribosomal proteins (RPL3, RPL5, RPS6, RPS8, and 9 others; Methods) participate in the same pathway but at different subcellular locations. In the raw embedding space, these two groups are highly similar—6× more similar to each other than to the genomic background (cosine similarity 0.04 vs 0.006). Yet on LD1, they separate cleanly: nucleolar assembly factors at LD1 = +1.42, cytoplasmic ribosomes at LD1 = +0.24. Cohen's $d = 2.72$, $p = 2.1 \times 10^{-6}$. The spatial signal overrides the functional similarity.

2.4.3 Representations encode finer structure than supervision labels

If LDA were simply recovering the supervision signal, all genes within a single HPA category would cluster identically, since they share the same label. The endoplasmic reticulum provides the clearest test. The single "ER" label in HPA conceals at least seven functional subgroups, which the model separates along LD1 in biologically meaningful order:

The ordering is biologically correct: lumenal chaperones and the translocon sit deepest (most negative LD1), while calcium-handling proteins at the ER membrane sit closest to the cytoplasmic end. LDA was trained on 21 coarse HPA labels—it never saw any finer-grained annotation. Gene Ontology Cellular Component terms, an entirely independent ontology not used anywhere in the LDA training or evaluation, confirm this substructure:

ER subcompartment	LD1 score
ER chaperones (BiP, GRP94)	−3.26
Translocon (SEC61, TRAP complex)	−3.05
ERAD quality control	−2.37
ER–Golgi transport (COPII, KDEL)	−1.56
Lipid synthesis (DGAT)	−1.31
CYP450s (drug metabolism)	−0.74
Calcium handling (SERCA, IP3R)	−0.51

GO "ER lumen" genes (GO:0005788, $n = 327$) group at mean LD1 = −1.59, while GO "ER membrane" genes (GO:0005789, $n = 487$) sit at mean LD1 = −1.05. The representation recovered a distinction that the supervision labels do not make and that an independent annotation system validates.

Notably, LD1 does not purely encode physical distance. The endoplasmic reticulum is physically adjacent to the nucleus—the outer nuclear membrane is continuous with the ER membrane—yet ER lumenal proteins sit at the opposite extreme of LD1. The reason is topological. Just as the body is a series of tubes whose lumina are continuous with the exterior—the digestive tract, the airways—so too is the cell: the ER lumen is continuous with the extracellular space via the secretory pathway (Cooper, 2000). Proteins translocated into the ER reach the cell surface through vesicular fusion without crossing another membrane. The *endo*plasmic reticulum is, luminally, exoplasmic. LD1 encodes not just where compartments sit, but which side of the membrane they are on. The Golgi apparatus replicates this membrane-face distinction independently—cisternal enzymes in the same exoplasmic zone as ER chaperones, cytoplasmic sorting adaptors near zero (Supplement G).

2.4.4 Heteromeric complexes resolve topology across the membrane

To test whether LD1 resolves topology across the membrane within individual signaling complexes, we examined 18 heteromeric receptor-kinase pairs where the receptor subunit is surface-exposed and its signaling partner is cytoplasmic. In 17 of 18 cases, the receptor subunit has a more negative LD1 value than its cytoplasmic partner (Wilcoxon $p = 1.9 \times 10^{-5}$). The single exception, EPOR, has notoriously low surface expression with most protein retained in the biosynthetic pathway. Dedicated receptor-kinase pairs (IFNAR1→IFNAR2, IFNGR1→IFNGR2) maintain correct ordering even when both subunits carry the same HPA label, while promiscuous shared signaling chains fail systematically (0/5). Of 85 surface kinases identified by DGIdb, only 27 carried HPA "Plasma membrane" labels; the model resolves topology where annotation databases do not.

2.4.5 Receptor architecture orders by membrane traversal count

Receptor classes order monotonically by membrane traversal count: GPI-anchored proteins (zero traversals, most exoplasmic) → single-pass type I (one traversal) → receptor tyrosine kinases (one traversal plus cytoplasmic kinase domain) → ion channels (4–6 traversals) → GPCRs (seven traversals, least exoplasmic; Jonckheere-Terpstra $p = 5.8 \times 10^{-23}$; **Fig. 4B**). Within single-pass receptors, intracellular domain length drives LD1 position ($r = +0.197$, $p = 4.9 \times 10^{-9}$; **Fig. 4C**) while extracellular domain size contributes minimally ($r = -0.068$)—the model encodes intracellular signaling burden, not extracellular sensing apparatus.

This structural ordering extends genome-wide: 85 surface-associated kinases separate from 1,824 cytoplasmic kinases on LD1 (Mann-Whitney $p = 1.4 \times 10^{-12}$, $d = 0.86$), and antibody targets separate from small molecule targets among surface proteins ($p = 5.4 \times 10^{-4}$). LD1 predicts signal peptide presence genome-wide (AUC = 0.754), consistent with the principal axis encoding the secretory pathway sorting decision.

Cosine similarity between receptor and kinase embeddings recovers known signaling partnerships: among 13 validated receptor-kinase pairs, 6 rank in the top 5% of 1,909 kinases genome-wide (e.g., CD8A→LCK rank 7). Cosine rank reflects cell-type specificity rather than interaction strength: TREM2 reaches its adaptor TYROBP (rank 4) but not its downstream kinase SYK (rank 1,649), because SYK is shared across myeloid lineages while TREM2 is microglia-specific.

2.4.6 Summary

The same embedding matrix, trained with a single ranking objective on observational expression data, recovers transcription factor binding, regulatory sign, protein complex composition, and subcellular spatial topology—each validated against independent experimental ground truth. It further resolves subcompartment structure within the endoplasmic reticulum that its own training labels do not distinguish. This convergence of biological properties in a single learned geometry parallels findings that language models trained on text acquire spatial and temporal representations of the physical world (Engels et al., 2025; Gurnee and Tegmark, 2024).

2.5 Embedding geometry organizes the druggable genome without pharmacological data

Antibodies, by virtue of their size, cannot penetrate lipid bilayers; they are confined to the exoplasmic face. Small molecules labor under no such constraint. We now ask whether the same axis that organizes the cell from nucleus to secreted proteins also organizes the druggable genome by therapeutic modality. This is a cross-domain transfer: the training data is transcriptomic, the ground truth is pharmacological. The model never saw a drug label, an interaction annotation, or a target classification during training.

2.5.1 Topology orders signaling cascades by therapeutic modality

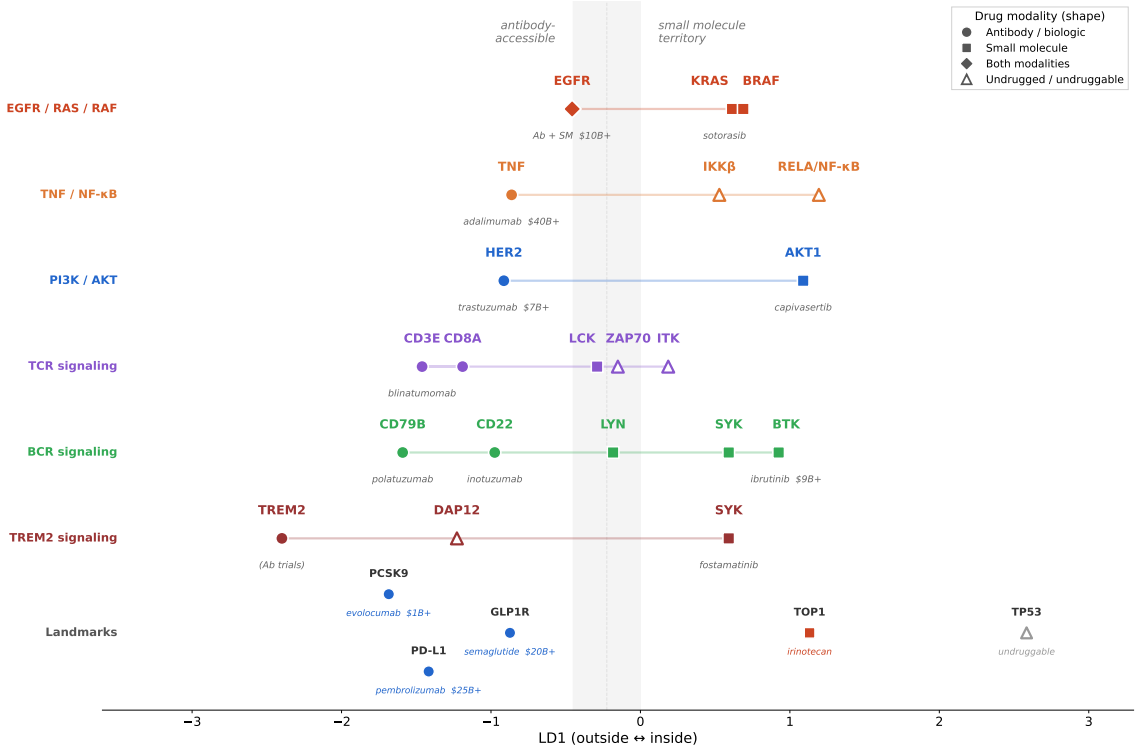

Figure 5. Six signaling pathways and landmark targets plotted on the first LDA discriminant axis (LD1). Each gene is positioned by its LD1 score; shapes indicate therapeutic modality (circle = antibody, square = small molecule, triangle = undrugged, diamond = both). Drug names and peak annual revenues are annotated. The TCR and BCR tracks reveal a conserved five-step signaling architecture ordered from exoplasmic to cytoplasmic on LD1. Approved therapeutics partition by membrane position—antibodies at surface nodes, small molecule inhibitors at cytoplasmic nodes. Not all pathway members order monotonically—PIK3CA and MAP2K1 land more exoplasmic than expected, potentially reflecting their membrane-proximal recruitment during signaling. The topology axis captures dominant localization, not transient states.

The model recovers parallel signaling architectures across independent immune pathways (**Fig. 5**). In the TCR cascade, the topology axis orders CD3E (−1.46) → CD8A (−1.19) → LCK (−0.29) → ZAP70 (−0.15)

Table 4. Parallel signaling architectures in TCR and BCR pathways. Five functional steps—receptor, co-receptor, Src-family kinase, SH2-recruited kinase, TEC-family kinase—are ordered identically on LD1 in both pathways. Antibodies target the surface nodes; small molecule inhibitors target the cytoplasmic nodes.

Step	TCR	LD1	BCR	LD1
Receptor	CD3E	−1.46	CD79B	−1.59
Co-receptor	CD8A	−1.19	CD22	−0.98
Src-family kinase	LCK	−0.29	LYN	−0.18
Recruited kinase	ZAP70	−0.15	SYK	+0.59
TEC-family kinase	ITK	+0.19	BTK	+0.93

Table 5. Cosine similarity versus co-expression for SYK neighbors. Embedding rank is computed genome-wide; Pearson ranks are computed within B cells at two purity levels (1,723 total B cells; 910 cells after removing myeloid-contaminated clusters). Functional partners rank highly in embedding space but deteriorate with increasing purity, while identity markers improve—demonstrating that the embedding encodes signaling relationships inaccessible to co-expression.

Gene	Role	Embedding	All B (1,723)	Purest B (910)
LAT2	SYK substrate	**4**	1,406	8,178 ↑
BLNK	SYK substrate	**7**	3,750	6,493 ↑
TEC	SYK effector	**11**	18,440	19,769 ↑
PIK3AP1	PI3K adaptor	**14**	7,475	17,305 ↑
LYN	Src kinase	**18**	428	6,212 ↑
CD79B	Identity marker	634	345	69 ↓

→ ITK (+0.19); the BCR cascade mirrors this: CD79B (−1.59) → CD22 (−0.98) → LYN (−0.18) → SYK (+0.59) → BTK (+0.93). Five functional steps—receptor, co-receptor, Src-family kinase, SH2-recruited kinase, TEC-family kinase—ordered identically in both pathways (**Table 4**), with approved therapeutics at multiple nodes. Antibodies target the surface nodes (polatuzumab, inotuzumab, blinatumomab); small molecule inhibitors target the cytoplasmic nodes (ibrutinib, fostamatinib). The drug modality boundary falls where the cascade crosses the membrane.

2.5.2 Cosine similarity recovers signaling mechanism beyond co-expression

The topology axis reveals where each signaling component sits relative to the membrane. But positioning alone cannot tell you which proteins interact, through what intermediates, or whether a scaffold with no kinase domain mediates the critical step between two kinases that cannot reach each other directly. Cosine similarity in embedding space recovers these relationships.

SYK's nearest scaffold in embedding space is its direct phosphorylation substrate BLNK (rank 7/36,591), despite BLNK having no kinase domain and no sequence homology to SYK. The embedding encodes functional interaction, not protein family membership. BLNK in turn reaches its recruited kinase BTK (rank 21), but SYK cannot reach BTK directly (rank 1,024)—the two-hop cosine chain traces the physical protein–protein interaction path through the BCR signaling cascade, recovering biochemistry from RNA without access to interaction databases.

This raises the question of whether cosine similarity simply recapitulates co-expression. Within purified B cells, SYK's direct substrates rank progressively worse as contaminating myeloid cells are removed—BLNK drops from 3,750th to 6,493rd, TEC from 18,440th to 19,769th—while B cell identity markers improve (CD79B: 345 → 69), confirming that constitutively co-expressed signaling cascade members do not covary within a cell type (**Table 5**).

The embedding recovers functional signaling modules inaccessible to co-expression at any resolution. Within the BCR pathway, the embedding separates activating from inhibitory co-receptors for the same kinase: CD79A→LYN ranks 71st while CD22→LYN ranks 14,949th—a 200-fold separation consistent with the tighter constitutive asso-

ciation between the activating co-receptor and LYN compared to the more conditional engagement of the inhibitory co-receptor, whose ITIM-mediated recruitment of LYN serves to dampen rather than propagate signaling.

2.5.3 Endocytic route prediction and the limits of static embeddings

Cosine neighborhoods recover signaling cascades and distinguish their regulatory logic. We next asked whether the same representation encodes the membrane trafficking biology that determines a drug's intracellular fate—the endocytic routes, adaptor couplings, and lysosomal routing that govern whether an antibody-drug conjugate delivers its payload.

The embedding resolves clathrin-mediated and caveolar endocytosis as distinct compartments: CLTC and CAV1 neighborhoods share zero genes in the top 50, recovering the complete caveolar complex (CAV2 rank 1, CAVIN1 rank 3, CAVIN3 rank 4) and clathrin coat machinery (CLTA rank 42, PICALM rank 49). TACSTD2/Trop-2 bridges exclusively to caveolar machinery through TM4SF1 (rank 7; CAV1 rank 23 from TM4SF1), with eight caveolar and zero clathrin bridges—a testable prediction for sacituzumab govitecan's internalization route.

Among surface targets for approved therapeutics, cosine neighborhoods differentially predict the presence of constitutive endocytic programs—and this distinction tracks clinical mechanism of action. CD20 shows zero endocytic bridges, consistent with rituximab's mechanism through complement and antibody-dependent cellular cytotoxicity rather than internalization-dependent payload delivery. CD33 presents a sharper case. It connects to neither its signaling effector SHP1 (PTPN6 rank 27,857) nor trafficking machinery—only myeloid identity markers. SHP1 can be recruited to CD33's ITIMs upon tyrosine phosphorylation, but this interaction is cellularly rare: most CD33 protein is normally nonphosphorylated, with only trace phosphorylation detected even after antibody stimulation, and the phosphatases Shp1 and Shp2 actively restore the dephosphorylated state when phosphorylation does occur (Walter et al., 2005, 2008). The embedding quantifies this rarity. LILRB3, a fellow ITIM-bearing inhibitory receptor in CD33's immediate cosine neighborhood—an independent negative prognostic factor on tumor-associated macrophages whose silencing reverses the immunosuppressive phenotype, consistent with constitutive inhibitory function (Zhuang et al., 2023; Yu et al., 2026)—reaches PTPN6 at rank 369—a 75-fold improvement over CD33 for the same biochemical interaction. Both receptors can recruit SHP1 through phosphorylated ITIMs; LILRB3 engages SHP1 constitutively across myeloid contexts, whereas CD33 almost never does. The static embedding encodes the frequency of a functional association across cellular contexts, not its existence as a biochemical interaction—and CD33's characterization as a target with suppressed endocytic capacity is consistent with gemtuzumab ozogamicin's clinical difficulties—withdrawn in 2010 for hepatotoxicity without clear survival benefit before re-approval at reduced doses in 2017.

Endocytic predictions are strongest where trafficking constitutively defines a gene's identity—as for TACSTD2 in caveolar microdomains, GPNMB in lysosomal pathways, and STAB1 as a professional endocytic receptor. The embedding's limitations follow the same principle. Where internalization is constitutive but ubiquitous within a cell type, the embedding captures what distinguishes the gene instead: CD22 cycles through clathrin pits constitutively, but the embedding places it near BCR signaling components (BLNK rank 15)—its identity as an inhibitory co-receptor, not as clathrin cargo. (CD22 and LILRB3 are both ITIM-bearing inhibitory receptors, but their embedding neighborhoods differ because their inhibitory programs serve different roles: LILRB3 provides tonic immunosuppression in myeloid cells—SHP1 engagement is its default function—while CD22 acts as a tunable brake on BCR activation, making its inhibitory program secondary to its identity as a BCR regulatory component.) Where internalization is incidental to a gene's primary role, the embedding captures that role: HBEGF internalizes efficiently enough to serve as the diphtheria toxin receptor, yet its neighborhood is wound-healing biology (PLAUR, PTGS2, EPHA2). CD30 illustrates the boundary most precisely: its neighborhood is pure immune signaling—IL2RA, IL5, IL10, IL23R, CCR8—with zero endocytic markers at any depth. Yet brentuximab vedotin works through clathrin-mediated internalization and lysosomal MMAE release (**?**). The reconciliation is that CD30 internalization is antibody-induced rather than constitutive—the static embedding correctly characterizes resting-state biology but cannot predict whether a therapeutic antibody will create the internalization event the target does not constitutively provide. In each case the embedding encodes what a gene is across cellular contexts.

Table 6. Expression-based characterization of surface proteins. LD1 placement and two-hop cosine neighborhoods provide functional context for Tbio genes with known therapeutic programs (calibration) and Tdark genes with no existing characterization. ECD and ICD denote extracellular and intracellular domain lengths (amino acids). DLL3 (LD1 = +0.30) falls outside the surface receptor zone—the screen would not have nominated this target. TMEM167A illustrates false positive detection: despite passing topological criteria, its two-hop neighborhood reveals ER retention machinery.

Gene	LD1	ECD	ICD	Two-hop markers	Assessment	Status
Tbio—known biology (calibration)						
GPNMB	−2.09	476	53	CLTA, RAB7A, CTSB, CTSD	Clathrin → lysosome	Phase II ADC
STAB1	−2.11	2453	71	DAB2, LGMN, CD209	Professional endocytic	Phase II Ab (bexmarilimab)
HBEGF	−1.32	141	24	PLAUR, PTGS2, EPHA2	Wound/inflammatory	DT receptor; Phase I mAb (KHK2866) terminated for neurotoxicity; no ADC
DLL3	+0.30	—	—	TIMM50, HES6	Intracellular	**Phase III ADC—failed**
Tdark—novel						
KIAA2013	−1.96	549	24	AP1M1, EXOC7, RAB11B, RAB5B	Forward trafficking	UniProt: uncharacterized; CB2 co-purification
SUSD1	−1.48	692	5	ITSN2, RAB14	Receptor w/ endocytic adaptor context	No characterization
CNTNAP3B	−1.11	1220	22	EXOC6, COLEC12	Vascular/endothelial	No characterization
Excluded (false positive caught by method)						
TMEM167A	−2.15	27	—	CALR, KDELR1, DNAJB11	ER resident	Flagged

It cannot represent what a therapeutic intervention makes it do in a specific one.

Cell-level representations cross this boundary. Mechanistic interpretability analysis of an earlier model—decomposing hidden activations into interpretable features using sparse autoencoders—recovered a single learned feature linking RAB21-mediated endosome transport to NF-κB signaling (IL6, PTGS2, NFKB2), active in barrier tissue cells—enterocytes, oligodendrocytes, decidua, centrilobular hepatocytes—and absent in pre-B precursors. Each link in this program has independent support in the literature: RAB21 promotes TRAF3–MKK3 association driving pro-inflammatory cytokine production; Rab21 depletion in *Drosophila* enterocytes induces the inflammatory cytokine Upd3 through Yorkie signaling. The feature was described prior to this literature review (Green, 2024), from an earlier model—the mechanistic validation is post hoc. In the static embedding, RAB21's neighborhood is trafficking machinery (ARL8B rank 3, ATP6V1C1 rank 6, SCYL2 rank 12) while inflammatory effectors form a separate cluster (TRAF3→NFKB2 rank 4, IL6→PTGS2 rank 16)—but RAB21→TRAF3 ranks 5,129th and RAB21→PTGS2 ranks 4,158th. The cell-level feature unifies what the static embedding separates: a trafficking program and an inflammatory cascade that co-activate in specific cellular contexts. The same boundary appears for CD33, whose two-hop relay through fellow ITIM receptors narrows the gap to PTPN6 75-fold (LILRB3→PTPN6 rank 369 versus CD33→PTPN6 rank 27,857) without closing it—a conditional association that cell-level representations, processing the specific myeloid states where ITIM phosphorylation occurs, should resolve.

2.5.4 From known targets to uncharacterized genes

Within the constraints of static representations, LD1 placement and cosine neighborhood inspection extend to genes with no existing druggability assessment. The LD1 topology gradient, validated against HPA immunofluorescence ($d = 2.43$) and UniProt receptor architecture ($p = 5.8 \times 10^{-23}$) in preceding sections, extends to Tdark genes: LD1 orders UniProt-annotated Tdark compartments in the expected sequence from ER to nuclear (Kruskal–Wallis $p = 4.4 \times 10^{-85}$). Among 428 Tdark genes with confirmed single-pass transmembrane domains, LD1 spans 7.7 units—from deeply ER-lumenal to cytoplasmic-facing. Sequence-based tools assign all 428 the same structural annotation; LD1 resolves the continuous placement within this class that determines therapeutic modality. For individual candidates, cosine neighborhoods provide functional context where none exists (**Table 6**).

To calibrate, we examined Tbio genes with known therapeutic programs: GPNMB (Phase II ADC) shows the complete clathrin → late endosome → lysosome pathway in its two-hop neighborhood; STAB1 (Phase II antibody, bexmarilimab) shows the endocytic adaptor DAB2 and scavenger receptor machinery. DLL3 (Phase III

ADC, rovalpituzumab tesirine—terminated for inferior survival after a $5.8 billion acquisition) passes every sequence criterion for a surface target—466-amino-acid ectodomain, single-pass type I, signal peptide—and UniProt annotates it as plasma membrane. Yet the embedding places it at LD1 = +0.30, outside the surface receptor zone, with neuroendocrine identity markers (HES6, ASCL1, INSM1) and zero trafficking machinery at any depth. DLL3 normally functions as a Golgi-retained cis-inhibitor of Notch signaling (**?**); in SCLC it is overexpressed and a fraction reaches the surface, but the embedding captures the dominant biology across contexts—a neuroendocrine lineage marker, not a constitutive surface receptor. The screen would not have nominated this target.

Applied to Tdark genes, the same analysis provides first functional context for uncharacterized proteins. KIAA2013—the 2,013th novel cDNA clone sequenced by the Kazusa Institute in the late 1990s—is a 634-residue single-pass membrane protein with a 549-amino-acid extracellular domain, a 24-amino-acid cytoplasmic tail, and a conserved DUF2152 domain. Twenty-five years after its discovery, UniProt still classifies it as "Uncharacterized protein KIAA2013 precursor." The model places it at LD1 = −1.96, deep in the exoplasmic zone where approved ADC targets cluster, with a cosine neighborhood of forward trafficking machinery: AP1M1 (adaptor protein complex 1, sorting cargo at the trans-Golgi for delivery to the plasma membrane), EXOC7 (exocyst complex, tethering secretory vesicles at the cell surface), RAB11B and RAB5B (recycling and early endosomal GTPases). This is not the ER retention machinery (CALR, KDELR1) that flags a false positive like TMEM167A—it is the machinery that moves cargo to the surface and cycles it back.

The few researchers who have studied KIAA2013 approach it through the endocannabinoid signaling system, where it was identified as a CB2-interacting protein in immune cells (Sharaf et al., 2019; Oyagawa and Grimsey, 2021). The model surfaces it from an orthogonal direction—not as a component of cannabinoid receptor signaling but as a candidate surface antigen with ADC-compatible geometry (549 aa ectodomain, 24 aa tail) and active trafficking biology, on immune cells expressing it at levels consistent with therapeutic targeting. The same protein, two completely independent lines of evidence for surface localization, zero overlap in the therapeutic hypothesis.

The analysis also catches false positives: TMEM167A passes topological criteria but its two-hop neighborhood identifies it as an ER membrane resident. Systematic characterization at the resolution these targets require will depend on cell-level representations that resolve context-dependent biology invisible to static embeddings.

The preceding analyses demonstrate that a single embedding, trained on expression rankings alone, encodes the therapeutic landscape from modality selection (LD1 drug boundary) through mechanism of action (multi-hop signaling cascades) to the trafficking biology that determines payload delivery—and that these representations correlate with clinical outcomes for approved therapeutics. For uncharacterized genes, the same two readouts—continuous topology and predicted interaction partners—convert each candidate into a concrete experimental program. For KIAA2013, classified as uncharacterized by UniProt twenty-five years after its discovery, the model predicts a deeply exoplasmic single-pass receptor (LD1 = −1.96, within the antibody-accessible zone defined at the opening of this section) with active forward trafficking biology—predictions independently convergent with its co-purification with the surface receptor CB2 in immune cells. The model's prediction—surface-localized, actively trafficked, immune cell context—is testable in an afternoon: surface biotinylation for localization, co-immunoprecipitation against AP1M1 for the predicted interaction, pHrodo uptake for internalization. Three experiments from one embedding, for a protein that has waited a quarter century for someone to ask what it does.

3 Discussion

We set out to test a simple hypothesis: the training objective is the missing ingredient for scalable self-supervised pretraining on single-cell data. The foregoing results suggest it is. Under identical architecture, the ranking objective outperforms count-based and proportion-based alternatives at every scale. These representations transfer: finetuned on perturbation data, ranking models predict the effects of held-out genetic perturbations, and this capability improves monotonically with pretraining scale.

Yet scale alone is not sufficient. Untrained models and perturbation-based pretraining both demonstrate that

model capacity per se does not drive downstream transfer; rather, the generalizable biological representations captured by this capacity when directed at the right training signal do. And when the right objective meets sufficient scale, these representations begin to encode biology far beyond the training signal.

Zero-shot performance on individual biological benchmarks does not yet rival dedicated models trained directly on these tasks—a stage familiar from other domains (Radford et al., 2018; Dosovitskiy et al., 2021). In each, task-specific architectures (Collobert et al., 2011), hand-engineered objectives (Lafferty et al., 2001), and curated datasets (Marcus et al., 1993) gave way to, and were ultimately surpassed by, simple self-supervised pretraining on abundant raw data—yielding general-purpose representations that transferred far beyond the training task distribution. We believe the single-cell field, and perhaps the broader enterprise of biological foundation models, is undergoing this same transition.

Inklings of such representations have already emerged in our model, which was never directly pretrained on transcription factor–DNA binding data or protein subcellular localization data. These results suggest that the statistical structure of gene expression encodes information about both the upstream regulatory processes that generate it and the downstream functional consequences of the gene products it represents, adumbrating a biological instance of the Platonic Representation Hypothesis (Huh et al., 2024)—the conjecture that models trained on different data modalities converge toward a shared statistical representation of the reality that generates these data. Sequencing reads, the shared denomination among biological measurement modalities, may provide a common currency in which this hypothesis can be cashed out empirically. A more speculative reading—that self-supervised pretraining on molecular measurements recovers not only the statistical structure of these measurements but a compressed world model of the cell—awaits further investigation (Green, 2024).

We make no claims of causal identification. That said, the distinction between observational and perturbational data is arguably a category error: what distinguishes an exogenous genetic perturbation from an eQTL or a paracrine signal? If observational pretraining recovers gene regulatory networks, encodes protein localization, and predicts perturbation outcomes, then the claim that paired interventional data is a prerequisite for learning cellular dynamics probably is a canard.

Our models improve on existing baselines for perturbation prediction, but it remains unclear what level of predictive accuracy would constitute experimental reproducibility. The economics are sobering regardless: if one wanted to test all triple-gene knockouts in a single human cell type, the combinatorial space alone approaches 10^{12}. This is intractable even in silico, let alone in the wet lab—trawling this space is no more viable with FLOPs than with bioFLOPs. Yet the field's preoccupation with perturbation prediction as the benchmark of virtual cell utility obscures a broader set of capabilities these representations afford. These models may prove more immediately valuable not as simulators but as specimens—objects whose internal representations, once elucidated, reveal the regulatory vocabulary through which cellular behavior can be understood and ultimately directed (Green, 2024, 2025). This vocabulary provides the coordinate system for iterative navigation: chart the most promising regions of therapeutic design space, deploy targeted experiments to explore them, and adjust course with each iteration—using FLOPs to better allocate bioFLOPs. Even an imperfect surrogate, if directionally correct, can guide and accelerate this search—virtual screens take seconds; real ones take months. And these are the worst these models will ever be.

Yet biology is flesh and blood. Single-cell profiling has achieved extraordinary resolution by dissociating cells from their native milieu, but almost no cell is an island. The path from virtual cell to virtual organism therefore passes through biopsies and archival tissue blocks—millions of them, sitting in hospital basements. As we thaw and assay these samples, the counts grow sparser and the signal less reliable—degraded in time through fixation, in space through sampling geometry—but the ordinal structure these counts induce remains. The substrate is noisy bags all the way up (and perhaps the principle demonstrated here extends further still, from psychometric assessment to ecological abundance estimation—and reinforcement learning from human feedback, too (Rafailov et al., 2023)).

4 Methods

4.1 Architecture and input representation

Each training example is a single cell's transcriptome: a multiset of integer counts across ~36,000 genes, with no natural ordering among genes. We perform autoregressive prediction over a randomly permuted subsequence of each cell's expressed genes, using a Transformer decoder with causal attention—analogous to permutation language modeling (Yang et al., 2019), adapted for biological multisets.

Input construction. For each cell, expressed genes (count ≥ 1) are selected and a subsequence of up to 2,048 is sampled. These genes are fully randomly permuted—the model sees a different gene ordering at every training step. A small proportion (~10%) of undetected genes (count = 0) are appended to provide the model with a detection signal. Each gene is represented by a learned token embedding (gene identity) combined with its expression rank (injected via RoPE; see below). Training is autoregressive—each position attends only to preceding positions in the permuted sequence via causal attention.

Positional encoding. A flexible variant of rotary position embeddings (RoPE; Su et al. 2021) encodes expression **ranks**, not sequence positions. Each gene's rank—its position in the expression ordering of that cell—is passed as the positional argument to RoPE, injecting relative expression level directly into the attention computation via per-head rotations. Because the gene sequence is randomly permuted while the rank encoding is preserved, the model learns order-invariant representations conditioned on expression rank. The attention pattern is thus informed by "gene A is more highly expressed than gene B" regardless of where A and B appear in the permuted input sequence.

Prediction. At each sequence position, the model outputs a logit (lambdit) for every gene in the full vocabulary (~36,000 genes), conditioned on the preceding genes in the permuted sequence via causal attention. These per-gene logits parameterize the GPL ranking likelihood (see below). The training signal at each position is the complete expression ranking of the cell—not just the identity of the next gene.

Metadata conditioning. Cell-level metadata (tissue, disease, sequencing chemistry, cell preparation method) is encoded as learned categorical embeddings and injected via adaptive layer normalization (AdaLN), providing per-layer scale and shift parameters. This conditions the model on experimental context without requiring it to learn batch effects from expression patterns alone.

Scaling. Model width is scaled via width-only muP (Yang et al., 2022), ensuring hyperparameter transfer across model sizes: the same learning rate, weight decay, and initialization are used at all scales. We train models at $d_{\text{model}} \in \{256, 512, 1024, 1536\}$ with $d_{\text{head}} = 128$, corresponding to approximately 50M, 250M, 1B, and 2B parameters. All models use QK-normalization for training stability at scale.

The architecture is deliberately simple. Our claim is that the training objective—not architectural innovation—is the key variable for scaling biological foundation models.

4.2 Training objective: Geometric Plackett–Luce (GPL)

Rather than reconstructing expression counts, we train on the ranking induced by observed counts. Training on rankings requires a generative likelihood over permutations that handles ties and scales to large gene vocabularies. The Plackett–Luce model (Plackett, 1975; Luce, 1959) provides a sequential factorization for rankings—the probability of a permutation decomposes as a product of softmax-like selection steps—but assigns zero probability to ties. With thousands of genes tied at zero count per cell, marginalizing over tie-consistent permutations is intractable. The Geometric Plackett–Luce model (GPL; Henderson 2022) resolves this by replacing the continuous exponential latent variables of standard Plackett–Luce with their discrete counterparts—geometric random variables. Because the geometric distribution has positive probability mass on integer values, ties arise naturally when multiple items draw the same waiting time. The resulting likelihood accommodates ties through a group factor that depends only on item parameters and group size, while inheriting the tractability and sequential factorization of

standard Plackett–Luce. The intellectual lineage from psychometric measurement theory through choice modeling to ranking likelihoods for biological data is detailed in Supplement A.

4.2.1 The GPL likelihood

The model outputs a logit ϕ_k for each gene k, from which we compute $\theta_k = \sigma(\phi_k) \in (0, 1)$.

GPL models the ranking as a "race" among Geometric random variables. Each gene draws a waiting time $W_k \sim \text{Geometric}(\theta_k)$: $P(W_k = w) = \theta_k(1 - \theta_k)^w$. Higher $\theta \to$ shorter expected wait $\to$ higher rank $\to$ more expressed. Genes are ranked by their waiting times; ties occur naturally because the Geometric distribution is discrete. We never run this race—we observe the ranking from data and compute its exact probability.

At each stage j of the ranking (all genes at rank j and below still "racing"), the per-stage probability is:

$$P_j = \frac{\theta_j \cdot \prod_{l>j}(1 - \theta_l)}{1 - \prod_{l \geq j}(1 - \theta_l)} \tag{1}$$

for untied positions, with a corresponding tied-group factor when multiple genes share a rank. The total log-probability is the sum of $\log P_j$ over all stages. Suffix products become suffix sums in log space, yielding an $O(V + S \log S)$ computation (see below). The group factor for tied items depends only on θ values and group size, not on any internal ordering—making tie equivariance a mathematical property of the distribution.

Notably, GPL does not require a separate expression classifier. The ranking likelihood over the complete gene set—including the tied-zero bucket—implicitly trains the model to distinguish expressed from unexpressed genes. Binary detection falls out of the ranking math (Supplement A).

The full derivation and connection to Thurstone's Law of Comparative Judgment (Thurstone, 1927) is provided in Supplement C. The development path from standard Plackett–Luce to GPL is detailed in Supplement A.

4.2.2 Computational tractability

Naive evaluation of the GPL likelihood over V genes with S expressed genes scales as $O(V \cdot S)$. We exploit the structure of single-cell data—where all unexpressed genes (50–95% of the vocabulary) form a single tied group—to factorize the likelihood into $O(V)$ order-independent terms (contributions from the zero bucket) and $O(S \log S)$ order-dependent terms (contributions from expressed genes). The total cost is $O(V + S \log S)$, making ranking-based pretraining on ~36,000-gene vocabularies computationally tractable.

4.2.3 Depth conditioning

All objectives share a per-cell logit shift that removes the first-order effect of sequencing depth. Given total UMI u and vocabulary size V, the expected count per gene under uniform expression is $\mu = u/V$, and the corresponding Geometric parameter is $p = 1/(\mu + 1)$. All logits are shifted by $\text{logit}(p) \cdot s_0$, where s_0 is a learned scale factor. After the shift, each gene's logit encodes deviation from the depth-expected baseline—how much more or less expressed the gene is relative to what sequencing depth alone would predict. This ensures depth is not a confound in the ablation: all objectives receive the same first-order depth correction.

4.2.4 Regularization

We apply z-loss regularization $\alpha \cdot (\log \sum_k e^{\phi_k})^2$ to prevent unbounded logit growth, following Chowdhery et al. (2022).

4.3 Training data and procedure

We pretrain on human single-cell RNA-seq data from scBaseCount (Youngblut et al., 2025), a uniformly processed repository of over 500 million cells mined from the Sequence Read Archive across 75 tissues. Gene expres-

sion profiles are preprocessed into tokenized training examples, each containing up to 2,048 genes from a single cell with expression counts and rank annotations.

To evaluate generalization to genuinely unseen biology, we hold out two tissue-disease groups entirely from pretraining: "healthy-respiratory" (healthy cells from respiratory tissue) and "other-unclear" (non-healthy cells with unclear tissue classification). These groups receive zero sampling weight during training and are used exclusively for evaluation.

Training details. All models are trained with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\varepsilon = 10^{-14}$) with linear warmup and cosine cooldown (final 20% of training). Learning rate and weight decay are scaled with batch size via muP conventions. Gradient clipping is applied at norm 1.0. Training uses bfloat16 mixed precision.

Scale	d_{model}	Layers	Heads	Parameters	Total tokens	Tok/param
XS	256	32	2	50M	~2.5B	~50
Small	512	32	4	250M	~10B	~41
Medium	1024	32	8	1B	~40B	~44
Large	1536	32	12	2B	~80B	~40

Parameter counts include the gene embedding matrix ($\sim 36{,}604 \times d_{\text{model}}$), per-layer metadata conditioning, and the output head. All models within the ablation (GPL, GC, PROP, MSE) are trained for the same number of tokens at each scale, ensuring compute-matched comparison. Token-to-parameter ratios of approximately 40 are roughly 2× the Chinchilla-optimal ratio for language, which may reflect the lower information density per token in gene expression data relative to natural language.

4.4 Ablation design

To isolate the effect of the training objective, we train four objectives at each of three scales ($d \in \{512, 1024, 1536\}$) under identical conditions—same architecture, same training data, same compute budget, same muP hyperparameters—varying only the loss function:

1. **GPL (Geometric Plackett–Luce).** Loss computed on the ranking induced by observed counts. Tied genes contribute a single group factor. Never sees a count value.
2. **Geometric Count (GC).** Per-gene negative log-likelihood under the Geometric distribution: $-\log P(k \mid \theta) = -[\log \theta + k \log(1-\theta)]$. Same $\theta = \sigma(\phi)$ parameterization as GPL, same depth shift. Differs from GPL only in computing loss on individual counts rather than the ranking.
3. **Proportional (PROP).** Softmax over all ~36,000 genes produces normalized proportions ρ_g; loss is cross-entropy against observed proportions (log-normalized counts divided by total UMI). Unlike GC, this couples all gene predictions through a shared normalizer—expressed and unexpressed genes compete for probability mass.
4. **MSE on log-normalized counts.** The standard baseline used by several existing models. Mean squared error between predicted and observed $\log(1 + \text{counts}/\text{total\_UMI} \times 10{,}000)$.

We additionally tested Negative Binomial (NB), which generalizes GC by adding a learned per-gene dispersion parameter—the standard distributional assumption in scRNA-seq analysis (Lopez et al., 2018). Setting the dispersion to 1 recovers GC. NB represents the most flexible count-based objective in our comparison. Implementation details and results are reported in Supplement A.

Two-step decomposition. The four objectives form a controlled decomposition. GPL and GC both avoid softmax coupling of zeros (independent per-gene likelihoods); PROP does not. GPL and GC share the same Geometric parameterization and differ only in the training target (rankings vs counts). Thus: GPL/GC vs PROP isolates zero coupling as a capacity sink; GPL vs GC isolates the ranking objective from count fitting. Both comparisons are

needed—without PROP, we cannot prove zeros matter; without GC, we cannot prove rankings matter beyond zero handling.

4.5 Pretraining evaluation

We evaluate pretraining quality by Spearman rank correlation between predicted and observed gene rankings on held-out cells. The model's output logits induce a ranking over the full vocabulary (~36,000 genes); ground-truth rankings are derived from observed expression counts, with undetected genes tied at the lowest rank. Ties are resolved using the standard midrank (fractional rank) convention. We report results stratified by cell quality: cells with fewer than 4,096 total UMI are excluded to ensure reliable ground-truth rankings.

At evaluation, the model observes 50% of a held-out cell's genes as context and predicts logits over the full vocabulary. All objectives are evaluated using the same pipeline: raw output logits are argsorted to produce a predicted ranking, which is compared to the ground-truth ranking via Spearman correlation. No objective-specific transformation is applied. The evaluation is objective-agnostic—no objective is privileged by the use of a rank-based evaluation metric, since all objectives produce per-gene logits that can be ranked (Supplement D).

We report two complementary metrics:

Full-transcriptome Spearman (`spearmanscipy`): Spearman correlation computed over all ~36,000 genes, including the undetected (typically 50–80% of the vocabulary). This is the most stringent metric, as it requires the model to correctly predict both the ranking among expressed genes and which genes are expressed at all.

Binary correctness: The fraction of truly expressed genes that the model predicts as expressed—a set-overlap measure of detection accuracy, independent of ranking quality.

4.6 Perturbation prediction

Input construction. Each finetuning example is a single perturbed cell. The sequence begins with a learned perturbation embedding that identifies the target gene (the gene that was knocked out), followed by a randomly sampled subsequence of the perturbed cell's gene expression profile. The model conditions on this perturbed context and predicts expression across the full gene vocabulary. No control (unperturbed) cells appear in the input—the model learns perturbation effects directly from perturbed cell profiles, conditioned on the perturbation identity.

Finetuning protocol. The pretraining output head is replaced with a fresh two-layer MLP ($d_{\text{model}} \to 4 \times d_{\text{model}}$, SiLU, $\to V$ genes), initialized with muP-scaled weights. The pretrained encoder is unfrozen. Finetuning is trained for 3–20 epochs with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.97$).

Finetuning loss. Rather than GPL, finetuning uses a count-based loss—demonstrating that GPL-pretrained representations transfer to count-based downstream tasks. Specifically, the loss is KL divergence between predicted and observed expression profiles in log1p-normalized proportion space, matching the normalization conventions established by the Virtual Cell Challenge (VCC) benchmark.

Directional loss scaling (DSCALE). Optionally, the per-gene loss can be reweighted by the magnitude of the ground-truth differential expression (perturbation minus control). Genes with larger perturbation effects receive proportionally larger loss weight, encouraging the model to prioritize getting the direction and magnitude of perturbation effects correct rather than minimizing error uniformly across all genes. The `ease` parameter controls the strength of this reweighting. Critically, DSCALE uses only training-set ground truth—held-out perturbations never contribute to the loss weights, so there is no information leakage. DSCALE exposes a systematic tradeoff between absolute expression accuracy (MAE) and perturbation discrimination (dPDS): increasing the ease parameter improves discrimination at a modest cost to absolute error.

Evaluation. For the Nadig–Replogle benchmark, we use leave-one-cell-type-out (LOOV) evaluation: the model is trained on perturbation data from three of four cell lines (HepG2, Jurkat, K562, RPE1) and tested on the held-

out cell line. This tests cross-cell-type transfer—whether a model trained on perturbation effects in one cellular context can predict effects in an unseen context.

Predicted and observed profiles are pseudobulked (averaged across single cells from the same condition) before metric computation.

We report mean absolute error (MAE): the average absolute difference between predicted and observed pseudobulked expression across all genes, $\text{MAE}_k = \frac{1}{G}\sum_g |\hat{y}_g^k - y_g^k|$, averaged over perturbations. A trivial mean baseline (predicting the unperturbed average for every perturbation) performs surprisingly well on MAE. Additional metrics (centroid accuracy, Pearson Δ, DE overlap) are reported in Supplement E.

4.7 Gene regulatory network evaluation

We evaluate zero-shot GRN inference on ENCODE ENETS2 (Gerstein et al., 2012), a benchmark of ~26,000 TF–target regulatory edges derived from ChIP-seq binding data across 98 sequence-specific transcription factors in five ENCODE cell lines (K562, GM12878, HeLa-S3, HepG2, H1-hESC). For each TF–target pair, we compute the cosine similarity between their pretrained gene embeddings. No task-specific training or learned probe is used. We report AUROC: the probability that a true TF–target pair scores higher than a random non-target pair.

For repressor TFs, whose targets are expected to be anti-correlated with the TF in embedding space, we additionally report AUROC under the inverted metric (negative cosine similarity). The STAT2 case study (main text) validates embedding predictions against three independent lines of evidence—ChIP-seq binding (ENCODE IDR peaks), DNA sequence motifs (JASPAR PWM scan, no ChIP-seq), and heterodimer complex composition—with appropriate negative controls. Supplement F provides full per-TF results, peak stringency analysis, and anticipated critiques.

4.8 GRN extended analyses

Methodological details for the Pearson co-expression baseline, motif enrichment analysis, ChIP-seq enrichment analysis, co-expression heatmap, and GAS vs ISRE complex discrimination are provided in Supplement F.

Pearson co-expression baseline for GRN. For each of 95 sequence-specific transcription factors, we computed signed Pearson correlation between the TF's 154-dimensional expression vector (HPA v24 nTPM) and every target gene's expression vector. Genes were ranked by this correlation and evaluated against the same ENCODE ENETS2 ChIP-seq targets using the identical per-TF AUROC pipeline, on the same gene set (7,315 genes at the intersection of ENETS2 targets, embedding vocabulary, and HPA expression data). The co-expression baseline and embedding evaluation use the same cross-TF comparator design: for each TF, positives are the TF's ENETS2 targets and negatives are genes targeted by other ENCODE TFs but not by the focal TF.

Motif enrichment analysis. Promoter sequences were extracted from hg19 refGene annotations (±2.5 kb from TSS) for 10,518 genes with both embedding cosine similarity scores and promoter annotations. The JASPAR STAT1::STAT2 position weight matrix (MA0517.1, 15-bp PWM) was scanned against all promoter sequences using a fixed score threshold. 325 of 10,518 promoters (3.1%) carry the motif above threshold, establishing the background rate. For each ranking method, genes were ranked and the top N genes ($N = 25, 50, 100, 200, 500$) were tested for motif enrichment by one-sided Fisher's exact test against the 3.1% background. The compositional test ranks genes by the minimum cosine similarity to multiple complex members—$\min(\text{STAT1}, \text{STAT2})$ requires a gene to be embedded near both factors.

ChIP-seq enrichment analysis. STAT2 and STAT1 ChIP-seq peaks in K562 were obtained from the ENCODE portal (IDR-thresholded narrowPeak files, hg19 assembly). Background binding rates: STAT2 peaks at 217 of 10,517 evaluable promoters (2.1%, excluding the query TF), STAT1 peaks at 74/10,517 (0.7%), co-binding at 17/10,517 (0.2%). ChIP-seq enrichment was computed at the same top-N thresholds as motif enrichment.

GAS vs ISRE complex discrimination. All 10,518 promoters were scanned for two motifs: JASPAR MA0517.1

(ISRE, recognized by the ISGF3 complex; background rate 3.1%) and JASPAR MA0071.1 (GAS, recognized by the GAF complex; background rate 22.2%). Genes were ranked by four methods: STAT2 cosine similarity alone, STAT1 cosine similarity alone, min(STAT1, STAT2), and the difference (cosine STAT1 − cosine STAT2) selecting the putative GAF homodimer zone. The subtraction operation tests a specific geometric prediction: if the embedding encodes complex composition, homodimer targets and heterodimer targets should occupy geometrically distinct regions identifiable by their relative proximity to the shared (STAT1) and distinguishing (STAT2) subunits.

4.9 Subcellular localization evaluation

We evaluate subcellular localization prediction using 12,500 genes with annotations from the Human Protein Atlas (HPA; Thul et al. 2017), which assigns each gene to one of 21 subcellular compartments based on immunofluorescence imaging. We apply linear discriminant analysis (LDA) as a probe on frozen pretrained gene embeddings (512 dimensions). LDA finds directions in embedding space that maximally separate the 21 HPA categories and cannot create structure not already present in the representation.

We report five-fold cross-validation accuracy. As a negative control, we shuffle which gene gets which embedding vector, destroying gene-specific structure while preserving the distributional properties of the embedding space, and repeat the LDA analysis. For cross-modality comparison, we apply the identical LDA pipeline to embeddings from ESM2 (Lin et al., 2023) and ESM-C (Hayes et al., 2024) protein language models and Evo 2 (Brixi et al., 2025) DNA language models, with PCA to 256 components applied before LDA to ensure fair comparison across embedding dimensions. Supplement G provides full results including confusion matrices and fine-structure analysis.

Ribosome lifecycle test. Nucleolar ribosome assembly factors (n = 10): NCL, NPM1, FBL, NOP56, NOP58, DKC1, GAR1, NHP2, BYSL, WDR12. Cytoplasmic ribosomal proteins (n = 13): RPL3, RPL5, RPL7, RPL8, RPL11, RPL13, RPS3, RPS6, RPS8, RPS14, RPS18, RPS19, RPS27. Cohen's d and p-value from two-sample t-test on LD1 projections. Cross-group cosine similarity computed as the mean pairwise cosine between the two groups in the raw 512-dimensional embedding space; background cosine computed from 10,000 random gene pairs.

ER fine structure. Seven ER subcompartments were defined by curated gene lists: (1) ER chaperones (n = 8): HSPA5, HSP90B1, CALR, CANX, PDIA3, PDIA4, P4HB, HYOU1; (2) Translocon (n = 9): SEC61A1, SEC61B, SEC61G, SSR1, SSR2, SSR3, SSR4, SEC62, SEC63; (3) ERAD (n = 12): DERL1, DERL2, DERL3, SEL1L, SYVN1, EDEM1, EDEM2, EDEM3, OS9, ERLEC1, UGGT1, UGGT2; (4) ER–Golgi transport (n = 13): SEC23A, SEC23B, SEC24A, SEC24B, SEC24C, SEC24D, SEC13, SEC31A, KDELR1, KDELR2, KDELR3, SAR1A, SAR1B; (5) Lipid synthesis (n = 7): DGAT1, DGAT2, SCD, HMGCR, SQLE, ACAT1, ACAT2; (6) CYP450s (n = 17): CYP1A1, CYP1A2, CYP1B1, CYP2A6, CYP2B6, CYP2C8, CYP2C9, CYP2C19, CYP2D6, CYP2E1, CYP3A4, CYP3A5, CYP4A11, CYP4F2, CYP51A1, CYP7A1, CYP27A1; (7) Calcium handling (n = 10): ATP2A1, ATP2A2, ATP2A3, ITPR1, ITPR2, ITPR3, RYR1, RYR2, STIM1, STIM2. LD1 scores are mean projections across each group. GO:CC validation used Gene Ontology terms GO:0005788 (ER lumen, n = 327) and GO:0005789 (ER membrane, n = 487), queried via MyGene.info.

Single-pass ICD correlation. Within single-pass type I receptors (n = 868), intracellular domain (ICD) length was correlated with LD1 position. Six genes with ICD > 1,000 residues—the ferlin family of vesicle fusion proteins (DYSF, OTOF, MYOF, FER1L5, FER1L6) plus ITGB4—were excluded as non-receptor structural proteins. SYNE1–4 (nuclear envelope linkers) were also excluded. Extracellular domain (ECD) length was computed from UniProt Topological domain annotations.

4.10 Co-expression baseline

Cross-cell-type Pearson baseline. Co-expression baseline computed on pseudo-bulk expression data from Human Protein Atlas v24 single-cell type RNA expression (proteinatlas.org). Normalized counts per million for 20,151 genes across 154 cell types were pivoted into a gene × cell-type matrix; for each gene pair, Pearson correlation was computed across the 154 cell-type dimensions. This cross-cell-type correlation is the appropriate comparator: both the embedding and the baseline answer the same question—which genes covary across biological contexts—making any difference attributable to the learned representation rather than the data modality. Within-cell-type single-cell correlation addresses a fundamentally different question (context-specific co-variation, dominated by dropout noise at the resolution of individual cells) and is not comparable to static representations that aggregate across contexts.

Within-cell-type Pearson baseline (B cells). Within-cell-type baseline: 10x Genomics PBMC 10k v3 dataset (10,516 cells after QC filtering), library-size normalized (10,000 counts per cell) followed by log1p transformation across all genes. B cells identified by standard scanpy workflow (2,000 highly variable genes, PCA, Leiden clustering at resolution 0.8), selecting clusters with mean log-normalized CD79A > 1.0 and MS4A1 > 1.0. Gene–gene Pearson correlations computed on the full log-normalized expression matrix (∼20,000 genes) across identified B cells. Repeated at three purity levels: N = 1,723 (all B-containing clusters), N = 1,650 (clean B clusters excluding myeloid-contaminated), N = 910 (purest B cluster only). The embedding was not fitted to B cell data or the PBMC dataset.

4.11 Receptor classification and structural analysis

Receptor classes were assigned from UniProt human reviewed proteins (TSV, ∼20,400 entries) using the following hierarchy: (1) GPI-anchored: UniProt keyword KW-0336 ("GPI-anchor"); (2) RTK: UniProt EC 2.7.10.1 (receptor tyrosine kinase activity); (3) single-pass non-RTK: transmembrane count = 1 and not in RTK list; (4) GPCR: transmembrane count = 7 or UniProt "Protein families" contains "G-protein coupled receptor"; (5) multi-pass non-GPCR (ion channels): DGIdb "ION CHANNEL" category, supplemented by transmembrane count > 1 and not GPCR; (6) secreted/soluble: Keywords contains "Secreted" and transmembrane count = 0. Extracellular fraction computed as extracellular residues divided by total residues from UniProt Topological domain annotations. Jonckheere–Terpstra ordered trend test across five receptor classes.

Heteromer pair test. To test whether LD1 resolves topology across the membrane within individual signaling complexes, we curated 18 heteromeric pairs from five categories: (i) 7 coarse receptor→cytoplasmic kinase pairs (TCR, BCR co-receptors, GDNF, CNTF), (ii) 5 ITAM receptor→SYK pairs (FcγR, FcεRI, Dectin-1, TREM2, GPVI), (iii) 4 cytokine receptor→JAK2 pairs (EPOR, MPL, GHR, LEPR), (iv) 2 dedicated interferon receptor pairs (IFNAR1→IFNAR2, IFNGR1→IFNGR2), and (v) 5 promiscuous shared chain pairs as expected negative controls. Significance assessed by one-sided Wilcoxon signed-rank test.

4.12 Cosine neighborhood and chain analysis

For each query gene, all 36,591 genes were ranked by embedding cosine similarity. Multi-hop chain analysis: for each validated receptor→kinase pair, the kinase's cosine neighborhood was inspected for known downstream partners. Two-hop biochemical distance: for consecutive pairs in known signaling cascades (e.g., SYK→BLNK→BTK), the rank of each downstream partner was reported among all genes. Skip connections (e.g., SYK→BTK directly) test whether rank degrades when an intermediate is bypassed.

ADC internalization analysis. For each ADC target, ranks of endocytic machinery genes (CLTC, AP2M1, DNM2, EPS15, CAV1, CAVIN1, RAB5A, EEA1, RAB7A, LAMP1, LAMP2, CTSD, RAB11A, RAB4A) were computed by embedding cosine similarity. Multi-hop bridges: two-hop paths from receptor through intermediate neighbors to clathrin (CLTC) or caveolar (CAV1) machinery were enumerated. Positive control: MRC1/CD206 (professional endocytic receptor). Negative controls: CD20 (ADCC mechanism), CD30 (bystander/extracellular cleavage).

Tdark/Tbio screening. Tdark and Tbio genes (IDG classification) were filtered to the LD1 range occupied by approved ADC targets (−0.5 to −2.5). UniProt structural annotations (single-pass transmembrane, extracellular domain size, cytoplasmic tail length) were used to identify candidates with ADC-compatible geometry. Cosine neighborhoods were manually inspected for individual candidates. Therapeutic status verified against ClinicalTrials.gov, PubMed, and company pipeline disclosures.

References

10x Genomics. Chromium single cell 3' reagent kits v3 user guide. *10x Genomics Technical Documentation*, 2019. Reports 30–32% transcript capture efficiency for v3 chemistry.

10x Genomics. Resolving biology to the level of single cells. *10x Genomics Technical Note*, 2020. Overview of droplet-based single-cell methods.

10x Genomics. GEM-X technology for single cell gene expression. *10x Genomics Technical Note*, 2024.

S. Adduri et al. STATE: a foundation model for therapeutic response prediction. *bioRxiv*, 2025. Retrained by Wang et al. 2026 on matching data.

Arc Institute. Virtual cell challenge 2025: Wrap-up. https://arcinstitute.org/news/virtual-cell-challenge-2025-wrap-up, 2025.

R. D. Baker. New order-statistics-based ranking models and faster computation of outcome probabilities. *IMA Journal of Management Mathematics*, 31(1):33–48, 2020.

R. D. Baker and P. A. Scarf. Modifying Bradley-Terry and other ranking models to allow ties. *IMA Journal of Management Mathematics*, 32(4):451–463, 2021.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3: 993–1022, 2003.

G. A. Bouland et al. The case for binarization of single-cell RNA-seq data. *Genome Biology*, 24:99, 2023.

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

C. Bravo González-Blas, S. De Winter, G. Hulselmans, et al. SCENIC+: single-cell multiomic inference of enhancers and gene regulatory networks. *Nature Methods*, 20:1355–1367, 2023. doi: 10.1038/s41592-023-01938-4.

G. Brixi, J. Durairaj, et al. Genome modeling and design across all of life with Evo 2. *Nature*, 2025.

C. Bunne, Y. Roohani, T. Engber, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 2024.

K. Choi et al. Bayesian model selection reveals biological origins of zero inflation in single-cell transcriptomics. *Genome Biology*, 21:1–16, 2020.

A. Chowdhery et al. PaLM: Scaling language modeling with pathways. *arXiv:2204.02311*, 2022.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

G. M. Cooper. *The Cell: A Molecular Approach*. Sinauer Associates, 2nd edition, 2000.

H. Cui, C. Wang, H. Maan, K. Pang, F. Luo, N. Duan, and B. Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21:1470–1480, 2024.

J. E. Darnell, I. M. Kerr, and G. R. Stark. Jak-STAT pathways and transcriptional activation in response to IFNs and other extracellular signaling proteins. *Science*, 264(5164):1415–1421, 1994.

R. R. Davidson. On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970.

M. Dong, A. Adduri, D. Gautam, C. Carpenter, R. Shah, C. Ricci-Tam, Y. Kluger, D. P. Burke, and Y. H. Roohani. Stack: In-context learning of single-cell biology. *bioRxiv*, 2026. doi: 10.64898/2026.01.09.698608.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

A. Eilers, D. Georgellis, B. Klose, C. Schindler, A. Ziemiecki, A. G. Harpur, A. F. Wilks, and T. Decker. Differentiation-regulated serine phosphorylation of STAT1 promotes GAF activation in macrophages. *Molecular and Cellular Biology*, 15(7):3579–3586, 1995.

A. E. Elz, D. Gratz, A. Long, D. Sowerby, A. Hadadianpour, and E. W. Newell. Evaluating the practical aspects and performance of commercial single-cell RNA sequencing technologies. *NAR Genomics and Bioinformatics*, 2025. doi: 10.1093/nargab/lqaf215.

J. Engels, E. J. Michaud, I. Liao, W. Gurnee, and M. Tegmark. Not all language model features are one-dimensionally linear. *arXiv preprint arXiv:2405.14860*, 2025.

M. Fang and L. Pachter. Extrinsic biological stochasticity and technical noise normalization of single-cell RNA sequencing data. *bioRxiv*, 2025. doi: 10.1101/2025.05.11.653373.

S. Gandhi, F. Javadi, V. Svensson, U. Khan, M. G. Jones, J. Yu, D. Merico, H. Goodarzi, and N. Alidoust. Tahoe-x1: Scaling perturbation-trained single-cell foundation models to 3 billion parameters. *bioRxiv*, 2025. doi: 10.1101/2025.10.23.683759.

M. B. Gerstein et al. Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489: 91–100, 2012.

A. Green. Through a glass darkly: Mechanistic interpretability as the bridge to end-to-end biology. 2024. https://www.markov.bio/research/mech-interp-path-to-e2e-biology.

A. Green. Markov blankets and mech interp—why an esoteric philosophy of statistical physics is the key to unlocking the virtual cell, drug discovery, and all of biology, 2025. https://x.com/adamlewisgreen/status/1881515348127797644.

W. Gurnee and M. Tegmark. Language models represent space and time. *ICLR*, 2024.

C. Hafemeister and R. Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20:296, 2019.

Z. S. Harris. Distributional structure. *Word*, 10(2–3):146–162, 1954.

T. Hayes et al. ESM-C: Language models for protein design. *EvolutionaryScale Technical Report*, 2024.

D. A. Henderson. Modelling and analysis of rank ordered data with ties via a generalized Plackett-Luce model. *Bayesian Analysis*, 2022. arXiv:2212.08543.

J. Hewitt and C. D. Manning. A structural probe for finding syntax in word representations. In *NAACL*, 2019.

M. Huh, B. Cheung, T. Wang, and P. Isola. The platonic representation hypothesis. *ICML*, 2024.

A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5): 1187–1201, 2015.

J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

D. Levine et al. Cell2Sentence: Teaching large language models the language of biology. *bioRxiv*, 2023.

X. Li, S. Leung, S. Qureshi, J. E. Darnell, and G. R. Stark. Formation of STAT1-STAT2 heterodimers and their role in the activation of IRF-1 gene transcription by interferon-α. *Journal of Biological Chemistry*, 271(10): 5790–5794, 1996.

J. Lienen and E. Hüllermeier. Monocular depth estimation via listwise ranking using the Plackett-Luce model. In *CVPR*, pages 14595–14604, 2021.

Z. Lin, H. Akin, R. Rao, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15:1053–1058, 2018.

R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

R. Milo and R. Phillips. Cell biology by the numbers. 2015.

C. R. Oyagawa and N. L. Grimsey. Cannabinoid receptor CB1 and CB2 interacting proteins: Techniques, progress and perspectives. In *Biomolecular Interactions Part A*, volume 166, pages 83–132. Academic Press, 2021.

P. Pearce et al. Finding the tree of life in Evo 2. 2025. Goodfire Research. https://www.goodfire.ai/research/phylogeny-manifold.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

P. Qiu. Embracing the dropouts in single-cell RNA-seq analysis. *Nature Communications*, 11:1169, 2020.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI Technical Report*, 2018.

R. Rafailov, A. Sharma, E. Mitchell, et al. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.

D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications*, 9:284, 2018.

A. Sarkar and M. Stephens. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature Genetics*, 53:770–777, 2021.

A. Sharaf, L. Mensching, C. Keller, S. Rading, M. Scheffold, L. Palkowitsch, et al. Systematic affinity purification coupled to mass spectrometry identified p62 as part of the cannabinoid receptor CB2 interactome. *Frontiers in Molecular Neuroscience*, 12:224, 2019.

S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946.

J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. RoFormer: Enhanced transformer with rotary position embedding. In *arXiv:2104.09864*, 2021.

V. Svensson. Droplet scRNA-seq is not zero-inflated. *Nature Biotechnology*, 38:147–150, 2020.

V. Svensson, K. N. Natarajan, L.-H. Ly, R. J. Miragaia, C. Laber, Q. Deng, V. Bystry, and S. A. Teichmann. Power analysis of single-cell RNA-sequencing experiments. *Nature Methods*, 14:381–387, 2017.

I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. *ACL*, 2019.

C. V. Theodoris et al. Transfer learning enables predictions in network biology. *Nature*, 618:616–624, 2023.

P. J. Thul et al. A subcellular map of the human proteome. *Science*, 356(6340), 2017.

L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.

R. B. Walter, B. W. Raden, D. M. Kamikura, J. A. Cooper, and I. D. Bernstein. Influence of CD33 expression levels and ITIM-dependent internalization on gemtuzumab ozogamicin–induced cytotoxicity. *Blood*, 105(3): 1295–1302, 2005.

R. B. Walter, B. W. Raden, R. Zeng, P. Häusermann, I. D. Bernstein, and J. A. Cooper. ITIM-dependent endocytosis of CD33-related Siglecs: role of intracellular domain, tyrosine phosphorylation, and the tyrosine phosphatases, Shp1 and Shp2. *Journal of Leukocyte Biology*, 83(1):200–211, 2008.

C. Wang, M. Karimzadeh, N. G. Ravindra, L. R. Bounds, N. Alerasool, A. C. Huang, S. Ma, D. R. Gulbranson, H. Cui, et al. X-Cell: Scaling causal perturbation prediction across diverse cellular contexts via diffusion language models. *bioRxiv*, 2026. Table B7 used for direct comparison.

D. I. Warton. Why you cannot transform your way out of trouble for small counts. *Biometrics*, 74(1):362–368, 2018.

G. Yang, E. J. Hu, I. Babuschkin, et al. Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer. In *NeurIPS*, 2022.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

N. D. Youngblut et al. scBaseCount: A uniformly processed repository of single-cell RNA-seq count data. *bioRxiv*, 2025. Arc Institute.

J. Yu, X. Cui, G. Shao, J. Li, Q. Li, Q. Liang, N. Li, and X. Li. LILRB3 inhibition reverses immunosuppression in glioma: a nanoparticle-based therapeutic strategy. *Journal of Nanobiotechnology*, 2026.

Q. Zhuang, Y. Liu, H. Wang, Z. Lin, L. Sun, Y. Liu, Y. Lyu, L. Chen, H. Yang, and Y. Mao. LILRB3 suppresses immunity in glioma and is associated with poor prognosis. *Clinical and Translational Medicine*, 13(10):e1396, 2023.

C. Ziegenhain, B. Vieth, S. Parekh, B. Reinius, A. Guillaumet-Adkins, M. Smets, H. Leonhardt, H. Heyn, I. Hellmann, and W. Enard. Comparative analysis of single-cell RNA sequencing methods. *Molecular Cell*, 65(4): 631–643, 2017.

Supplementary Material

A. Pretraining objective ablation and loss function analysis — mathematical comparison of four objectives, NB collapse, full 48-metric evaluation, cooldown dynamics (§A)

B. The data generating process: from chemistry to count distributions — sequencing pipeline, dropout resolution, analytical derivation of Geometric count histograms (§B)

C. From Thurstone to GPL: intellectual history and model selection — ranking model landscape, elicited vs induced rankings, loss function evolution (§C)

D. Evaluation pipeline and metric interpretation — masked prediction protocol, Spearman tie structure, objective-agnostic evaluation (§D)

E. Perturbation prediction: extended results and scaling — full metric comparison with published baselines, scaling across three model sizes (§E)

F. Gene regulatory network inference: extended analysis — ENCODE benchmark, per-TF AUROC, repressor geometry, STAT2 case study, motif enrichment (§F)

G. Subcellular localization: extended analysis — three-modality comparison (protein, DNA, RNA), confusion matrix, per-compartment accuracy (§G)

A Pretraining objective ablation and loss function analysis

A.1 All Four Loss Functions — Mathematical Summary

We compare four pretraining objectives that share the same transformer backbone and output head (plus NB as a fifth, tested separately; see §A.7). They differ only in how the output logits are converted to a training signal. (GPL, GC, and NB additionally share the same UMI depth conditioning; see §A.3.)

Notation. The model's output head produces a single vector of per-gene logits. We write ϕ_g for gene g. Higher ϕ_g means more expressed — the intuitive direction. All distributional quantities are derived from ϕ_g:

Symbol	Name	Definition	Range	Direction	Meaning
ϕ_g	Raw logit	Model output	$(-\infty, +\infty)$	Higher = **more** expressed	The only quantity the model produces
λ_g	Geometric rate	e^{ϕ_g}	$(0, \infty)$	Higher = **more** expressed	$\mathbb{E}[k] = \lambda_g$ in the Geometric
θ_g	Geometric stopping prob.	$\sigma(-\phi_g) = \frac{1}{1+e^{\phi_g}}$	$(0, 1)$	Higher = **less** expressed	Prob. of "stopping" at each count step
$\log(1-\theta_g)$	Log continuation prob.	$\text{logsigmoid}(\phi_g)$	$(-\infty, 0)$	—	Used in GC/NB loss (count term)
$\log \theta_g$	Log stopping prob.	$\text{logsigmoid}(-\phi_g)$	$(-\infty, 0)$	—	Used in GC/NB loss (baseline term)
r_g	NB dispersion	Learned parameter	$(0, \infty)$	—	NB only; $r_g = 1$ recovers Geometric
s_0	Depth scaling factor	Learned parameter	$(0, \infty)$	—	Scales the UMI depth shift (§A.3)
α_z, α_r	Regularization weights	Fixed hyperparameters	—	—	z-loss and NB dispersion penalty

The Geometric distribution. We use the "failures before first success" convention (support on $k \in \{0, 1, 2, \ldots\}$, where $k = 0$ means the gene was not detected). Each gene independently flips a coin with stopping probability θ_g. The count k_g is how many times it continues before stopping, giving $P(k) = \theta \cdot (1-\theta)^k$ — one "stop" (θ) and k "continues" ($1-\theta$ each). The expected count is $\mathbb{E}[k] = (1-\theta)/\theta$, which is λ_g.

λ and θ are inversely related: $\theta = 1/(\lambda + 1)$. More counting means less stopping — a gene producing lots of mRNA (high λ) has a low probability of stopping at each molecule (low θ), while a barely-expressed gene (low λ) stops almost immediately (high θ). We call λ_g the "rate parameter" by analogy with the Poisson, where λ is also the mean.

The central identity: $\phi_g = \log(\lambda_g)$. In the Geometric distribution parameterized by ϕ_g, the expected count is $\lambda_g = e^{\phi_g}$. So ϕ and λ are the same quantity in different spaces — ϕ in log space (where the model works), λ in count space. Under GC, the loss calibrates the magnitudes, so e^{ϕ_g} is a meaningful count prediction. Under GPL, only the ordering is trained, so ϕ values are ordinally correct but not calibrated as expected counts (§A.1, "How to interpret the logits"). The identity $\phi_g = \log(\lambda_g)$ is a property of the parameterization, not a claim about calibration. This is also, by definition, the log-odds of the Geometric distribution: $\phi_g = \log\left(\frac{1-\theta_g}{\theta_g}\right)$ — the log-odds of continuing versus stopping. The word "logit" itself was coined as a contraction of "log-odds" (Berkson 1944), so the name and the math coincide: our logits are literally log-odds of the Geometric coin flip.

Some concrete values: $\phi = 0 \Rightarrow \lambda = 1, \theta = 0.5$, expected count 1. $\phi = 3 \Rightarrow \lambda \approx 20, \theta \approx 0.05$, expected count 20. $\phi = -3 \Rightarrow \lambda \approx 0.05, \theta \approx 0.95$, expected count near zero.

The key implementation point: the log-space quantities $\log \theta_g$ and $\log(1-\theta_g)$ are computed directly from ϕ_g via `logsigmoid`, without ever materializing θ_g or λ_g as intermediate floating-point values. The entire path from raw logit to log-probability is a single numerically stable function call. This is why our implementation requires no epsilon guards (§A.4).

We reserve λ exclusively for the Geometric rate parameter throughout this supplement. Regularization coefficients use α.

GPL (Geometric Plackett–Luce). A ranking likelihood over gene expression orderings, using Geometric latents to handle ties (Henderson 2022):

$$\mathcal{L}_{\text{GPL}} = -\log P_{\text{GPL}}(\text{observed ranking} \mid \boldsymbol{\theta})$$

The likelihood decomposes into a binary detection term and a ranking term among expressed genes (§A.2). No per-gene distributional parameters beyond the shared output head.

Geometric Count (GC). The Geometric distribution log-likelihood, which is the NB special case at $r = 1$:

$$\mathcal{L}_{\text{GC}} = -\sum_g \left[\text{logsigmoid}(-\phi_g) + k_g \cdot \text{logsigmoid}(\phi_g)\right]$$

where k_g is the observed count for gene g. The first term is the log-probability of "stopping" (the Geometric baseline); the second is the log-probability of "continuing" k_g times (the count). No per-gene distributional parameters.

GPL and GC are the cleanest possible comparison. Both produce per-gene logits ϕ_g interpreted as Geometric parameters via $\theta_g = \sigma(-\phi_g)$. They share the same output head, the same depth conditioning, and the same regularization. They differ *only* in the loss function: GC minimizes the negative log-likelihood of the observed *counts* under the Geometric distribution; GPL minimizes the negative log-likelihood of the observed *ranking* under the Geometric Plackett–Luce model. Every difference in downstream transfer quality between GPL and GC is attributable to this single choice — counts vs rankings.

Negative Binomial (NB). Generalizes GC with a learned per-gene dispersion r_g:

$$\mathcal{L}_{\text{NB}} = -\sum_g \left[r_g \cdot \text{logsigmoid}(-\phi_g) + k_g \cdot \text{logsigmoid}(\phi_g) + \log\Gamma(k_g + r_g) - \log\Gamma(r_g) - \log\Gamma(k_g + 1)\right]$$

One learned parameter per gene (r_g), initialized at $r_g = 1$ (the Geometric prior). We test both unregularized ($\alpha_r = 0$) and regularized ($\alpha_r = 0.01$, L2 penalty on $r_g - 1$).

PROPORTIONAL (PROP). Cross-entropy against normalized expression proportions:

$$\mathcal{L}_{\text{PROP}} = -\sum_g p_g \cdot \log\ \text{softmax}(\phi)_g$$

where $p_g = k_g / \sum_j k_j$ are the observed UMI proportions. The softmax operates over all ~36,000 genes. No per-gene distributional parameters. Higher $\phi_g \rightarrow$ higher softmax probability $\rightarrow$ higher predicted expression — the intuitive direction.

Shared infrastructure. All four objectives use:

- The same transformer backbone and per-gene output head
- The same z-loss regularization: $\alpha_z \cdot \|\text{logsumexp}(\phi)\|^2$

GPL, GC, and NB additionally use the UMI depth shift: $\phi'_g = \phi_g + \log(\mu_{\text{tgt}}) \cdot s_0$ where $\mu_{\text{tgt}} = \text{UMI}/V$ (§A.3). PROP does not need it — softmax is invariant to adding a constant to all logits, so the shift has no effect. Proportions already normalize out depth by construction, which is the compositional approach scVI also takes.

This is worth emphasizing: all five models produce the same output — a vector of per-gene logits from the same linear head. The code paths are identical up to the loss computation:

```
# Identical for ALL five objectives:
phi = model.output_head(hidden_states)          # [batch, cells, genes]

# Depth shift (GPL, GC, NB only -- no effect on PROP due to softmax invariance):
if objective in ("GPL", "GC", "NB"):
    phi = depth_shift(phi, umi, V, s0)

# Only this part differs:
if objective == "GPL":       loss = gpl_ranking_loss(phi, counts)
elif objective == "GC":      loss = geometric_count_nll(phi, counts)
elif objective == "NB":      loss = nb_count_nll(phi, counts, r)
elif objective == "PROP":    loss = cross_entropy(softmax(phi), proportions)
```

At evaluation time, we interpret the logits identically for all objectives: `argsort(logits)` gives the predicted ranking, and the Spearman correlation against ground truth is computed the same way regardless of how the logits were trained. The ablation is as clean as it can be — same architecture, same data, same evaluation, different loss function.

What's under the hood: the distributions. It may help to see what generative story each objective tells about a gene with logit ϕ_g:

Objective	Generative story	What ϕ_g controls
GPL	Gene g draws a random count from Geometric($\sigma(-\phi_g)$). The *ranking* of these draws across all genes is the observed expression ordering.	Detection probability and relative rank — simultaneously
GC	Gene g's count is drawn from Geometric($\sigma(-\phi_g)$). The model tries to predict the *exact count*.	Expected count: $\mathbb{E}[k] = \lambda_g = e^{\phi_g}$
NB	Same as GC but with a per-gene learned "spread" parameter r_g that allows the variance to differ from the mean.	Expected count (with more flexibility to fit variance)
PROP	The fraction of this cell's mRNA belonging to gene g is softmax$(\phi)_g$. The model matches observed proportions.	Relative share of total expression — coupled to all other genes

GPL and GC tell the *same generative story* (Geometric draws) but ask different questions about it: GC asks "what was the count?", GPL asks "what was the rank?" This is why the GPL-vs-GC comparison is so clean — the underlying probability model is identical, and the entire difference in transfer quality comes from whether the loss function cares about count magnitudes or ordinal relationships.

How to interpret the logits under each objective. Under GC, the logits are calibrated as log expected counts — the loss explicitly penalizes count prediction errors, so e^{ϕ_g} is a meaningful estimate of $\mathbb{E}[k_g]$. Under GPL, the logits are partially calibrated. The binary detection term (§A.2) gives magnitudes real meaning *at the detection boundary*: $\phi_g > 0$ means "more likely detected than not," and how far above zero controls the model's confidence. This is why the depth shift (§A.3) matters for GPL — it moves the detection boundary to the right place for each cell's depth. But among expressed genes, the ranking term constrains only the *ordering* of logits, not their spacing. GPL is free to learn whatever magnitudes make the gradients flow well, without committing to correct count predictions. This freedom to ignore count magnitudes among expressed genes is precisely what prevents overfitting to technical variance — and it is why GPL outperforms GC despite sharing the same parameterization.

The regularized NB total loss is:

$$\mathcal{L} = \frac{1}{N}\left(\alpha_z \cdot \|\text{logsumexp}(\phi)\|^2 + \text{NLL}_{\text{NB}} + \alpha_r \cdot \frac{1}{G}\sum_g (r_g - 1)^2\right)$$

Loss	Target	Per-gene params	Key property
GPL	Ranking	0	Handles tied zeros as a single bucket
GC	Counts	0	NB at $r = 1$; fits count magnitudes
PROP	Proportions (all genes)	0	Softmax couples all gene predictions
NB	Counts + dispersion	1 (r_g)	Richer count model; more flexibility to overfit

A.2 The GPL Decomposition — Binary Detection + Ranking

The GPL loss looks complex in notation, but it rests on two elegant ideas, each simple on its own.

Idea 1: Plackett–Luce sequential choice (Luce 1959, Plackett 1975). Imagine ranking items by picking them off one at a time from a pool. At each stage, you choose an item with probability proportional to its "strength" λ. Once chosen, it's removed from the pool. The probability of a complete ranking $y_1 > y_2 > \cdots > y_K$ is:

$$P(\text{ranking} \mid \boldsymbol{\lambda}) = \prod_{j=1}^{K} \frac{\lambda_{y_j}}{\sum_{\ell \geq j} \lambda_{y_\ell}}$$

This has a powerful property: Independence of Irrelevant Alternatives (IIA). The relative probability of A being ranked above B doesn't depend on what other items are in the pool. This tractability — the likelihood factoring into a product of simple ratios — is not a coincidence. It depends on the exponential distribution being *memoryless*: given that an item hasn't been chosen yet, its future probability of being chosen doesn't depend on how long it's been waiting. Memorylessness is what makes the "remove and renormalize" step exact rather than approximate — after removing the chosen item, the remaining items' relative probabilities are unchanged. Baker (2020) and Baker and Scarf (2021) proved that the exponential and Geometric are the *only* continuous and discrete distributions (respectively) that yield closed-form ranking likelihoods — a uniqueness that follows from the fact that they are the only memoryless distributions in their respective classes. This is why PL and GPL are uniquely tractable among ranking models. But PL assigns zero probability to ties, because the underlying latent variables are continuous (exponential).

Idea 2: Geometric latents → ties become natural (Henderson 2022). Replace the continuous exponential latent variables with their discrete counterparts — Geometric random variables. Each gene independently flips a biased coin (stopping probability θ_g) repeatedly; the count is how many flips until it stops. Because the Geometric is discrete, two genes can stop on the same flip — a tie. This one substitution (exponential → Geometric) gives us a proper generative model for rankings with ties, while preserving PL's tractability. Henderson (2022) showed that PL is recovered as a limiting special case when $\theta \to 0$ for all items (corresponding to very large counts where ties become vanishingly rare).

For gene expression, ties are not rare — they're dominant. With ~30,000 genes at count zero and most detected genes at counts 1–3, ties are the *typical* case. A ranking model that cannot handle ties is useless for this data. This is why we need GPL rather than PL.

The GPL decomposition. The GPL log-likelihood over a cell with V genes, of which S are expressed (count > 0), decomposes into two terms:

$$\log P(\text{data} \mid \boldsymbol{\theta}) = \underbrace{\sum_{i \in \text{expr}} \log(1-\theta_i) + \sum_{i \in \text{zeros}} \log \theta_i}_{\text{binary: which genes are detected?}} + \underbrace{\log P_{\text{GPL}}(\text{ranking among expressed genes})}_{\text{ranking: in what order?}}$$

where $\theta_g = \sigma(-\phi_g)$ is the Geometric stopping probability and $\lambda_g = e^{\phi_g}$ is the Geometric rate parameter. In the assumed Geometric distribution, λ_g is the expected count; under GPL, only the *ordering* of λ values is trained, not their magnitudes (see "How to interpret the logits" in §A.1). Higher ϕ_g means higher λ_g, lower θ_g — more expressed.

In log-space, the binary term is computed directly from ϕ_g without materializing θ_g: $\log(1-\theta_g) = \text{logsigmoid}(\phi_g)$ for expressed genes, $\log \theta_g = \text{logsigmoid}(-\phi_g)$ for zeros.

The same parameter does double duty. Each gene's ϕ_g controls both its detection probability (the binary term) and its relative ranking among expressed genes (the ranking term). This is not a design choice — it is a mathematical consequence of the Geometric distribution's properties. To see this concretely:

ϕ_g	$\lambda_g = e^{\phi_g}$	θ_g (stop prob)	P(zero)	P(detected)	Interpretation
4.6	100	0.01	0.01	0.99	Highly expressed — almost certainly detected
2.3	10	0.09	0.09	0.91	Expressed
0	1	0.50	0.50	0.50	Borderline — coin flip between detected and zero
−2.3	0.1	0.91	0.91	0.09	Likely zero
−4.6	0.01	0.99	0.99	0.01	Almost certainly zero

The $\phi = 0$ boundary is where the model is maximally uncertain about detection. The depth shift (§A.3) moves this boundary to the right place for each cell's sequencing depth. Among detected genes, the same ϕ values determine the ranking: a gene at $\phi = 4.6$ ranks above a gene at $\phi = 2.3$, purely from the ordering of their logits.

Handling tied counts. For N items tied at the same observed count, any of $N!$ orderings is equally likely under the model. The irreducible entropy $\log(N!)$ is subtracted from the loss — no model can do better than uniform over tied items. For the zero bucket ($N \approx 30{,}000$), this entropy is large but constant, contributing no gradient signal. GPL handles ties automatically through the group factor (Henderson 2022), avoiding the heuristic tie-breaking required by earlier ranking objectives (§A.14, loss function history).

Computational efficiency. The binary term requires only independent per-gene evaluations. The ranking term operates over the $S \ll V$ expressed genes only — the ~30,000 zeros are handled entirely by the binary term.

The detection boundary is depth-adaptive. The UMI depth shift (§A.3) adjusts all logits before the binary term is evaluated. The model learns "is this gene expressed *given how deeply this cell was sequenced*" rather than applying a fixed detection threshold.

Why this decomposition matters — and why it's not two losses bolted together. In implementation, the binary and ranking terms appear as separate computations, which might suggest they are independent losses added together as a design choice. They are not. They are a factorization of a single GPL likelihood, just as a joint probability $P(A, B)$ factors into $P(A) \cdot P(B|A)$. The binary term computes "what's the probability that *exactly this set*

of genes escaped the zero bucket?" The ranking term computes "given they escaped, what's the probability of *this ordering*?" Together, they give the full bucket-order probability. When the bottom bucket contains ~30,000 zeros, this factorization is what makes GPL tractable: the binary term handles the massive zero bucket with independent per-gene evaluations, the ranking term handles only the ~5,000 expressed genes, and the tie entropy for the zero bucket is a constant that drops out of the gradient. Count-based objectives have no such decomposition — they must process every gene's count individually, spending equal effort on the 30,000 zeros and the 5,000 expressed genes.

Worked example. To see how GPL computes its loss, consider a toy cell with 6 genes and observed counts $[0, 0, 0, 3, 3, 7]$. The model outputs logits $\phi = [-2.1, -1.3, -0.8, 1.2, 1.8, 2.8]$. GPL interprets these as parameters of the Geometric distribution, with rates $\lambda = e^{\phi} \approx [0.12, 0.27, 0.45, 3.3, 6.0, 16.4]$, and asks: how probable is the observed detection pattern and ranking under these parameters?

Binary term — which genes are detected? Expressed genes contribute logsigmoid(ϕ); zeros contribute logsigmoid($-\phi$):

Gene	Count	ϕ_g	Contribution	Value	Interpretation
A	0	−2.1	logsigmoid(2.1)	−0.12	Small penalty: model confidently predicts "off"
B	0	−1.3	logsigmoid(1.3)	−0.24	Moderate penalty: less confident
C	0	−0.8	logsigmoid(0.8)	−0.37	Larger penalty: model is unsure about this gene
D	3	1.2	logsigmoid(1.2)	−0.26	Moderate penalty: model predicts "on" but not strongly
E	3	1.8	logsigmoid(1.8)	−0.15	Smaller penalty: more confident
F	7	2.8	logsigmoid(2.8)	−0.06	Small penalty: model confidently predicts "on"

Binary term total: −1.20.

Ranking term — the Bernoulli trial "race." The GPL ranking term has an elegant stage-wise structure (Henderson 2022). Imagine all genes still "in play" simultaneously flipping independent biased coins — gene g has probability $\theta_g = \sigma(-\phi_g)$ of heads. Each round, the genes that flip heads "stop" (drop out). Genes that stop in the same round are tied. The order in which genes drop out — from first to stop (fewest counts, least expressed) to last to stop (most counts, most expressed) — determines the ranking.

Each stage of the GPL likelihood asks: "given that at least one gene flips heads this round, what's the probability that *exactly these genes* did?" For a non-tie stage (one gene drops out alone):

$$\frac{\theta_g \cdot \prod_{\text{remaining}}(1-\theta_\ell)}{1-\prod_{\text{all in play}}(1-\theta_\ell)} = \frac{\text{this gene stops} \times \text{all others continue}}{\text{at least one stops}}$$

For a tie stage (gene drops out, and the next gene is also stopping this round):

$$\frac{\theta_g \cdot \left(1-\prod_{\text{remaining}}(1-\theta_\ell)\right)}{1-\prod_{\text{all in play}}(1-\theta_\ell)} = \frac{\text{this gene stops} \times \text{at least one more also stops}}{\text{at least one stops}}$$

The full GPL likelihood is the product of these stage factors across all positions. In our example: 3 zero-count genes drop out in a tied group (they all "stopped" in the same round), then genes D and E drop out in a tied group (same round), then gene F is the last one standing. Each transition contributes one factor to the likelihood. The beauty of the Geometric distribution is that these factors have closed forms involving only products of θ and $(1-\theta)$ values — no factorials, no combinatorial explosion, no summation over permutations.

Tie handling — the key insight. Genes D and E are both at count 3. The model assigns them different logits ($\phi_D = 1.2$, $\phi_E = 1.8$), meaning it "thinks" E is more expressed than D. But the data doesn't support distinguishing them — they have the same count. GPL handles this correctly: any ordering of D and E is equally likely, contributing irreducible entropy $\log(2!) = 0.69$. The model is not penalized for its ordering of D and E, nor rewarded for getting it "right." Similarly, the 3 zero-count genes contribute $\log(3!) = 1.79$ of irreducible entropy. This tie-equivariance — the property that the likelihood is invariant to reorderings within a tie group — is a mathematical property of the Geometric Plackett–Luce distribution (Henderson 2022), not a heuristic imposed on the loss.

For comparison, the GC loss on the same cell. It computes the Geometric log-likelihood for each gene individually: $\text{logsigmoid}(-\phi_g) + k_g \cdot \text{logsigmoid}(\phi_g)$. GC treats genes D and E differently even though they have the same count — D gets a worse loss than E because the model predicts $\lambda_D = 3.3$ vs $\lambda_E = 6.0$, and the true count is 3. GC penalizes E for "overestimating" gene E's count. GPL ignores this distinction entirely, because the data contains no information about which of these two genes is more expressed.

A.3 UMI Depth Conditioning

Every model that predicts gene expression must deal with sequencing depth. A cell sequenced to 50,000 UMI will show 10× higher counts across the board than the same cell sequenced to 5,000 UMI — not because the biology is different, but because the sequencer ran longer. If the model doesn't account for this, it wastes capacity encoding "this cell was deeply sequenced" into every gene's prediction, learning a technical confounder rather than biology.

The standard approach in the field (scVI, scGPT, Geneformer) is to handle depth through library size normalization, latent variables, or explicit size factors — machinery that adds architectural complexity and interacts with the rest of the model in ways that are hard to reason about. We do something simpler: a single additive shift in logit space.

The idea starts with a simple question: if a cell has been sequenced to U total UMI and has V genes, what would you expect each gene's count to be if all genes were expressed equally? The answer is $\mu_{\text{tgt}} = U/V$ — just the total counts divided evenly. A cell with 36,000 UMI across 36,000 genes would have $\mu_{\text{tgt}} = 1$ count per gene. A deeply sequenced cell with 360,000 UMI would have $\mu_{\text{tgt}} = 10$.

Now, recall that $\phi_g = \log(\lambda_g)$ where λ_g is the Geometric rate parameter. Under GC and NB, λ_g is the expected count and this identity is calibrated; the model is trained to get magnitudes right. The "baseline" logit for a uniformly-expressed gene at depth U is simply $\log(\mu_{\text{tgt}})$. If the model's logit for a gene equals this baseline, the gene is expressed at exactly the level depth alone would predict — nothing interesting. If the logit is above baseline, the gene is more expressed than depth would predict. If below, less expressed.

The shift adds this baseline to all logits, so the model's output becomes a residual relative to what depth alone would predict:

$$\phi'_g = \phi_g + \log(\mu_{\text{tgt}}) \cdot s_0$$

This is where the logit parameterization pays off most elegantly. Because $\phi_g = \log(\lambda_g)$, addition in logit space is multiplication in count space. When $s_0 = 1$, exponentiating both sides gives:

$$\lambda'_g = e^{\phi'_g} = e^{\phi_g} \cdot e^{\log(\mu_{\text{tgt}})} = \lambda_g \cdot \mu_{\text{tgt}}$$

The expected count gets scaled by the baseline count per gene. And the model's residual logit is:

$$\phi_g = \log\left(\frac{\lambda_g}{\mu_{\text{tgt}}}\right) = \log\left(\frac{\text{gene's Geometric rate}}{\text{depth-expected baseline}}\right)$$

That's a log fold change relative to the depth baseline. Under GC, where λ_g is a calibrated expected count, this

is literally a log fold change in the biological sense. Under GPL, the magnitudes aren't calibrated but the *ordering* of these residuals determines the ranking — genes above baseline are ranked higher than genes below.

The intuition is direct: deep cell → high μ_{tgt} → positive shift → all logits pushed up → the model "expects" higher counts everywhere, so it only needs to learn which genes deviate from that higher baseline. Shallow cell → low μ_{tgt} → negative shift → all logits pushed down → the model expects lower counts. In both cases, the model's residual logits encode *biology*, not *sequencing depth*.

This is the natural way to do depth correction for any model that outputs log-rate logits — addition in log space, multiplication in count space, residuals as log fold changes. The learned scaling factor s_0 allows the model to calibrate how aggressively it corrects, since the exact relationship between log-depth and logit shift might not be exactly 1:1 due to nonlinearities in the sequencing process. (The log fold change interpretation is exact when $s_0 = 1$. When $s_0 \neq 1$, the residual is $\log(\lambda_g / \mu_{\text{tgt}}^{s_0})$ — still a meaningful depth-corrected quantity, but scaled by a learned exponent rather than a pure fold change. In practice s_0 converges near 1, so the approximation is close.)

The implementation is four lines of code:

```
def depth_shift(phi, umi, vocab_size, scale):
    """Additive logit-space depth correction."""
    mu_tgt = umi / vocab_size             # expected count per gene under uniform
    expression
    shift = torch.log(mu_tgt) * scale     # log-rate baseline, learned scaling
    return phi + shift                    # shift all gene logits
```

That's it. No latent variables, no VAE, no ELBO, no separate library size encoder. The shift is differentiable, the scaling factor s_0 is learned end-to-end, and the same four lines work for any logit-parameterized count model — Geometric, NB, or anything else that outputs per-gene logits. We expect this to become the standard approach for depth conditioning in logit-parameterized models.

	Shallow cell (5k UMI)	Deep cell (50k UMI)
$\mu_{\text{tgt}} = U/V$	0.14	1.39
$\log(\mu_{\text{tgt}})$	−1.97	+0.33
GAPDH true count	∼5	∼50
GAPDH rank	top 100	top 100

Concrete example — GAPDH in two cells with identical biology, different depth: Without the shift, the model must output very different logits for GAPDH in these two cells despite identical biology. With the shift, the baseline absorbs the depth difference — the shallow cell gets a negative baseline shift (lower expected counts), the deep cell a positive one (higher expected counts), and GAPDH's residual logit encodes "well above baseline" in both cells.

Shared across GPL, GC, and NB. These three objectives use the same depth shift, ensuring that performance differences among them reflect the objective, not depth handling. (PROP does not use the shift — softmax is invariant to adding a constant to all logits, so the shift has no effect. Proportions normalize out depth by construction.)

This may seem surprising for GPL, which operates on rankings rather than counts. But the GPL binary term — "which genes are detected?" — is depth-dependent: a gene with 2 copies might be detected in a deeply sequenced cell and undetected in a shallow one. The shift adjusts the detection boundary so the model asks "is this gene expressed given this cell's depth?" rather than learning separate detection thresholds for every depth level. The ranking term among expressed genes is depth-invariant by construction (rankings don't change when you multiply all counts by a constant), so the shift affects only the binary term — exactly where it's needed. This is the formal

sense in which ordinal measurement (Stevens 1946) provides robustness: rankings are invariant to monotone transformations of the underlying counts, so any technical confounder that acts multiplicatively (depth, amplification bias, capture efficiency) or through any other monotone distortion is invisible to the ranking term. The binary term is not ordinal — it depends on whether a count crosses zero — so it requires depth correction. The GPL decomposition naturally separates the part that needs correction from the part that doesn't.

For NB, the shift is calibrated for $r = 1$ (Geometric prior); when $r \neq 1$, the NB mean is $r \cdot \mu_{\text{tgt}}$, creating a slight mismatch. We deliberately retain the Geometric-calibrated shift so that the only difference between GC and NB is the learned r_g. Using the same shift for all objectives ensures that performance differences reflect the objective, not depth handling.

The scaling factor s_0 is a single learned scalar, shared across all genes and updated via backpropagation during training. The model discovers the optimal scaling — effectively learning how aggressively to correct for depth — rather than requiring it to be set by hand. In practice, s_0 converges quickly and varies little across objectives.[1]

What the shift does not correct for. The shift removes only the first-order (linear, gene-independent) depth effect. It does not correct for per-gene depth sensitivity (Poisson noise $\propto \sqrt{k}$), heterogeneous dropout thresholds, or library preparation artifacts. Rankings are robust to these residual technical effects because they depend only on the *ordering* of expression values, not their magnitudes. This is one reason GPL outperforms even depth-corrected count objectives — it is less sensitive to the technical variation that the shift cannot remove.

A.4 Numerical Stability — Logit Parameterization vs (μ, θ)

Our NB implementation parameterizes the distribution in logit space, with all log-probability terms computed via `logsigmoid`:

```
r * F.logsigmoid(-phi) + k * F.logsigmoid(phi)
    + lgamma(k + r) - lgamma(r) - lgamma(k + 1)
```

This is algebraically identical to the standard NB log-probability but avoids the numerical issues inherent in the (μ, θ) parameterization used by scVI (Lopez et al. 2018) and adopted by most of the single-cell field. The scVI implementation, `log_nb_positive`, is the field-standard NB log-likelihood:

```
# scVI -- (mu, theta) parameterization
log_theta_mu_eps = log(theta + mu + eps)          # eps = 1e-8
res = (
    theta * (log(theta + eps) - log_theta_mu_eps)  # log(theta/(theta+mu))
    + x * (log(mu + eps) - log_theta_mu_eps)       # log(mu/(theta+mu))
    + lgamma(x + theta) - lgamma(theta) - lgamma(x + 1)
)
```

Both compute the same mathematical quantity. The difference is entirely in how boundary cases are handled during floating-point evaluation.

Why `logsigmoid` and `softplus` are special. In machine learning, we frequently need to compute $\log(\text{sigmoid}(x))$ — the log of a probability. The naive approach computes sigmoid first, then takes the log: but when x is a large negative number, $\text{sigmoid}(x)$ rounds to 0.0 in float32, and $\log(0.0)$ is negative infinity. The `logsigmoid` function avoids this entirely by never materializing the intermediate sigmoid value. It is implemented as $-\text{softplus}(-x)$, where $\text{softplus}(x) = \log(1 + e^x)$ itself uses a piecewise evaluation that avoids computing e^x when x is large (which

[1] The logit-additive structure accommodates natural extensions: per-chemistry scaling (s_{chem}) for different protocols, per-gene factors (s_g) for transcript-length effects, etc. We leave these for future work.

would overflow) or log of a tiny number when x is very negative. The result is a function that returns the mathematically correct value of $\log(\text{sigmoid}(x))$ for any real-valued input, with no clamping, no epsilon, and no loss of precision. This is not novel — it is standard numerical engineering in PyTorch. What is surprising is that the single-cell field's standard NB implementation does not use it.

Issue 1: $\log(\mu + \varepsilon)$ when $\mu \to 0$. For unexpressed genes in a given cell, the decoder outputs $\mu \approx 0$. Then $\log(0 + 10^{-8}) = -18.4$, a large negative value that can dominate the loss and produce steep gradients ($\partial/\partial\mu$ of $\log(\mu + \varepsilon)$ is $1/(\mu + \varepsilon)$, which diverges as $\mu \to 0$). In scRNA-seq, where most entries in the cell × gene matrix are zero, this boundary case is the *typical* case. Our implementation handles this naturally: for a barely-expressed gene with $\phi_g \ll 0$ (large negative logit, low expected count), the count term $\text{logsigmoid}(\phi_g) \approx \phi_g$ — exact, no epsilon. The gradient with respect to ϕ_g is $\sigma(\phi_g) \approx 0$, bounded and smooth. No epsilon required.

Issue 2: Catastrophic cancellation. The scVI expression computes $\log(\theta + \varepsilon) - \log(\theta + \mu + \varepsilon)$. When the inverse dispersion θ is large (a low-variance gene) and μ is small, both terms are nearly equal. For example, with $\theta = 50{,}000$ and $\mu = 0.001$: $\log(50{,}000) - \log(50{,}000.001) \approx -2\times10^{-8}$, computed by subtracting 10.82 from 10.82 in float32 — catastrophic loss of significant digits. Our implementation computes $\text{logsigmoid}(\phi) = -\text{softplus}(-\phi)$ as a single numerically stable operation. No subtraction of nearly-equal quantities.

Issue 3: The epsilon is a hidden hyperparameter. The value $\varepsilon = 10^{-8}$ is arbitrary. Too large and it biases the loss; too small and it fails to protect against numerical instability. Different epsilon values produce different gradient dynamics near distribution boundaries. Our formulation contains no epsilon anywhere in the NB computation. `logsigmoid` is numerically exact for all real-valued inputs because PyTorch's `softplus` implementation uses a piecewise evaluation: $\text{softplus}(x) \approx x$ for large positive x, $\text{softplus}(x) \approx e^x$ for large negative x, and $\log(1 + e^x)$ in the moderate range.

Context. These are not obscure edge cases. In scRNA-seq, most entries in the cell × gene matrix are zero and most genes have very low mean expression — the boundary cases described above dominate the typical training batch. The (μ, θ) parameterization is standard in the statistics literature and was a reasonable default choice when scVI was developed. The numerical issues documented here are well known to the scVI maintainers. A 2019 pull request (scvi-tools #663) proposed moving to log-space computation throughout, but was declined as requiring "major global backend changes." In 2023, a maintainer acknowledged that even the library size transformation (a separate component from the NB likelihood) uses `exp` where `softplus` would be more stable (scvi-tools #1903) — but this would only fix the library size overflow, not the three issues above, which live inside the NB log-likelihood itself. As of 2024, users continue to report NaN losses and overflow with large dispersion values (scvi-tools #2965). Fixing the NB likelihood would require reparameterizing from (μ, θ) to logit space end-to-end — the approach we take here. The logit parameterization eliminates all of these problems by construction rather than by patching.

Depth conditioning in logit space vs count space. scVI handles depth multiplicatively: $\mu_g = \ell \cdot \rho_g$. In log space this becomes $\log \mu_g = \log \ell + \log \rho_g$, requiring log of a library size that can span three orders of magnitude and log of a softmax output that can approach zero. Our depth correction is additive in logit space: $\phi'_g = \phi_g + \log(\mu_{\text{tgt}}) \cdot s_0$. No log of a quantity that could be zero or extremely large.

A.5 Independent Logits vs Softmax-Coupled Proportions

Beyond numerical stability, the parameterization choice has a deeper consequence for what the model is forced to learn. To understand why, it helps to understand why scVI made the choice it did — because it was genuinely well-reasoned for its original purpose.

Why scVI uses softmax. scRNA-seq is, at the point of measurement, a compositional process. The sequencer produces a fixed budget of reads; every read assigned to gene A is a read not assigned to gene B. This is not a modeling assumption — it is the physics of the instrument. scVI's softmax decoder respects this: by outputting proportions that sum to 1, it says "I will model what the instrument actually measures — relative abundances — and handle depth separately through library size." This is compositionally correct, consistent with the Aitchison tradition, and for scVI's intended use cases — batch correction, imputation, clustering on a single dataset — it works well. Those tasks mainly require preserving coarse cell-type structure, and the top few hundred genes carry that signal easily.

The problem emerges when you shift from inference on one dataset to pretraining representations across many datasets. The goals diverge in two ways.

First, pretraining cares about the long tail. Rare cell-type markers, low-abundance transcription factors, signaling molecules expressed at a handful of copies per cell — these are the genes that distinguish closely related cell states (naive vs. memory T cells, pre-malignant vs. normal epithelium, quiescent vs. activated stem cells). Under softmax, these genes have $\rho_g \approx 10^{-5}$ and receive gradient signal attenuated by five orders of magnitude relative to housekeeping genes (see below). For single-dataset clustering this is acceptable. For pretraining, where the representations must transfer to unpredictable downstream tasks — perturbation prediction, drug response, fine-resolution annotation, gene program discovery — those weak gradients mean the model never learns rich features for the genes that actually matter.

Second, pretraining encounters massive cross-dataset technical variation — different labs, protocols, platforms, species — and the downstream tasks that consume pretrained representations are diverse and unpredictable. The softmax couples all gene predictions through its denominator, so a batch effect that inflates housekeeping genes in one protocol propagates through the normalization to suppress every other gene's proportion. With independent logits, batch effects in one gene's prediction don't mechanically interfere with another's.

None of this makes scVI's choice wrong for scVI's purpose. The insight is that the design pressures for generative modeling and representation learning point in opposite directions: generative modeling rewards faithful reconstruction of the data as measured, including its compositional structure; representation learning rewards learning equally rich features for the 35,000th gene as for the 1st.

Gradient suppression in the softmax regime. The softmax Jacobian for gene g has diagonal entry $\partial\rho_g/\partial z_g = \rho_g(1-\rho_g)$. With $V \approx 36{,}000$ genes and typical expression distributions, most genes have $\rho_g \approx 1/V \approx 2.7\times10^{-5}$, giving a gradient scaling factor of $\sim10^{-5}$. For the top $\sim$10 genes in a cell where $\rho_g \approx 0.05$, the gradient factor is $\sim$0.048 — roughly 1,800× larger. The top genes dominate the gradient signal; the remaining $\sim$36,500 genes are nearly invisible to the optimizer.

The off-diagonal terms ($\partial\rho_g/\partial z_h = -\rho_g \cdot \rho_h$) provide small indirect gradients, but these are even smaller — products of two small numbers — and entangled: perturbing one gene's logit affects all others simultaneously, not independently.

With independent logits, no such suppression occurs. Each gene's gradient depends only on its own loss contribution. A gene at count 0 and a gene at count 5,000 receive gradient signals proportional to their own prediction errors, with no 10^{-5} attenuation.

Biological mismatch. Gene expression is not compositional. A cell can upregulate an entire pathway — dozens of genes simultaneously — without downregulating anything. It simply produces more total mRNA. The softmax constraint contradicts this biology. scVI's library size variable ℓ partially addresses total expression changes, but does not fix the gradient problem: the softmax Jacobian eigenvalues are determined by the proportions ρ_g, regardless of library size scaling applied afterward.

The implicit temperature problem. There is an additional practical issue with softmax over 36,000 genes. Real gene expression is extremely peaked — a handful of genes account for most of the mRNA, while most genes are near zero. To produce this peaked distribution through softmax, the logits must have large magnitude (effectively, softmax over a large vocabulary is implicitly very cold). This pushes the decoder's weight magnitudes up during training, which interacts badly with weight decay, initialization schemes, and learning rate — standard training difficulties that compound at scale. With independent logits, there is no pressure to inflate logit magnitudes simply to achieve the right distributional shape.

"Just add a temperature parameter." A natural response is: use softmax(ϕ/τ) with a learnable temperature τ to control the sharpness, avoiding the need for large logit magnitudes. This does not help. The gradient suppression is a property of the softmax *probabilities* ρ_g, not the logit scale. The Jacobian diagonal is $\rho_g(1-\rho_g)$ regardless of temperature — a gene with $\rho_g = 10^{-5}$ gets 10^{-5} gradient scaling whether $\tau = 0.1$ or $\tau = 10$. A sharper softmax (lower τ) makes the distribution more peaked, which makes the gradient suppression *worse* for non-top genes, not better. Temperature rescales the logits but does not change the fundamental problem that softmax over 36,000 genes with a peaked expression distribution sends negligible gradient to the vast majority of them.

Connection to PROP ablation. Our PROPORTIONAL objective is structurally analogous to scVI's decoder head: softmax over genes → normalized proportions → loss against a target distribution. (The upstream computation differs — scVI's decoder reconstructs from a 10-dimensional latent variable via a small hidden layer, while our model projects from up to 1536-dimensional transformer hidden states. The decoder head architecture is analogous; the full computation graph is not.) Our ablation shows PROP loses to GPL across all scales, providing empirical evidence that the softmax coupling — specifically at the decoder output — harms representation quality for pretraining.

A.6 PROPORTIONAL Loss — The Zero-Suppression Problem

The PROPORTIONAL loss treats gene expression as a probability distribution: each gene's count is divided by total UMI to get a proportion ($p_g = k_g / \sum_j k_j$), and the model minimizes cross-entropy between the true proportions and the predicted softmax distribution over all ~36,000 genes.

The softmax forces all genes to compete for probability mass. In each cell, ~30,000 genes have zero counts — their true proportion is 0. But the softmax denominator $\sum_g \exp(\phi_g)$ sums over all genes, and the model must push ϕ_g to large negative values for each zero-count gene individually to prevent it from claiming probability mass. This is ~30,000 per-gene suppressions per cell, at every training step.

These zero-count genes do receive gradient signal — for each, the gradient is $q_g - 0 = q_g$, pushing the predicted probability toward zero. But this signal is small (§A.5: suppressed by the softmax Jacobian) and carries no useful information about gene expression biology. The zero counts reflect a mixture of genuine biological silence and technical dropout that the model cannot distinguish at the per-cell level.

GPL handles this with a single computation: all undetected genes form one tied bucket. The binary term determines the bucket boundary; the ranking term handles ordering among expressed genes. One boundary determination replaces ~30,000 individual suppressions.

A.7 The NB Escalation — Adding Distributional Flexibility Hurts

The Negative Binomial is the count distribution that scVI uses. It is, by standard statistical measures, a better model of scRNA-seq count data than the Geometric. And it produces the worst representations we tested. This is the single most important result in the ablation: the best count model in the field is the worst pretraining objective.

Here is the escalation in detail.

Geometric Count (GC, $r = 1$). Overfits during mid-training at d=1024 and d=1536. Cooldown rescues performance at d=1024 but accelerates collapse at d=1536 (§A.11).

Negative Binomial, unregularized (r_g learned, $\alpha_r = 0$). Overfits *catastrophically*. PAP-289: Spearman correlation collapsed from 0.45 (peak) to 0.047, present-subset at chance (0.50), z-loss exploded from 186k to 330k. The per-gene dispersion parameters r_g absorbed all variance — the model achieved good NLL by fitting count noise rather than learning biological structure.

Negative Binomial, regularized (r_g learned, $\alpha_r = 0.01$). The L2 penalty on $(r_g - 1)$ anchors dispersion toward the Geometric prior. The dispersion parameters stabilize near $r \approx 1$ rather than diverging, and the catastrophic collapse is avoided — but Spearman metrics remain well below GPL, multiple 0.1's lower. The optimizer spends gradient budget on a tug-of-war between "fit the counts better via r_g" and "don't deviate from Geometric."

The insight. If count-modeling quality drove representation quality, NB should outperform GC. The opposite occurs. The extra distributional flexibility becomes a channel for memorizing technical variance (sequencing noise, amplification bias, batch effects) rather than learning transferable biology.

The regularized NB is the most informative data point: it sits exactly where theory predicts, worse than GC (which has no extra parameter to overfit) but more stable than unregularized NB. You cannot regularize your way out of a bad objective class — the regularizer fights the optimizer, and the resulting representations bear the scars of both.

Why count objectives collapse at scale. The Geometric count loss for a gene with logit ϕ_g and observed count k_g is $\text{logsigmoid}(-\phi_g) + k_g \cdot \text{logsigmoid}(\phi_g)$. The first term (stopping) and the second term (counting) pull in opposite directions. As the model correctly learns that a highly expressed gene should have large ϕ, the counting term improves ($\text{logsigmoid}(\phi) \to 0$) but the stopping term gets worse ($\text{logsigmoid}(-\phi) \to -\phi$). At large model scales, the model has enough capacity to push logits to extremes, making these opposing forces larger. The gradients from matching count magnitudes grow without bound as the model becomes more expressive — a recipe for instability.

NB makes this worse by giving the model an escape valve. The NB stopping term is $r_g \cdot \text{logsigmoid}(-\phi_g)$, multiplied by the dispersion parameter. The model discovers it can reduce total NLL by inflating r_g — which rebalances the stopping and counting terms — rather than by learning better rankings. The more capacity the model has, the more precisely it can exploit this shortcut. This is the mechanism behind the PAP-289 collapse: r_g absorbs variance, NLL improves, but the logits ϕ_g are no longer encoding biology.

GPL avoids the entire pathology. The ranking term doesn't care about magnitudes — the model can place ϕ_E above ϕ_D by 0.1 or by 100, same ranking probability. There is no incentive to push logits to extremes, no opposing gradient forces that grow with scale, and no dispersion parameter to absorb variance. The z-loss keeps logits bounded, the binary term calibrates the detection boundary, and the ranking term learns biology. Stable at every scale.

Scale note. The NB experiments (both unregularized PAP-289 and regularized) were conducted at d=1024 (1B parameters). GPL, GC, PROP, and MSE were compared at d=$\{512, 1024, 1536\}$. We did not run NB at d=1536 because the d=1024 results were already conclusive — catastrophic collapse for unregularized NB, and substantially worse than GC for regularized NB. Running NB at larger scale would, based on the mechanism described above (opposing gradient forces amplified by model capacity), produce worse results, not better.

The distributional flexibility hierarchy: The trend is monotonic: every additional degree of distributional freedom is another opportunity for the optimizer to memorize technical variance rather than learn biology.

Model	Per-gene params	Transfer quality
GPL (rankings)	0	Best
PROP (softmax proportions)	0	Worse (softmax coupling)
GC (Geometric, $r = 1$)	0	Worse (fits count magnitudes)
NB reg (r_g learned, L2→1)	1	Worse (regularizer fights optimizer)
NB unreg (r_g free)	1	Worst (catastrophic collapse)

Implications for scVI's ZINB. scVI uses Zero-Inflated Negative Binomial by default, adding a per-gene zero-inflation parameter π_g on top of the NB's r_g — two learned distributional parameters per gene. We did not test ZINB because our results predict the outcome: additional distributional flexibility should further degrade transfer quality. We note that scVI's context is different (single-dataset VAE inference rather than multi-dataset pretraining), and the ZINB parameterization may be appropriate for that setting. For pretraining across diverse tissues and sequencing protocols, our ablation suggests that distributional simplicity is a virtue.

A.8 Experimental Fairness

The obvious objection to everything above is: maybe the count objectives just needed more tuning. Maybe a better learning rate, a different initialization, a more careful regularization scheme would close the gap. We took this seriously, and we think the evidence rules it out.

Shared infrastructure. All objectives use the same transformer backbone, the same output head, the same optimizer (learning rate, schedule, warmup), the same UMI depth shift (§A.3), the same metadata conditioning, and the same z-loss regularization.

Numerical advantage for count-based objectives. Our GC and NB implementations use the logit parameterization with `logsigmoid` throughout (§A.4), making them *more* numerically stable than the standard implementations used by scVI and the broader field. If anything, our count-based objectives have an unfair advantage over their field-standard counterparts.

NB-specific provisions. Per-gene dispersion is initialized at the Geometric prior ($r_g = 1$). L2 regularization ($\alpha_r = 0.01$) anchors r_g toward this prior, preventing unconstrained collapse. The UMI shift mismatch when $r \neq 1$ is small (because r is regularized near 1) and the output head has sufficient capacity to compensate.

NB collapse is diagnostic, not artifactual. The unregularized NB collapse (PAP-289: Spearman → 0.047, z-loss 186k → 330k) is clearly driven by r_g absorbing variance, not by shift miscalibration or numerical issues. The collapse pattern — good NLL coexisting with destroyed rankings — is a signature of the model fitting technical noise at the expense of biological structure.

One limitation. The NB dispersion r_g is implemented as a bare `nn.Parameter` without weight decay separate from the L2 regularizer. More sophisticated parameterizations (e.g., ABC reparameterization, separate optimizer groups) might improve NB performance. However, the need for such engineering itself supports the thesis: count objectives require careful tuning to avoid overfitting, while GPL works with default settings.

Hyperparameter sensitivity. The shared hyperparameters (α_z for z-loss, s_0 for depth scaling) were set once and used across all objectives. For NB, we tested two settings: unregularized ($\alpha_r = 0$) and regularized ($\alpha_r = 0.01$). The unregularized NB collapses catastrophically; the regularized NB stabilizes but remains well below GPL. The theoretical bound is clear: as $\alpha_r \to \infty$, r_g is forced to 1 and NB recovers GC exactly — so NB's best possible

performance with strong regularization is GC's performance, which itself is below GPL. No value of α_r can close the gap.

A.9 Cross-Evaluation — All Objectives Evaluated Identically

All objectives produce per-gene logits from the same output head. Spearman evaluation is identical for all:

```
argsort(logits, descending=True) -> predicted ranking -> Spearman vs ground truth
```

No objective-specific transformation is applied. The evaluation is objective-agnostic: it asks "do the logits rank genes in the right order?" regardless of how those logits were trained.

"Isn't Spearman biased toward the ranking objective?" All objectives produce per-gene logits that induce a ranking. A count-based model that correctly predicts counts also correctly predicts rankings — accurate count prediction is *sufficient* for accurate ranking prediction (but not necessary, which is GPL's advantage). GPL also wins on **binary correctness** — the fraction of truly expressed genes predicted as expressed — which is a set-overlap metric with no ordinal structure. This confirms that GPL's advantage is not an artifact of evaluating on a ranking metric.

For count-level evaluation (reported separately), each objective uses its native inference:

- **GPL:** $e^{\phi} = \lambda \rightarrow$ Geometric rates (not calibrated as counts, since GPL trains on rankings only)
- **GC:** $e^{\phi} = \lambda \rightarrow$ expected count directly
- **PROP:** $\text{softmax}(\phi) \times \text{UMI} \rightarrow$ counts
- **NB:** $e^{\phi}, \text{softplus}(r_g) \rightarrow r \cdot \lambda$ (NB expected value)

A.10 Computational Cost

The GPL factorization is not only theoretically motivated but practically efficient. The binary term requires only independent per-gene evaluations. The ranking term operates over expressed genes only. In contrast, the PROPORTIONAL softmax must normalize over all ~36,000 genes including ~30,000 with zero counts.

A.11 Full Metrics

All evaluations at `qc_4096`, masking = 0.50.

Scale	Params	GPL	MSE	PROP	GC (best†)	GPL−GC
d=512	250M	**0.6159**	0.6134	0.6055	0.6107†	+0.005
d=1024	1B	**0.6456**	0.6369	0.6382	0.6334†	+0.012
d=1536	2B	**0.6681**	0.6572	0.6572	0.6291†	**+0.039**

Spearman (scipy) — full-transcriptome ranking correlation:

Scale	GPL	MSE	PROP	GC (best†)	GPL−GC
d=512	**0.6324**	0.6304	0.6227	0.6275†	+0.005
d=1024	**0.6598**	0.6515	0.6526	0.6486†	+0.011
d=1536	**0.6806**	0.6705	0.6704	0.6453†	**+0.035**

Binary correctness — detection accuracy: †GC "best" = pre-cooldown peak. GC collapsed during cooldown at all scales tested (d=512: 0.6107→0.5735; d=1024: 0.6334→0.6316; d=1536: collapsed catastrophically).

GPL wins at every scale on both metrics. Fan-out is monotonic: Spearman gap +0.005 → +0.012 → +0.039; binary correctness gap +0.005 → +0.011 → +0.035.

A.12 Summary — Two Independent Failure Modes

The four-way ablation at d=1536 (2B parameters), together with NB experiments at d=1024 (1B), reveals two independent failure modes for non-ranking objectives. At d=1536: GPL (0.6681) > MSE ≈ PROP (0.6572) > GC (0.6291). At d=1024: NB collapses catastrophically (see §A.7). All objectives were pretrained on the same multi-dataset corpus spanning diverse chemistries and sequencing depths; the results reflect cross-dataset training, not single-dataset performance.

Failure mode 1: Capacity waste on uninformative targets (PROP, MSE). PROP and MSE both spend gradient on targets that are dominated by technical rather than biological variation. For PROP, this takes three forms: (i) the softmax normalization couples all ∼36,000 genes through a shared denominator, so every training step wastes gradient on ∼30,000 zero-count genes (§A.5–A.6, the 1800× gradient suppression for most genes); (ii) proportions are compositional — they sum to one — so chemistry-specific capture biases on any subset of genes propagate through the normalizer to the proportions of all genes, making the targets non-comparable across experiments with different chemistries; and (iii) proportions are still cardinal values, so the model fits magnitudes (e.g. "gene X is 0.003 of the transcriptome") rather than ordinal structure, retaining depth-dependent noise that rankings discard. For MSE on log1p counts, the targets are even noisier: Warton (2018) proved that no monotonic transformation can stabilize variance when mean counts fall below one — the regime where the bulk of the transcriptome resides.

Proportional objectives also lack a natural mechanism for depth conditioning. Deeper sequencing doesn't just increase existing counts — it detects new genes at the zero-to-nonzero boundary, the single noisiest transition in the data (a gene observed at count 1 has roughly equal probability of representing real expression versus technical dropout). Each newly detected gene enters the softmax denominator, changing the target proportion for every other gene. The noisiest possible signal — "did this one molecule get captured or not" — propagates through the normalizer to contaminate the learning signal for all 36,000 genes. The logit-space depth shift that conditions GPL and GC cannot help here: softmax is shift-invariant (adding a constant to all logits before softmax changes nothing). PROP has no knob to turn for depth conditioning; it absorbs the depth variation into the learned representations, which is exactly the capacity waste the objective can least afford.

That PROP and MSE converge to identical held-out performance at 2B (both 0.6572) despite their very different mathematical forms suggests a shared ceiling set by the decision to fit cardinal values rather than ordinal structure.

Failure mode 2: Count fitting / noise memorization (GC, NB). Objectives that predict count magnitudes develop opposing gradient forces at scale: the stopping term and counting term pull in opposite directions, and larger models push logits to extremes where these forces grow. This causes *collapse*, not just a ceiling. GC declined from peak performance at every scale tested. NB, with its additional dispersion parameters, collapsed catastrophically — learned dispersions absorbed technical variance instead of learning biology. More distributional flexibility means more capacity to memorize sequencing noise. This failure mode is more destructive than failure mode 1 at 2B — a ceiling is better than a cliff.

The MSE–GC crossover. One of the most revealing results in the ablation is that MSE — a naive heuristic with a variance-stabilizing transform that provably fails in the relevant regime — outperforms GC at 2B, despite GC being a proper generative likelihood for the data. GC is the more principled model: it correctly specifies the

Geometric distribution and scores each gene's observed count against its own parameter. MSE is theoretically unjustified. Yet at 2B, the principled model collapses and the naive one doesn't. The explanation: MSE is too simple to overfit the noise structure. GC is sophisticated enough to memorize it and does. Being more correct about the data-generating process is *actively harmful* when that process includes technical noise and the model has sufficient capacity to learn it. This inverts the field's conventional wisdom — the standard trajectory from Poisson to Geometric to Negative Binomial to ZINB, each step a better count model, is a ladder leaning against the wrong wall.

Caveat on regularization. We did not exhaustively sweep regularization for GC at any scale. It is possible that more aggressive regularization (weight decay on the output head, logit magnitude constraints, gradient clipping) could prevent GC's collapse at 2B. However, the NB results suggest this is not a path to competitive performance: regularized NB avoided the catastrophic explosion of unregularized NB but still performed poorly, well below GC and MSE — regularization prevented the worst failure mode but did not produce good representations. The structural issue is that count-level objectives reward noise memorization at their optimum. Regularization fights the optimizer — the loss says "fit the counts precisely" while the regularizer says "don't" — and the resulting representations bear the scars of both. GPL does not require this balancing act because its optimum does not include technical noise: the ranking is approximately invariant to the noise process, so there is nothing for excess capacity to memorize.

GPL avoids both. The ranking term has no zero coupling (unexpressed genes form a single tied bucket handled by the binary term). It has no compositional coupling (independent per-gene logits). It fits no cardinal values (the loss sees only the ordering, not magnitudes, so there is no incentive to push logits to extremes and no depth-dependent noise to memorize). These are not separate engineering fixes. They are consequences of a single design choice: model the ranking rather than the counts.

A.13 Practical Recommendations

Practical recommendations for virtual cell pretraining. Based on our ablation across five objectives at multiple scales:

1. **Use ranking objectives, not count reconstruction.** The gap between GPL and count-based objectives widens with model scale — the advantage compounds, meaning the choice of objective matters *more*, not less, as you scale up. If you are planning to invest significant compute in training a large virtual cell model, this is the single most consequential design decision.
2. **Parameterize in logit space with `logsigmoid` — never use epsilon-guarded `log`.** The logit parameterization is numerically exact at all distribution boundaries. The (μ, θ) parameterization with epsilon guards is the field default, but it introduces gradient spikes, catastrophic cancellation, and a hidden hyperparameter. These are not edge cases — they dominate the typical scRNA-seq training batch, where most genes have near-zero expression.
3. **Use independent per-gene logits, not softmax-coupled proportions.** Softmax over ~36,000 genes suppresses gradients for all but the top ~100 by a factor of 10^5. The model never learns rich representations for the low-expression genes that matter most for fine-grained biology. Independent logits give every gene equal access to the gradient signal.
4. **Condition on depth via the logit-space additive shift.** Four lines of code replace scVI's latent library size variable, scGPT's rank-value encoding, and Geneformer's implicit normalization. The shift is differentiable, model-agnostic, and the learned scaling factor adapts to data automatically.

5. **Resist the temptation to add distributional parameters.** The field has been climbing the distributional complexity ladder — Poisson → NB → ZINB — assuming that better count models yield better representations. Our ablation shows the opposite: every additional degree of distributional freedom is an opportunity to overfit technical variance rather than learn biology. The simplest objective (GPL, zero distributional parameters per gene) produces the best representations, and the advantage grows with scale.

A.14 Loss Function History — From MLM to GPL

The GPL objective was the result of iterative development, each step encoding more of the sampling-noise equivariance structure into the loss function. We document this progression because the intermediate objectives illuminate *why* GPL works — each step was more principled in its treatment of ties and count noise, and each worked better empirically.

Phase 1–2: Cross-entropy on gene tokens. Standard MLM. Each masked gene predicted independently. No ranking structure.

Phase 3: Rank-pooled CE with entropy adjustment. Predictions averaged across items at the same rank. The irreducible entropy $\log(N_r)$ is subtracted. Recognizes that tied items are informationally indistinguishable, but pooling is post-hoc.

Phase 4: Smeared CE / Spearman loss. Partial credit for nearby predictions with exponential decay. Captures the intuition that adjacent ranks carry similar information, but the smearing bandwidth is a hyperparameter.

Phase 5: Plackett–Luce (exponential latents). A proper generative model: items chosen sequentially with probability proportional to λ. Principled, but assigns zero probability to ties — intractable with ~30,000 genes at count zero.

Phase 6: Geometric Plackett–Luce (current). Replaces exponential with discrete Geometric latents (Henderson 2022). Ties have positive probability. The group factor depends only on θ values and group size — tie equivariance is a mathematical property of the distribution, not a heuristic.

Each step encoded more structure into the loss; each worked better. The ablation (Fig. 1) provides empirical confirmation that the final form outperforms all predecessors.

B The data generating process: from chemistry to count distributions

This supplement provides a self-contained account of how single-cell gene expression data are generated, why the resulting count distributions take the form they do, and what this implies for the choice of pretraining objective. We begin with the physical chemistry of single-cell sequencing (§B.1), trace the field's evolving understanding of zeros and dropout (§B.2–§B.3), derive analytically why sequencing depth determines the entire count distribution (§B.4), and connect this to the GPL framework developed in Supplement S1 (§B.5).

The material in §B.1–§B.3 draws on and formalizes arguments originally presented in "From Transcriptomes to Transformers: A New Machine Learning Paradigm for Scalable Biological Simulators"[2]. §B.4–§B.5 present, to our knowledge, the first analytical characterization of integer count distributions as a function of sequencing depth.

[2]Green, A. "From Transcriptomes to Transformers." https://x.com/adamlewisgreen/status/1988727157112361362

B.1 How Single-Cell Transcriptomics Data Are Generated

The data we train on are single-cell gene expression profiles. Cells vary in their functional properties based in part on which genes they express and at what levels. When a gene is expressed, the cell creates RNA copies called transcripts. These transcripts serve diverse functions: acting as templates for protein synthesis, regulating the expression of other genes, and catalyzing biochemical reactions.

We now have the ability to read out snapshots of these gene expression states from individual cells by capturing, sequencing, and counting these transcripts. The resulting cell profile is a vector of integer counts per gene, where the vocabulary comprises approximately 20,000 protein-coding genes in the human genome. This profile is the core data primitive we train on: 5 of gene A, 2 of gene B, 3 of gene C, 0 of gene D, and so on.

The journey from cell to count vector involves several steps, each introducing potential sources of noise. We focus on 3' end-capture droplet-based methods (10x Genomics and similar platforms), which generate the majority of public single-cell data:

1. **Cell encapsulation.** Individual cells are flowed through a microfluidic channel and encapsulated inside oil droplets along with gel beads (10x Genomics, 2020, 2024).

2. **Transcript capture.** The gel beads carry oligonucleotides composed of a cell-level barcode (shared among all oligos on the bead), a unique molecular identifier (UMI, unique to each oligo), and a poly-dT sequence. Upon lysis, the cell releases its RNA transcripts—approximately 10^5–10^6 mRNA molecules in a typical mammalian cell (Milo and Phillips, 2015). The poly-dT sequences selectively capture polyadenylated mRNAs via complementary base pairing with their poly-A tails, creating a natural handle for capture of ~20,000 protein-coding genes plus polyadenylated non-coding RNAs.

3. **Reverse transcription and barcoding.** Captured mRNA is reverse transcribed into cDNA, with each molecule now carrying its cell barcode and UMI, indexing its cell and transcript of origin.

4. **Amplification and library preparation.** The oil droplets are broken and the barcoded cDNAs from all cells are pooled for PCR amplification and library preparation. Amplification creates multiple copies of each cDNA molecule—all carrying the same UMI—increasing the probability of detecting each one during sequencing.

5. **Sequencing.** The amplified library is loaded onto a DNA sequencing flow cell. Individual cDNA molecules are captured and sequenced, generating millions of short reads, each containing the cell barcode, UMI, and approximately 90+ nucleotides derived from the original RNA transcript.

6. **Alignment and quantification.** Each read's transcript sequence is aligned to the reference genome to identify which gene it originated from—effectively the "tokenization" step. Reads are grouped by cell barcode and gene. Within each cell, UMIs identify which reads came from the same original mRNA molecule versus which are PCR duplicates. Reads sharing the same barcode, UMI, and gene are collapsed to a single count. The result is an integer count vector for each cell: a cells $\times$ genes count matrix where each entry is the number of unique mRNA molecules detected for that gene in that cell.

Not every transcript successfully makes this journey. Along the way, there are multiple sources of both biological and technical variation that influence the resulting counts. It is the nature of this variation—and its consequences for training models—to which we now turn.

B.2 The Sparsity Problem and Detection Limits

Observed single-cell gene expression profiles are sparse: 70–90% of entries are zeros across the ~20,000-gene vocabulary. One of the earliest questions the field faced was whether these zeros reflected true biological states (genes genuinely not expressed) or technical artifacts (transcripts present but not detected).

The earliest droplet methods papers addressed this using synthetic spike-in RNA controls. By injecting solutions with known RNA concentrations into droplets, biological heterogeneity could be ruled out and any observed zeros attributed to technical factors. The original inDrop method (Klein et al., 2015) found that truly present synthetic RNAs were routinely undetected, with approximately 8% transcript capture efficiency—requiring ~10 molecules in the cell for a 50% detection probability and ~45 for >95%.

Svensson et al. (Svensson et al., 2017) systematically quantified detection limits across multiple protocols using spike-in controls. Detection sensitivity depended critically on sequencing depth: moving from 100K to 1M reads per cell dramatically improved sensitivity. Protocols varied greatly in their detection limits, with some showing saturation around 4.5M reads per cell.

Ziegenhain et al. (Ziegenhain et al., 2017) extended this analysis to endogenous RNA rather than spike-ins. For some protocols, even at 10^6 reads per cell, genes with low true abundance had approximately 70% probability of remaining undetected—demonstrating that the detection problem was not fully resolved even with substantial sequencing.

B.3 The Dropout Debate and Its Resolution

The field initially treated the abundance of zeros as requiring special statistical treatment, proposing zero-inflated models (Risso et al., 2018) and biological mechanisms such as bursty transcriptional kinetics to explain the apparent excess of zeros beyond what standard count distributions would predict.

Svensson (Svensson, 2020) resolved this debate. Using spike-in RNA controls from multiple studies, he showed that for each gene, the observed dropout rate matched exactly what a standard negative binomial model would predict given that gene's mean expression level and the experiment's technical properties. No special zero-inflation term was needed. The zeros were simply a natural consequence of sampling from count distributions.

Sarkar and Stephens (Sarkar and Stephens, 2021) provided the theoretical framework. They distinguished between three types of models:

1. An **expression model** describing how true expression levels vary among cells (biological variation)
2. A **measurement model** describing how observed counts deviate from true expression due to technical noise
3. The combined **observation model** for the actual counts we see

These distinctions resolved considerable confusion in the field, where terms like "dropout," "zero-inflation," and "missing data" had been used inconsistently without clarifying whether they referred to the expression process or the measurement process.

Sarkar and Stephens showed that observed counts arise naturally from mixing biological expression variation (Gamma-distributed) with Poisson measurement noise, producing a negative binomial distribution—with no special dropout term required. The zeros fall out naturally from this framework: they are simply noisy measurements that occur when sampling from genes with low expression levels.

This resolved the statistical modeling question. But it left open the mechanistic question: what drives the dramatic variation in total zeros across cells of the same type? Two T cells in the same dataset, expressing similar genes at similar levels, might have 5,000 versus 8,000 zeros across 20,000 genes. The zeros follow from sampling in the measurement model—but what aspect of the sampling process varies so dramatically from cell to cell?

B.4 Sequencing Depth Determines the Entire Count Distribution

Sarkar and Stephens' framework points to the measurement noise term. Recall that the measurement process has two components: transcripts must first be captured (via poly-dT binding and reverse transcription) and then the captured molecules must be sequenced. Which factor drives the observed variation?

Modern capture rates suggest it is not primarily capture. Early methods were capture-limited: inDrop had ~8% capture, 10x v2 chemistries reached 14–15%. But 10x v3 chemistry achieved 30–32% capture—a ~4× improvement (10x Genomics, 2019). With modern capture rates, a substantial fraction of transcripts are captured. The bottleneck has shifted: we capture molecules but fail to sequence them all.

After capturing and tagging each molecule with a UMI, the library is amplified via PCR and loaded onto a sequencing flow cell. The sequencer generates millions of reads, and "sequencing depth" refers to how many reads are generated per cell. If a gene has 10 transcripts, 30% capture yields 3 molecules. Whether those 3 molecules are detected depends on sequencing depth. With insufficient reads, none may be sequenced—the gene appears as a zero despite being both present and captured.

The sequencing saturation curve reveals the bottleneck: plotting reads per cell against unique UMIs detected shows whether captured transcripts are being fully sequenced. Recent benchmarking of seven single-cell chemistries on PBMCs (Elz et al., 2025) showed that widely used platforms (GEM-X 3', NextGEM 3', Flex) had UMI detection still rising steeply at 25,000 reads per cell—nowhere near saturation. And PBMCs are small, RNA-poor cells; metabolically active cell types would show even steeper curves.

Choi et al. (Choi et al., 2020) quantified the consequences empirically: total UMI count per cell explained 95% of the variation in zeros within cell types. This relationship held within individual cell types—cells of the same type followed the same curve, varying by approximately 3–5× in detected UMIs. The sources of this per-cell UMI variation include biological extrinsic noise (cell size, cell cycle phase, transcriptional activity), Poisson sampling during capture, and stochastic sequencing allocation across barcodes (Fang and Pachter, 2025). For our purposes, the mechanism is less important than the consequence: whatever its source, variation in total UMI per cell determines the count distribution (§B.5).

Critically, the consequences of sequencing depth extend beyond zeros. At every expression level, observed counts are noisy samples of an underlying biological signal, and the noise magnitude is a direct function of depth. The relationship between sequencing depth and the count distribution extends beyond the well-characterized "dropout" of zeros: under Poisson sampling, true counts of k are observed as $k-1$, $k-2$, … 0 with probabilities that depend on depth. At typical depths (5,000–10,000 UMI), the majority of expressed genes have observed counts of 1–3, where measurement error dominates biological signal. The entire integer count distribution is shaped by a single technical parameter.

To our knowledge, the following section presents the first analytical derivation of this relationship.

B.5 The Geometric Histogram — Analytical Derivation

We now show that the count distribution observed in single-cell data—the number of genes at each integer count level—follows a Geometric distribution whose parameter is set entirely by sequencing depth. This result connects the empirical observation (exponential decay of count histograms) to the measurement model (Poisson sampling) and to the GPL framework (S1).

Setup. Consider a cell with V genes, each with true expression proportion π_g (the fraction of the cell's mRNA belonging to gene g). The cell is sequenced to depth D (total UMI). Under the Poisson measurement model (Sarkar and Stephens, 2021), each gene's observed count is independently drawn:

$$k_g \sim \text{Poisson}(\pi_g \cdot D)$$

Question. How many genes have observed count exactly k? Summing over genes:

$$n(k; D) = \sum_g P(k_g = k \mid \lambda_g = \pi_g D)$$

We can convert this sum over 36,000 individual genes into an integral over the continuous distribution of gene proportions:

$$\underbrace{\sum_{g=1}^{V} P(k \mid \pi_g, D)}_{\text{sum over genes}} \approx \underbrace{V \int_0^\infty f(\pi) \cdot P(k \mid \pi, D)\, d\pi}_{\text{integral over proportion distribution}}$$

where $f(\pi)$ is the density of true gene proportions across the transcriptome. If you plotted a histogram of all 36,000 π_g values, you would see a massively right-skewed distribution: a huge pile near zero (most genes are barely expressed), a long tail out to the few highly expressed housekeeping genes. That histogram is $f(\pi)$. The integral asks: for each possible true proportion π, how many genes have that proportion ($f(\pi)$), and what is the probability each shows observed count k at depth D (the Poisson term)? Integrating over all π gives the total count:

$$n(k; D) \approx V \int_0^\infty f(\pi) \cdot \frac{(\pi D)^k\, e^{-\pi D}}{k!}\, d\pi$$

The key assumption. Gene expression proportions are approximately exponentially distributed: most genes are barely expressed, a few are highly expressed. We model $f(\pi) = \alpha\, e^{-\alpha\pi}$ for some rate $\alpha > 0$, which the scBaseCount data confirms empirically (§B.9).

Derivation. Substituting $f(\pi) = \alpha\, e^{-\alpha\pi}$:

$$n(k; D) = V \cdot \alpha \cdot \frac{D^k}{k!} \int_0^\infty \pi^k\, e^{-\pi(\alpha+D)}\, d\pi$$

The integral is a standard Gamma function: $\int_0^\infty \pi^k\, e^{-\pi(\alpha+D)}\, d\pi = \frac{k!}{(\alpha+D)^{k+1}}$. The $k!$ cancels:

$$n(k; D) = V \cdot \frac{\alpha}{\alpha + D} \cdot \left(\frac{D}{\alpha + D}\right)^k$$

Defining $p = D/(\alpha + D)$:

$$\boxed{n(k; D) = V \cdot (1 - p) \cdot p^k} \tag{2}$$

This is a Geometric distribution. The number of genes at each count level decays geometrically with ratio $p = D/(\alpha + D)$. This is not an empirical regularity—it is an exact consequence of Poisson sampling from an exponentially distributed transcriptome.

Connection to Sarkar and Stephens (2021). This result fills a specific gap in their framework. Sarkar and Stephens showed that combining a Gamma expression model with a Poisson measurement model yields a Negative Binomial observation model—the general case. Our derivation is the special case where the Gamma shape parameter equals 1, i.e., the expression model is Exponential:

Expression model $f(\pi)$	+ Poisson measurement	= Observation model	Source
Gamma(α, β)	Poisson(πD)	Negative Binomial	Sarkar & Stephens 2021
Exponential(α) **= Gamma**$(1, \alpha)$	Poisson(πD)	**Geometric = NB**$(r = 1)$	**This work**

The Exponential is the maximum-entropy distribution for non-negative values with a given mean—the least informative assumption about gene proportions given only their average expression level. The fact that this simplest possible expression model produces the Geometric—exactly the distribution GPL assumes—is not a coincidence.

It means GPL's distributional assumption is the correct marginal model under the least informative prior about gene expression, combined with the Poisson measurement model the field has already validated.

We observed the Geometric count histogram empirically before deriving it analytically, and GPL's Geometric assumption was chosen independently for tractability of the ranking likelihood—not for the marginal count distribution. That all three converge on the same distribution is either a happy accident or a consequence of the underlying measurement process.

The S1 ablation (§S1.7) showed empirically that adding distributional flexibility beyond the Geometric (NB with learned $r > 1$, corresponding to Gamma shape $\neq 1$) degrades pretraining transfer. The derivation here shows why: the Geometric is already correct for the marginal count histogram. Adding overdispersion via r doesn't improve the model of biological variation—it gives the optimizer a knob to fit the residual technical variation that the Geometric's simplicity correctly ignores.

Connection to S1 notation. In the GPL framework (S1), each gene has a Geometric stopping probability $\theta_g = \sigma(-\phi_g)$. The marginal count histogram derived here shows that the *population-level* count distribution is also Geometric, with stopping probability $1 - p = \alpha/(\alpha + D)$. The GPL assumption—that gene counts follow a Geometric distribution—is not a convenient approximation. It is the correct marginal distribution for this data generating process.

The depth shift developed in S1 (§S1.3) removes the D-dependence from the Geometric rate parameter: the shift $\phi'_g = \phi_g + \log(\mu_{\text{tgt}}) \cdot s_0$ adjusts the baseline to account for depth. The formula $p = D/(\alpha+D)$ is the theoretical justification for why this shift works—it is removing the first-order D-dependence from the count distribution's shape.

B.6 The Two Poles of Measurement Quality

The parameter $p = D/(\alpha + D)$ continuously interpolates between two extremes of measurement quality. Understanding these poles clarifies what information is available to any model—ranking-based or count-based—at different sequencing depths.

The binary pole ($D \to 0$, $p \to 0$). $N(0) = V(1-p) \approx V$. Almost every gene registers as zero. $N(1) \approx V \cdot p$—a tiny number of genes at count 1. $N(2) \approx V \cdot p^2$—essentially zero genes at count 2 or above.

The entire transcriptome is compressed into two bins: zero and one. Genes at count 1 cannot be ranked relative to each other—they are all tied. The only reliable signal is detection: expressed or not expressed. The count distribution has collapsed to a binary state.

This is exactly the regime where Bouland et al. (Bouland et al., 2023) found that binarized representations perform comparably to full counts for clustering and cell type identification. The derivation shows this is not surprising: at typical depths, the count distribution IS approximately binary. Binarization does not discard information because there is almost no information beyond binary to discard.

The ordinal pole ($D \to \infty$, $p \to 1$). $n(k) = V(1-p) \cdot p^k$ with slow decay. Genes spread across counts 0, 1, 2, … 50, 100, 500. Ties become rare at high counts because the Poisson standard deviation ($\sqrt{k}$) is small relative to the count itself. The ranking among expressed genes becomes reliable—the ordinal structure converges toward the true biological ordering.

But even at high depth, the low end of the distribution remains noisy. Count 1 versus count 2 is still a single molecule's difference. The bottom of the ranking is always unreliable; the top becomes increasingly reliable with depth.

Depth D	p	% zeros	Expressed genes	Informative signal
2,000	~0.04	~95%	~200 (mostly count 1)	Binary + minimal ordinal
10,000	~0.17	~83%	~3,000 (counts 1–20+)	Rich ordinal structure
50,000	~0.50	~50%	~10,000 (counts 0–100+)	Near-complete ordinal structure

Regime	p	Count distribution	Reliable signal	What GPL does
Binary pole	$p \to 0$	All mass at 0 and 1	Detection only	Binary term dominates; ranking term is trivial
Ordinal pole	$p \to 1$	Spread across many values	Full ranking	Ranking term extracts rich biological structure
Typical data	$p \approx 0.05$–0.20	Geometric decay, most at 0–3	Binary + weak ordinal	Both terms contribute, weighted by information content

The continuous interpolation: GPL handles the entire spectrum with a single likelihood. At the binary pole, the ranking term contributes little gradient (everything is tied) and the binary term does the work. At the ordinal pole, the ranking term extracts rich signal. In between, both terms contribute in proportion to their information content. The model does not need to know which regime it is in—the GPL factorization (S1, §S1.2) automatically allocates capacity to wherever the signal is.

B.7 Why Variance-Stabilizing Transforms Cannot Help

A natural response to the depth-dependence of count distributions is to apply a variance-stabilizing transformation—log, square root, or more sophisticated methods—to make counts comparable across depths. Warton (2018) proved that this cannot work in the relevant regime (Warton, 2018).

The Poisson distribution has variance equal to its mean: $\mathrm{Var}(k) = \lambda$. For a gene with true expression rate λ, the coefficient of variation is $1/\sqrt{\lambda}$—exceeding 50% whenever $\lambda < 4$. At typical sequencing depths, the majority of expressed genes have $\lambda < 3$ (the Geometric decay from §B.5), placing them squarely in the high-noise regime.

Warton proved that no monotonic transformation can stabilize variance when mean counts fall below one—the regime where the bulk of the transcriptome resides at typical depths. The $\log(1 + x)$ transform, widely used in the field, actually makes the problem worse at low counts: it compresses the already-narrow range of low counts (0, 1, 2, 3) while expanding differences between high counts, inverting the relationship between signal quality and emphasis.

More principled approaches such as analytic Pearson residuals (Hafemeister and Satija, 2019) model the mean-variance relationship explicitly but remain count-level targets subject to the same depth-dependent noise floor. The residuals stabilize variance for intermediate counts but cannot rescue the low-count regime where most genes reside.

The derivation in §B.5 shows why this must be the case: the count distribution $n(k; D) = V(1-p) \cdot p^k$ depends on D through $p = D/(\alpha + D)$. The parameter p determines both the decay rate AND the interaction between the Poisson measurement model and the gene proportion distribution. No single transformation of the counts can remove both effects simultaneously.

Rankings sidestep this entirely. The ordering of genes by expression is approximately invariant to D—deeper sequencing shifts all counts up proportionally, preserving relative ordering while changing absolute magnitudes. The ranking is a (approximate) sufficient statistic for the biological ordering. The counts are a sufficient statistic for the noise.

B.8 Implications for Pretraining Objectives

The results of this supplement have direct consequences for the choice of pretraining objective:

1. **Count-based objectives train on a distribution whose shape is set by a technical parameter.** The Geometric count histogram is parameterized by $p = D/(\alpha + D)$. A model minimizing count reconstruction error is, to first approximation, learning to predict sequencing depth artifacts.

2. **The depth shift (S1, §S1.3) is theoretically grounded but incomplete.** The shift removes the first-order D-dependence by adjusting the Geometric rate parameter. But the full count distribution depends on D through both p and the Poisson-exponential interaction—effects that no single shift can remove.

3. **Rankings are robust to the noise that counts are not.** Among expressed genes, relative ordering is approximately preserved under variation in D. The ranking objective captures the depth-invariant biological signal while grouping the depth-dependent zeros into a single tied bucket.

4. **Binarization validates the intuition but overcorrects.** At low depths, the count distribution is approximately binary—binarization discards little information. But at moderate and high depths, the ordinal structure among expressed genes is rich and biologically meaningful (cf. STAT2 binding strength, $r = 0.42$, recovered from ordinal structure alone). Rankings retain both the binary detection signal and the ordinal relationships among expressed genes.

5. **GPL's Geometric assumption is not arbitrary.** The marginal count distribution IS Geometric under Poisson sampling from an exponentially distributed transcriptome. GPL is not fitting a convenient parameterization—it is fitting the correct distribution for this data generating process.

B.9 Empirical Validation (scBaseCount)

We have confirmed empirically that per-cell count histograms follow the predicted Geometric decay across a range of sequencing depths and tissue types in scBaseCount data, with the single-parameter theoretical fit $p = D/(\alpha + D)$ closely matching observed distributions. The full empirical analysis with visualizations will be added in a subsequent revision of this manuscript.

C From Thurstone to GPL: intellectual history and model selection

This supplement traces the intellectual lineage of the GPL training objective, explains why alternative ranking models are unsuitable for gene expression data, and introduces the distinction between elicited and induced rankings. The ICC-based measurement reliability analysis connecting Thurstone's framework quantitatively to single-cell sequencing noise is deferred to a future publication pending empirical validation on technical replicate data.

C.1 A Century of Ordinal Measurement

The insight that ordinal comparisons are more reliable than cardinal measurements under noisy observation has a 99-year intellectual lineage. Our work brings this thread into the era of foundation models.

Thurstone (1927). Thurstone published "A Law of Comparative Judgment" — one of the founding documents of mathematical psychometrics. His core insight: when a stimulus is presented to an observer, the resulting internal perception is not a fixed number but a random variable. Thurstone called this the **discriminal process** and its spread the **discriminal dispersion**. The consequence: absolute ratings are unreliable under noise, but ordinal comparisons are robust — the pairwise probability $P(A > B)$ is a clean function of the true distance between stimuli, even when individual percepts are noisy.

The parallel to single-cell genomics is structural:

Thurstone (1927)	Single-cell genomics
True stimulus quality on latent continuum	True gene expression level
Discriminal process: noisy internal percept	Observed count: Poisson sample from true expression
Discriminal dispersion: perception noise width	Sequencing depth: controls count noise magnitude
Absolute rating (e.g., "7/10"): unreliable	Raw count (e.g., 5 UMI): unreliable
Pairwise comparison ($A > B$): robust	Ranking (gene A > gene B): robust
Noise partially cancels in comparisons	Rank ordering approximately preserved under depth variation

Thurstone recognized in 1927 that ordinal data is more informative than cardinal data precisely when the measurement process is noisy. He did not have Plackett-Luce, Geometric distributions for ties, or 100 million cells to train on.

Luce (1959). Luce's *Individual Choice Behavior* axiomatized the relationship between pairwise comparisons and ranking probabilities. Luce's choice axiom states: the probability of choosing item i from any set is proportional to a quality parameter λ_i, independent of what other items are in the set — the "independence of irrelevant alternatives" (IIA). For gene expression, IIA translates to: the relative ordering between two expressed genes does not depend on what other genes are in the cell. This is approximately true and provides the formal justification for why rankings are robust to dropout — the ranking of detected genes is valid regardless of which genes dropped out.

Plackett (1975). Plackett gave the explicit likelihood for a complete ranking under Luce's axiom:

$$P(\sigma \mid \lambda) = \prod_{i=1}^{n} \frac{\lambda_{\sigma(i)}}{\sum_{j=i}^{n} \lambda_{\sigma(j)}}$$

This sequential factorization — pick the top item, remove it, repeat — is computationally tractable and provides a proper generative model of rankings.

The tie problem. Standard Plackett-Luce uses continuous latent variables (exponential), which means ties have probability zero. In single-cell data, ~70% of genes share count zero, and most detected genes have counts of 1–3 with massive ties. Marginalizing over all tie-consistent permutations is combinatorially intractable.

Henderson (2022). The Geometric Plackett-Luce model replaces continuous exponential latent variables with their discrete counterparts — geometric random variables. Because the Geometric distribution is discrete, ties arise naturally when multiple items draw the same waiting time. The group factor for tied items depends only on their θ values and the group size, not on any internal ordering — making tie equivariance a mathematical property of the distribution. Baker (2020) and Baker and Scarf (2021) proved that the exponential and Geometric are the *only* continuous and discrete distributions, respectively, that yield closed-form ranking likelihoods — a uniqueness that follows from memorylessness. GPL is not one ranking model among many; it is the unique tractable discrete ranking model.

Our contribution. We operationalize this 99-year lineage as a self-supervised pretraining objective for foundation models:

1. Thurstone (1927): Ordinal comparisons are more reliable than cardinal measurements under noise
2. Luce (1959): Ordinal preferences can be axiomatized as sequential choices proportional to quality parameters
3. Plackett (1975): The explicit likelihood for a complete ranking
4. Henderson (2022): Geometric extension that handles ties — the critical missing piece for biological data
5. This work: GPL pretraining at scale on single-cell data, with empirical evidence that the ranking objective changes scaling laws

Each step addressed a limitation of the previous one. Thurstone had the insight but no likelihood. Luce had the axiom but no ranking model. Plackett had the ranking model but no ties. Henderson had ties but no application to biological data at scale. We close the loop.

C.2 Why Not Simpler Ranking Models?

Given that the core insight is "train on ordinal structure," one might ask: why GPL and not something simpler? Several alternatives exist, each with a specific limitation:

Model	Gen.?	Tractable at V=36K?	Ties?	Why eliminated
Bradley-Terry (Bradley and Terry, 1952)	Yes	No — $O(n^2)$	Partial	~650M pairs per cell at V=36K
Thurstone-Mosteller (Thurstone, 1927)	Yes	No — $O(n^2)$	Partial	Same $O(n^2)$ as BT
Mallows (1957)	Yes	No — $O(n!)$	Partial	Normalizing constant intractable
RankNet (Burges 2005)	No	Yes	No	Discriminative — no likelihood
Spearman loss	No	Yes	No	No likelihood, no tie handling
Plackett-Luce (Plackett, 1975)	Yes	Yes — $O(n \log n)$	No	$P(\text{tie}) = 0$; ~30K zeros
GPL (Henderson, 2022)	**Yes**	**Yes — $O(V+S \log S)$**	**Yes**	**Selected**

GPL is the unique model in this landscape that is simultaneously generative (proper likelihood — you can sample from it), computationally tractable ($O(V + S \log S)$), and handles ties (Geometric latents). Every other ranking model fails on at least one of these three requirements. This is not a matter of taste — GPL is the only ranking model that works for single-cell data.

C.3 Elicited vs Induced Rankings — A Novel Distinction

Most applications of Plackett-Luce model **elicited rankings**: ordinal preferences directly observed through explicit comparison (horse racing, golf leaderboards, A/B tests, reinforcement learning from human feedback). In these settings, the ranking IS the primary observation — someone asked "which do you prefer?" and recorded the answer.

Our setting is fundamentally different. We observe noisy cardinal measurements (gene expression counts) and **induce rankings** from them — extracting ordinal structure from data that was never intended as a comparison. The counts were measured for their own sake; the ranking is a deliberate transformation we apply because ordinal structure is more robust to the measurement noise than the cardinal values themselves.

We distinguish:

1. **Elicited rankings:** Ordinal preferences directly observed. Classical PL domain. DPO (Rafailov et al., 2023) lives here.
2. **Induced rankings:** Ordinal structure extracted from noisy cardinal measurements. The measurements were never intended as comparisons, but the ranking among them is more robust than the raw values. GPL on gene expression lives here.

This is, to our knowledge, a novel distinction. The existing literature on ranking models assumes rankings as given. The move from cardinal measurements to induced rankings — and the recognition that a generative ranking model can serve as a training objective for the induced case — extends PL to a much larger class of problems than the preference-modeling domain it was designed for.

Related but distinct prior concepts:

- *Rank statistics* (Wilcoxon, Mann-Whitney): use rankings nonparametrically for hypothesis testing — not generative.

- *Ordinal → cardinal recovery* (Thurstone, 1927): estimate latent cardinal values FROM rankings — the reverse direction.

C.4 Independent Convergent Evidence — Depth Estimation

The principle of training on induced rankings from noisy cardinal data has independent precedent in computer vision. Lienen and Hüllermeier (2021) showed that Plackett-Luce depth estimation outperforms regression on noisy depth measurements:

Lienen et al. (depth estimation)	**This work (gene expression)**
Pixel depth values are noisy cardinal measurements	Gene counts are noisy cardinal measurements
Relative depth ordering more robust than absolute depth	Relative expression ranking more robust than absolute counts
PL model on depth rankings outperforms regression	GPL on expression rankings outperforms count reconstruction
Shift-invariant depth from ranking-only training	**Depth-invariant** expression from ranking-only training
Zero-shot transfer to unseen scenes	Zero-shot transfer to unseen biology (GRN, HPA)

Their key insight, in their words: depth prediction is "fundamentally an ordering problem rather than a regression task." Our key insight is the same, applied to a different domain: gene expression prediction is fundamentally an ordering problem rather than a count reconstruction task. Same mathematical framework (Plackett-Luce), completely different domain and scale. The convergence of two independent research groups on the same principle — induced rankings from noisy cardinal data — is evidence that the principle is general.

What distinguishes our setting: they use standard PL on a 2D spatial grid with pairwise sampling. We use the Geometric extension (GPL) on unordered multisets at genome scale (~36,000 items), with massive tied groups requiring Henderson's tie-handling machinery. The scale and tie structure of biological data necessitated the Geometric extension that depth estimation did not require.

D Evaluation pipeline and metric interpretation

D.1 The Evaluation Task

Pretraining evaluation is a masked prediction task. The model processes gene tokens autoregressively in descending expression order. At evaluation, a fraction of the sequence is masked—the model sees some genes as context and must predict the expression profile of the full cell.

Concretely, for each held-out cell:

1. **Input.** The cell's expressed genes are ordered by expression level (most expressed first) and presented as a token sequence. The model processes these tokens autoregressively.
2. **Masking.** At a specified masking rate (10%, 50%, or 90%), the model is allowed to see only the first (1 − mask_rate) fraction of the sequence as context. At 50% masking—the rate reported in the main text—the model sees the top half of expressed genes. At 90% masking (the hardest setting), the model sees only the top 10% of expressed genes.
3. **Prediction.** At the masking boundary (the position where the model has seen all context tokens), the model outputs a single continuous scalar (lambdit) for every gene in the full vocabulary (~36,000 genes)—not just the masked genes. This is a complete `d_vocab`-dimensional prediction vector with ~36,000 unique continuous values.
4. **Predicted gene ordering.** The ~36,000 lambdit values are argsorted to produce a predicted gene ordering. In our parameterization, lower lambdit = higher predicted expression, so the gene with the lowest lambdit is

predicted most expressed. This gives an ordered list of gene IDs: [most-expressed-gene, second-most, …, least-expressed-gene].

5. **Ground-truth ranking.** The cell's observed expression counts are converted to dense ranks via `compress_ranks_descending`: genes are sorted by count, and genes with the same count receive the same integer rank. This typically produces ~10–500 unique rank values—not 36,000—because many genes share the same count (e.g., ~2,000 genes at count = 1 all receive the same rank). Undetected genes receive rank 0.

6. **Rank assignment to predictions.** This is the critical step. The ground-truth rank values are sorted in descending order and assigned to the model's predicted gene ordering: the gene the model predicts as most expressed receives the highest ground-truth rank, the second-predicted gene receives the second-highest rank, and so on. The result is a `d_vocab`-sized vector where each gene's value is a ground-truth rank—but placed according to the model's predicted ordering, not the true ordering.

Both the prediction vector and the ground-truth vector now contain the **same multiset of** ~500 **unique integer rank values**, just assigned to different gene positions. The question Spearman answers is: did the model assign each rank value to the correct genes?

In pseudocode:

```
# Ground truth: scatter true ranks into vocab-sized vector
gt_vector = zeros(36000)
for gene_id, rank in cell.expressed_genes:
    gt_vector[gene_id] = rank          # ~500 unique values + 30k zeros

# Prediction: model's ordering gets the same rank values
pred_vector = zeros(36000)
predicted_ordering = argsort(lambdits)  # model's predicted gene order
sorted_ranks = sort_descending(cell.ranks)
for i, gene_id in enumerate(predicted_ordering):
    pred_vector[gene_id] = sorted_ranks[i]  # same ~500 values, different positions

# Both vectors: same values, potentially different gene assignments
spearman = correlation(fractional_rank(gt_vector), fractional_rank(pred_vector))
```

7. **Spearman correlation.** Both vectors are converted to fractional ranks via the standard midrank convention—genes sharing the same integer rank value (e.g., the ~30,000 genes at rank 0, or the ~2,000 genes at rank 1) are assigned the midpoint of the positions they collectively occupy. Spearman correlation is then Pearson correlation on these fractional ranks.

This pipeline does not privilege the ranking objective. All four objectives (GPL, GC, PROP, MSE) produce a `d_vocab`-dimensional vector of continuous scalars at the masking boundary. These scalars are argsorted by the same operation regardless of which objective produced them. The evaluation pipeline has no knowledge of the training objective. GPL is not advantaged by a rank-based evaluation: it produces continuous lambdits that are argsorted identically to GC's lambdits, PROP's logits, or MSE's predicted values.

Masking rate and difficulty. At 10% masking, the model sees 90% of the cell's genes—a relatively easy interpolation task. At 90% masking, the model sees only the top 10% of expressed genes and must reconstruct the remaining 90%—a much harder task that relies more heavily on learned gene-gene relationships. GPL's advantage over count-based objectives grows at harder masking levels, consistent with the hypothesis that ranking-based pretraining captures deeper co-regulatory structure.

Held-out biology. All evaluations are computed on cells from tissue-disease groups held out entirely from

pretraining. The evaluation groups (`healthy-respiratory`, `other-unclear`) are zeroed out during training—the model has never seen a healthy lung cell or any cell from these groups during training.

D.2 Tie Structure and What Spearman Actually Measures

As described above, both the prediction vector and the ground-truth vector contain the same ~500 unique integer rank values by the time Spearman is computed. The tie structure—which rank values are shared by many genes—determines what the metric can and cannot distinguish.

In a typical single-cell RNA-seq observation:

- ~30,000 **genes** are not in the cell's expression profile and receive rank 0 in the `d_vocab`-sized ground-truth vector. These form one massive tied group.
- Among expressed genes, many share the same count value:
  - ~2,000–3,000 **genes** typically have count = 1
  - ~1,000–1,500 **genes** have count = 2
  - Numbers decrease at higher counts, with a handful of highly expressed genes at counts > 100

The result is that ground truth has approximately **10–500 unique rank values** (depending on sequencing depth), not 36,000. Each unique value defines a "tier," and all genes within a tier receive the same fractional rank under the midrank convention.

The midrank convention. For N genes sharing the same rank value and occupying positions p through $p+N-1$, each receives the fractional rank:

$$r = \frac{2p + N - 1}{2}$$

This is the standard treatment for tied ranks in Spearman correlation. For the ~30,000 undetected genes, all receive the same fractional rank—the midpoint of the bottom ~30,000 positions they collectively occupy.

Consequence for the evaluation. Because both vectors share the same ~500 unique rank values, Spearman effectively measures:

1. **Detection**: Did the model place expressed genes above undetected genes? Correctly separating the ~6,000 expressed genes from the ~30,000 zeros accounts for a large fraction of the correlation.
2. **Tier ordering**: Among expressed genes, did the model rank genes in the correct count-tier order? A gene with count = 5 should be ranked above a gene with count = 2.
3. **Within-tier ordering is invisible.** If 2,000 genes all have count = 1, they share the same ground-truth fractional rank. However the model orders these genes among themselves has no effect on Spearman correlation—the metric cannot distinguish correct from incorrect within-tier ordering, because the ground truth provides no within-tier information.

This means Spearman is asking a coarser question than it might first appear: not "did you get the full ordering of 36,000 genes right?" but rather "did you assign each rank tier to the correct genes, and did you separate expressed from unexpressed?"

D.3 Interpreting Absolute Spearman Values

The tier structure has direct implications for interpreting the absolute magnitude of Spearman correlations.

The ceiling is well below 1.0. Even a hypothetical perfect model—one that knows every gene's true expression rate—cannot achieve Spearman = 1.0, because the observed counts are a noisy sample from the true rates. A gene with true expression rate 2 might show count = 0 (dropout), count = 1, or count = 5 in any given cell due to Poisson sampling. The ground-truth ranking is itself a noisy measurement, and no model can correlate perfectly with noise.

The zero tier dominates. The ∼30,000 genes at rank 0 form the largest single block in the ground truth. Correctly predicting that most genes are unexpressed—a relatively easy task, since the model can learn marginal expression frequencies during pretraining—contributes substantially to the Spearman correlation. This inflates absolute values relative to what the "interesting" part of the task (ordering expressed genes) would yield alone.

The random baseline is ∼0. A completely random permutation of gene ranks scores approximately 0 on Spearman. The massive zero-tie block does not inflate the random baseline, because midranks assign all ∼30,000 zero-ranked genes the same fractional rank—shuffling which genes land in the zero bucket does not help if the ordering within the expressed set is random.

What 0.65 spearmanscipy means. The model's gene ordering across all ∼36,000 genes—including correctly predicting which ∼30,000 are undetected—correlates at 0.65 with the observed ground truth. This number is a composite of (a) successful detection (expressed vs. unexpressed), (b) correct tier ordering among expressed genes, and (c) irreducible noise in the ground truth. It is not directly comparable to Spearman correlations in other biological contexts (e.g., bulk RNA-seq replicates, where correlations >0.95 are routine) because the single-cell measurement has fundamentally different noise properties: massive zero-inflation, low counts per gene, and Poisson sampling.

The story is in the relative gaps and scaling trends. Because absolute Spearman values are a composite of signal and irreducible noise, and because the noise floor depends on sequencing depth and cell quality rather than model quality, comparisons between objectives at the same scale—under identical data, architecture, and compute—are far more informative than the absolute values themselves. A +0.039 gap between GPL and GC at 2B parameters (0.6681 vs 0.6291) represents a large shift in the model's ability to order the full transcriptome, even though both numbers "look similar" in absolute terms.

D.4 Metrics Reported

We report two metrics in the main text, each isolating a different component of model quality:

Full-vocabulary Spearman correlation (spearmanscipy). The primary metric. Computed over all ∼36,000 genes as described above. This is the hardest metric because it requires the model to simultaneously predict (a) which genes are expressed and (b) the relative ordering among expressed genes. It is the metric on which GPL's advantage over count-based objectives grows most dramatically with model scale.

Binary correctness. Given the model's predicted gene ordering, binary correctness asks: of the genes that are truly expressed (rank > 0 in ground truth), how many appear in the model's top-N predictions, where N is the number of expressed genes? This is a set-overlap measure—it checks whether the model places the right genes in the "expressed" set, independent of their ordering within that set. Binary correctness is the evaluation most favorable to GC, which explicitly models $P(\text{count} = 0)$ via a per-gene Bernoulli component at every training step. GPL never sees counts and never directly models the zero/nonzero boundary, making binary correctness a particularly stringent test of whether ranking-based pretraining recovers detection as an emergent property.

D.5 Quality Stratification

We report all pretraining metrics stratified by cell quality, defined as total UMI count (sequencing depth). Our primary evaluation uses cells with $\geq$ 4,096 total UMI (`qc_4096`). This threshold ensures that the ground-truth rankings are reliable: in a cell with 4,096+ UMI, the expressed genes have sufficient counts to produce meaningful tier structure, and the zero/nonzero boundary is reasonably well-determined.

Evaluating on shallowly-sequenced cells (e.g., <1,000 UMI) conflates model quality with ground-truth noise—a model may "fail" on such cells not because its predictions are wrong but because the ground truth is too noisy to evaluate against.

D.6 Implementation Details

Our reported Spearman metric uses the standard midrank (fractional rank) convention for tie-handling, equivalent to `scipy.stats.spearmanr`. No custom tie-breaking or reweighting is applied to any reported evaluation numbers.

E Perturbation prediction: extended results and scaling

E.1 Full metric comparison on Nadig-Replogle HepG2 held-out fold

Published results from Wang et al. 2026, Table B7 (380 held-out HepG2 perturbations, 2,000 HVGs). Our model (GPL $d = 1024$, 1B parameters) evaluated on matching split.

Table 7. HVG metric comparison on Nadig-Replogle HepG2 held-out fold (2,000 highly variable genes). Published baselines from Wang et al. (2026) Table B7. All published models train and evaluate on these 2K HVGs; our model trains on the full transcriptome (~9,600 genes) and is evaluated on the matching HVG subset.

Metric	C2S-2B	STATE	scGPT	X-Cell	Ours	Rank
MAE ↓	0.2845	0.0864	0.0804	0.0682	**0.0667**	**1st**
Pearson Δ ↑	0.0538	0.4563	0.1884	**0.5137**	0.4239	3rd
Centroid Accuracy ↑	0.5023	0.5767	0.6786	**0.7876**	0.7290	**2nd**
Overlap@N ↑	0.1432	0.2600	0.2013	**0.3053**	0.0875†	5th†
DE Direction Match ↑	0.6007	0.7564	0.6867	**0.8762**	0.6177†	4th†
DE Spearman LFC ↑	0.3369	0.5798	0.4298	**0.7745**	—*	—*
dPDS ↑	—	—	—	—	0.7262	—
Pearson E-distance ↑	—	—	—	—	0.6907	—

Table 8. Full transcriptome metrics on Nadig-Replogle HepG2 held-out fold (~9,600 genes). X-Cell, STATE, scGPT, and C2S predict only 2K HVGs and cannot be evaluated at this scope. Baselines: control mean (predicting the unperturbed average) and perturbation train mean (predicting the training-set average for each perturbation).

Metric	Ours	Ctrl mean	Pert train mean
MAE ↓	**0.0551**	0.0621	0.0594
Pearson Δ ↑	**0.3755**	—	—
dPDS ↑	**0.7084**	—	—
Centroid Accuracy ↑	**0.7290**	—	—
Pearson E-distance ↑	**0.6907**	—	—
Overlap@N ↑	0.0366†	—	—
DE Direction Match ↑	0.6197†	—	—

*DE Spearman LFC is omitted due to a bug in `cell-eval` (GitHub issue #227, fixed in PR #228) in which an inner join dropped genes the model did not predict as differentially expressed, inflating scores for all models. Wang et al. (2026) used the pre-fix version; our evaluation uses the corrected version (left join, missing predictions assigned fold-change = 0). The two are not comparable.

†Our model was evaluated using mean predictions (no per-cell sampling). This eliminates within-group variance, causing Mann-Whitney U tests to call all genes differentially expressed. DE-dependent metrics (Overlap@N,

Direction Match) are consequently deflated. Count-level sampling at inference recovers realistic variance and is expected to improve these metrics; we defer this to a subsequent analysis.

E.2 Methodological differences

All published models (X-Cell, STATE, scGPT, C2S) train and evaluate on 2,000 highly variable genes pre-selected by STATE's preprocessing pipeline. Our model trains on the full transcriptome (~9,600 genes) — a strictly harder task that allocates model capacity across 5× more genes. Our HVG MAE numbers are therefore conservative relative to HVG-only training.

X-Cell uses 32 fixed control cells for inference. We use 1,024+ control cells for pseudobulking. More controls yield more stable pseudobulk estimates, which may affect metric comparison.

E.3 Scaling line

Table 9. Perturbation prediction scaling: GPL-pretrained models evaluated on 378 held-out HepG2 perturbations (cell-eval v0.7.0, full profile). All non-DE metrics improve monotonically with scale.

Metric	d = 256 (50M)	d = 512 (250M)	d = 1024 (1B)
HVG (2K genes)			
MAE ↓	0.0712	0.0691	**0.0667**
Pearson Δ ↑	0.3826	0.3930	**0.4239**
dPDS ↑	0.6534	0.6972	**0.7262**
Centroid Accuracy ↑	0.6563	0.6974	**0.7290**
Pearson E-distance ↑	0.5793	0.5936	**0.6907**
Overlap@N† ↑	0.0905	0.0921	0.0875
DE Direction Match† ↑	0.5759	0.5932	**0.6177**
Full transcriptome (~9,600 genes)			
MAE ↓	0.0576	0.0565	**0.0551**
Pearson Δ ↑	0.3223	0.3459	**0.3755**
dPDS ↑	0.6333	0.6782	**0.7084**
Centroid Accuracy ↑	0.6474	0.6920	**0.7290**
Pearson E-distance ↑	0.5309	0.5355	**0.6907**
Overlap@N† ↑	0.0416	0.0383	0.0366
DE Direction Match† ↑	0.5838	0.6138	**0.6197**

Pretrained scaling law: MAE = $0.110 \times N^{-0.024}$ (R^2 = 0.995). X-Cell scaling exponent on MAE: α = 0.006 (Wang et al. 2026, Table A5), four-fold flatter—and X-Cell's best model plateaus at larger scales while ours continues to improve monotonically.

F Gene regulatory network inference: extended analysis

F.1 Benchmark: ENCODE ENETS2

F.1.1 Network Construction

We evaluate zero-shot gene regulatory network inference using the ENCODE Networks Proximal Filtered (ENETS2) benchmark (Gerstein et al., 2012, see also `http://encodenets.gersteinlab.org/`). ENETS2 was derived from 458 ChIP-seq experiments profiling 119 transcription factors across five ENCODE Tier 1/2 cell lines: K562 (chronic myelogenous leukemia), GM12878 (lymphoblastoid), HeLa-S3 (cervical carcinoma), HepG2 (hepatocellular carcinoma), and H1-hESC (embryonic stem cells). Peaks were called using MACS2 with IDR (Irreproducible Discovery Rate) thresholding across biological replicates, then integrated across cell types via union followed by probabilistic filtering. The resulting network represents a cell-type-agnostic regulatory grammar:

an edge indicates reproducible TF binding within 2.5 kb of the target gene's transcription start site (TSS) in at least one of the five cell lines.

Benchmark file: `enets2_proximal_filtered.txt`, downloaded from http://encodenets.gersteinlab.org/enets2.Proximal_filtered.txt.

Benchmark statistics:

- 26,070 edges total
- 115 TFs (96 sequence-specific, 19 chromatin regulators)
- 9,026 target genes
- Edges restricted to TF binding within ±2.5 kb of the TSS (proximal-filtered)

F.1.2 TF Classification: Sequence-Specific vs. Chromatin Regulators

We report two AUROC metrics throughout:

We exclude 17 factors classified as chromatin regulators, whose binding profiles reflect accessible chromatin state rather than sequence-specific DNA recognition. These factors bind broadly through protein–protein interactions with chromatin machinery or general transcriptional apparatus, and their ChIP-seq "targets" are enriched for open chromatin regions rather than direct, sequence-specific regulatory elements. The excluded factors are:

> CCNT2, CTCF, EP300, HDAC2, KAT2A, POLR3A, RAD21, SIN3A, SMARCA4, SMARCB1, SMARCC1, SMARCC2, SMC3, SUZ12, TAF1, TBP, TRIM28

This list includes the cohesin complex (RAD21, SMC3), SWI/SNF chromatin remodelers (SMARCA4, SMARCB1, SMARCC1, SMARCC2), the Polycomb component SUZ12, the insulator protein CTCF, the histone acetyltransferase EP300, the histone deacetylase HDAC2, the general transcription factors TBP and TAF1, and the KRAB-associated corepressor TRIM28. REST and NR2C2 were restored to the sequence-specific set: REST has a well-characterized RE1/NRSE binding motif (JASPAR MA0138), and NR2C2 (TR4) is a nuclear receptor with DR1 sequence-specific binding. Per-TF mean AUROC across 94 evaluable sequence-specific TFs is the primary comparison metric throughout.

F.1.3 Comparator Design

For each TF, positive instances are the TF's ENETS2 target genes and negative instances are genes targeted by *other* ENCODE TFs but not by the focal TF. This cross-TF design controls for the gene expression prior: all genes in the negative set are protein-coding, expressed, and subject to documented transcriptional regulation. This comparator is substantially more conservative than random genes, which would be trivially separated by expression level alone.

F.1.4 Why This Benchmark

This benchmark is appropriate for evaluating zero-shot GRN inference from transcriptome embeddings for four reasons. First, it uses high-confidence peaks (IDR-thresholded across biological replicates), minimizing false positive edges. Second, the proximal filter restricts edges to TF binding within ±2.5 kb of the TSS, capturing direct promoter-proximal regulation rather than distal enhancer contacts. Third, the aggregation across five cell types produces a cell-type-agnostic regulatory grammar, which is the appropriate ground truth for static gene embeddings learned from pan-tissue pretraining. Fourth, it has been used by prior work (Geneformer, scGPT), enabling direct methodological comparison.

F.2 Methods: Embedding Cosine Similarity

F.2.1 Zero-Order Probe

The embedding cosine approach is a zero-order probe that reads directly from the static input embedding matrix without performing a forward pass through the transformer:

$$\text{score}(\text{TF}, \text{gene}) = \text{cosine\_similarity}(\text{embedding}[\text{TF}],\ \text{embedding}[\text{gene}])$$

For each TF, we compute cosine similarity between the TF's learned token embedding and every other gene's token embedding. AUROC then measures whether this similarity score ranks true ENETS2 targets above the cross-TF negative set. Models are evaluated only on TF–target pairs where both the TF and target gene exist in the model's vocabulary.

F.2.2 Interpretation

AUROC depends on the *separation* between the target and non-target cosine similarity distributions, not on absolute cosine similarity values. A model could have universally high cosine similarities with poor AUROC (no separation), or low absolute similarities with excellent AUROC (clean separation). Our 250M model achieves a mean target–TF cosine similarity of +0.0106 versus a mean non-target–TF similarity of +0.0077—a gap of 0.0029 that is 37% larger than the corresponding gap for Tahoe-3B. This separation, not the absolute magnitude, drives the AUROC advantage.

F.2.3 Model Details

The primary embedding cosine results use the 250M-parameter model ($d_{\text{model}} = 512$), trained with the Geometric Plackett–Luce (GPL) ranking objective on single-cell RNA-seq data. The embedding matrix has dimensions 36,591×512, where 36,591 is the full gene vocabulary. Embeddings are the learned token embedding weights—no fine-tuning, no forward pass, no cell context is used.

F.3 Full Model Comparison

F.3.1 Embedding Cosine Leaderboard

Table S6.1. Per-TF mean AUROC on ENCODE ENETS2, evaluated using embedding cosine similarity on the common-gene subset shared across all model vocabularies. Corrected exclusion list: 17 chromatin regulators excluded (94 sequence-specific TFs evaluable with ≥5 targets on the common subset).

Model	Parameters	AUROC (per-TF mean)
GPL 250M	250M	**0.5727**
Tahoe 3B	3B	0.5621
scGPT	~50M	0.5592
Tahoe 1B	1B	0.5494
Geneformer 316M	316M	0.5488
Tahoe 70M	70M	0.5446
Geneformer 104M	104M	0.5403
Arc STATE SE 600M	600M	0.5229
Geneformer 10M	10M	0.5017

GPL 250M achieves the highest per-TF AUROC (0.5727), beating Tahoe-3B (0.5621) at 12× fewer parameters. Both metrics are per-TF means on the same common-gene subset, ensuring a fair comparison.

Scaling behavior diverges between models. Tahoe shows monotonic improvement under its masked language modeling objective (70M: 0.545, 1B: 0.549, 3B: 0.562). GPL shows non-monotonic behavior under ranking-

based pretraining (250M: 0.573, 1B and 2B lower), consistent with regulatory knowledge migrating from static embeddings into transformer layers at scale (§F.6).

Arc STATE SE 600M uses ESM2 protein language model embeddings (5,120-dimensional) projected through a learned encoder. Despite incorporating protein sequence information, it scores below all models trained purely on expression data (0.5229 vs. 0.5727 for GPL, 0.5621 for Tahoe-3B, 0.5592 for scGPT). Protein sequence homology does not transfer effectively to expression-based regulatory network prediction: the relevant signal for GRN inference lies in co-regulatory patterns across cellular contexts, not amino acid sequences.

Stack-Large cannot be evaluated using the embedding cosine protocol. Stack employs a tabular attention architecture (alternating cell-wise and gene-wise attention on cell × gene matrices) that does not produce static gene embeddings—there is no gene-to-embedding lookup table. Evaluating Stack on GRN would require a fundamentally different methodology (forward passes with per-gene inputs), which changes the comparison. Stack is excluded for methodological consistency.

F.3.2 Hidden Activation Probe Results

GPL 1B scores lower than GPL 250M under embedding cosine. This reflects probe choice, not model quality: GRN knowledge migrates to transformer layers at scale (Section F.6).

Table S6.2. Hidden activation probe, 1B model, best layer = 28, 20 K562 cells, 93 sequence-specific TFs with ≥10 targets.

Metric	Value
Mean AUROC	0.5779
Median AUROC	0.5637
Std AUROC	0.069
TFs with AUROC > 0.60	22
TFs with AUROC > 0.70	8
Best TF (STAT2)	0.838

The hidden activation result (0.5779) exceeds the 250M embedding cosine result (0.5727), confirming that the 1B model contains equal or greater GRN information when probed through the transformer layers.

F.4 STAT2 Case Study: Three Layers of Orthogonal Evidence

STAT2 achieves the highest per-TF AUROC in both the embedding cosine probe and the hidden activation probe (0.838 at layer 28). We present a detailed validation using three independent data modalities, each progressively further removed from the model's training data. STAT2 was selected for its well-characterized biology (JAK-STAT signaling, ISGF3 complex), extensive ENCODE ChIP-seq coverage in K562, and a known heterodimeric binding partner (STAT1) that enables a compositional test of complex structure.

F.4.1 Layer 1: ChIP-seq Peak Enrichment

Data. ENCODE IDR-thresholded optimal narrowPeak files for STAT2, STAT1, and GATA1 in K562, downloaded from the ENCODE REST API (https://www.encodeproject.org/). Gene promoter annotations: UCSC refGene, hg19 (GRCh37) assembly, TSS ±2.5 kb windows. 10,518 genes with both model cosine similarity scores (250M model embeddings) and promoter annotations. Background rate of STAT2 promoter binding: 2.1% (221/10,518 genes).

Table S6.5. STAT2 ChIP-seq enrichment among genes ranked by embedding cosine similarity to STAT2.

Controls.

Top-N genes	% with STAT2 peak	Background	Fold enrichment	p (Fisher's exact)
25	40.0%	2.1%	19.3×	3×10^{-11}
50	30.0%	2.1%	14.5×	—
100	20.0%	2.1%	9.65×	1.1×10^{-14}
500	8.6%	2.1%	4.2×	—
1,000	6.3%	2.1%	3.0×	—

- *STAT1 (positive control):* STAT1 heterodimerizes with STAT2 in the ISGF3 complex (Darnell et al., 1994). Top 100 genes ranked by STAT1 cosine similarity: 9.82× enrichment for STAT2 ChIP-seq peaks, $p = 6.6 \times 10^{-6}$. The model captures the functional overlap between JAK-STAT pathway components.
- *GATA1 (negative control):* GATA1 is a hematopoietic TF active in K562 (erythroid lineage) but regulates an entirely different gene program from STAT2 (interferon response). Top 100 genes ranked by GATA1 cosine similarity: no significant enrichment for STAT2 peaks ($p = 0.21$). This is despite GATA1 having approximately 14,605 K562 ChIP-seq peaks—roughly 7× more than STAT2. Peak abundance alone does not generate spurious enrichment; the signal is pathway-specific.

Continuous measures. Spearman $\rho = 0.089$ (cosine similarity vs. peak count at promoter), $p = 4.5 \times 10^{-20}$. Mann–Whitney U test (bound vs. unbound genes): $p = 2.4 \times 10^{-20}$.

The top STAT2-similar genes are canonical interferon-stimulated genes: STAT1, IRF2, IRF7, OAS2, OAS3, XAF1, IFI16, PML, TRIM56, SLFN5.

F.4.2 Layer 2: DNA Sequence Motif Scan

Rationale. Layer 1 relies on ENCODE peak calls, which involve MACS2 statistical thresholding. A skeptic could attribute the enrichment to biases in peak calling or ChIP-seq experimental design. Layer 2 eliminates this concern entirely by using only DNA sequence information—no ChIP-seq data, no peak caller, no signal processing.

Method. We scanned all 10,518 gene promoters (TSS ±2.5 kb, hg19, UCSC refGene) for the JASPAR MA0517.1 position frequency matrix (https://jaspar.genereg.net/), representing the STAT1::STAT2 heterodimer binding site—an ISRE-like motif, 15 bp in length. Both strands scored. A gene was classified as "motif-positive" if any position scored ≥80% of the PWM's maximum possible score (18.04 / 22.55). Background rate: 3.1% (325/10,518 genes).

Table S6.6. STAT1::STAT2 heterodimer motif (JASPAR MA0517.1) enrichment among genes ranked by STAT2 embedding cosine similarity.

Top-N genes	Motif present	%	Fold enrichment	p (Fisher's exact)
25	3/25	12.0%	3.88×	4.1×10^{-2}
50	5/50	10.0%	3.24×	1.9×10^{-2}
100	8/100	8.0%	2.59×	1.2×10^{-2}
200	15/200	7.5%	2.43×	1.4×10^{-3}
500	37/500	7.4%	2.39×	6.6×10^{-7}

Continuous tests. Mann–Whitney on max PWM score (top 100 vs. bottom 100 by cosine similarity): mean 13.76 vs. 12.09, $p = 1.84 \times 10^{-4}$. Spearman (cosine similarity vs. motif hit count, all 10,518 genes): $\rho = 0.035$, $p = 3.65 \times 10^{-4}$.

Top motif-positive genes: IRF7, OAS2, HERC6, IFIT2, EPSTI1, PHF11—the same ISGs identified by ChIP-seq in Layer 1.

Interpretation. The weaker fold enrichment relative to ChIP-seq (2.4× vs. 10×) is expected. Not all STAT2 binding occurs via this specific 15-bp motif (STAT2 also binds as part of ISGF3 at variant ISRE elements and

through non-canonical interactions). Conversely, many genes carry the motif by chance without functional STAT2 binding. Nevertheless, $p = 6.6 \times 10^{-7}$ at top 500 is robust. This result is entirely independent of ChIP-seq: no peaks, no peak caller, no experimental data—only the intersection of a known TF binding motif (JASPAR) with raw genomic sequence (hg19) and model-derived cosine similarity scores.

F.4.3 Layer 3: Heterodimer Complex Composition

Rationale. The JASPAR MA0517.1 motif represents the STAT1::STAT2 heterodimer binding site. STAT1 and STAT2 heterodimerize as part of the ISGF3 complex (with IRF9; Darnell et al., 1994) and bind cooperatively at ISREs. If the model has learned the complex structure, genes should need to be embedded near *both* STAT1 and STAT2 to carry the heterodimer motif.

Table S6.7. STAT1::STAT2 motif enrichment (top 100 genes) under different ranking strategies.

Ranking metric	Motif hits in top 100	%	Fold enrichment	p (Fisher's exact)
STAT2 alone	8/100	8.0%	2.59×	1.2×10^{-2}
STAT1 alone	4/100	4.0%	1.29×	0.37 (NS)
Average(STAT1, STAT2)	13/100	13.0%	4.21×	1.2×10^{-5}
Min(STAT1, STAT2)	**16/100**	**16.0%**	**5.18×**	$\mathbf{6.1 \times 10^{-8}}$

STAT1 alone shows no significant enrichment for the heterodimer motif ($p = 0.37$). Requiring proximity to *both* partners (min of cosine similarities) doubles the fold enrichment relative to STAT2 alone (5.18× vs. 2.59×) and improves statistical significance by six orders of magnitude.

Continuous binding signal also improves with the min ranking:

Ranking metric	Pearson r	Spearman ρ
STAT2 alone	0.388	0.464
STAT1 alone	0.412	0.417
Min(STAT1, STAT2)	**0.448**	**0.512**

Interpretation. The model learned that STAT1 and STAT2 form a functional complex. The heterodimer binding motif is enriched specifically among genes embedded near *both* subunits—not near either alone. This recovers the protein–protein interaction structure of the ISGF3 complex from RNA expression patterns, without protein interaction data, protein structure information, or DNA sequence.

F.4.4 GAS Motif Negative Control (STAT1 Homodimer)

STAT1 homodimers (GAF complex) bind the GAS element (JASPAR MA0137.3, 11 bp, consensus TTCNNNGAA)—a distinct motif from the heterodimer ISRE. Background prevalence: 22.2% (substantially more common than the 15-bp heterodimer motif at 3.1%).

Ranking metric	Fold enrichment	p
STAT1 alone	1.2×	0.15
STAT2 alone	1.0×	0.56
Min(STAT1, STAT2)	1.5×	0.005

No significant enrichment under single-TF rankings. The modest enrichment under the min ranking (1.5×, $p = 0.005$) likely reflects partial overlap between GAS- and ISRE-regulated genes in the interferon pathway. The GAS element is too short and degenerate (11 bp with 3 free positions) to produce clean enrichment signals via sequence scan, confirming that the heterodimer result is specific to the longer, more informative ISRE-like motif.

F.4.5 Dual-Peak Genes (Small-N Caveat)

Only 17 of 10,517 genes carry both STAT1 and STAT2 IDR peaks at their promoter (background: 0.16%). Under the min(STAT1, STAT2) ranking: 2 of the top 25 genes carry dual peaks (49.5× enrichment, $p = 7.2\times10^{-4}$). These genes are PML, IRF2, and NMI—established ISGs and ISGF3 targets. Correlations within this 17-gene set are noisy ($r \sim 0.3$–0.5, not significant at $n = 17$). We report this result with the appropriate small-sample caveat.

F.5 Why the Heterodimer Result Is Not "Just Co-Expression"

The expected skeptical objection is: "Interferon-stimulated genes are co-expressed. STAT1 and STAT2 are co-expressed in K562. Of course they cluster together." This objection does not survive scrutiny for three reasons.

First, STAT1 alone fails. If pathway co-expression were sufficient, STAT1-alone ranking should enrich for the heterodimer motif comparably to STAT2-alone. It does not ($p = 0.37$). The model has learned that heterodimer targets are specifically regulated by the *combination* of both factors. Co-expression within the interferon pathway is symmetric (STAT1 and STAT2 are equally interferon-associated), but the motif enrichment is asymmetric: STAT2-similar genes carry the heterodimer motif; STAT1-similar genes do not.

Second, the min ranking doubles the signal. Requiring proximity to both partners (min) produces 5.18× enrichment while STAT2 alone produces 2.59× and STAT1 alone produces 1.29×. This compositional structure cannot arise from undirected co-expression, which would predict that any interferon-associated TF produces comparable enrichment.

Third, the biological mechanism is specific and clear. STAT1 homodimers (GAF complex) and STAT1:STAT2 heterodimers (ISGF3 complex) bind different DNA motifs (GAS vs. ISRE) and regulate partially overlapping but distinct gene programs. Genes specifically responsive to the heterodimer—those with ISRE motifs in their promoters—are co-regulated with *both* STAT1 and STAT2 across diverse cellular contexts but not with either alone. Ranking-based pretraining observes this: across thousands of expression profiles representing different interferon signaling states, these genes consistently co-rank with both TFs simultaneously. The geometric consequence is that they sit in the region of embedding space close to both STAT1 and STAT2, which is precisely what the min(STAT1, STAT2) cosine similarity measures.

F.6 Knowledge Migration: Embedding Cosine vs. Hidden Activation at Scale

F.6.1 The Observation

Model	Parameters	Embedding Cosine AUROC_seq	Hidden Activation AUROC (mean)
GPL 250M	250M	**0.5727**	0.591*
GPL 1B	1B	0.5533	**0.5779**
GPL 2B	2B	0.5451	—

*250M hidden activation measured in earlier experiments using a different K562 h5ad source. The gap vs. 1B reflects dataset differences, not model quality.

The full three-scale comparison on the common gene set (Table S6.1) confirms non-monotonic scaling under embedding cosine: 250M (0.5727) > 1B > 2B.

F.6.2 Training Curve: Embedding Cosine Declines During Training

We benchmarked the 1B model at 14 checkpoints spanning training (batch 5,000 through batch 101,000):

ba5k	ba10k	ba20k	ba50k	ba90k	ba101k
0.5710	0.5629	0.5567	0.5523	0.5475	0.5530

Embedding cosine AUROC peaks at batch 5,000 (0.5710—close to the 250M final checkpoint value of 0.5727) and then monotonically declines. The same pattern was observed for the 1B model in earlier experiments: peak early, decline late.

F.6.3 Interpretation: Knowledge Redistribution

As transformer models scale and train longer, relational knowledge that initially resides in the static embedding matrix migrates into the transformer's attention heads and MLP layers. The embedding cosine probe, which bypasses the transformer entirely, becomes an increasingly poor readout. The hidden activation probe, which reads from transformer layers, recovers the knowledge.

This is consistent with findings in the NLP literature on probing-based analysis of BERT-family models (Tenney et al., 2019; Hewitt and Manning, 2019), where syntactic and semantic information concentrates in different layers as a function of model depth and training duration.

A KNN geodesic metric—computed on the same static gene embeddings but measuring local manifold distance (via a KNN graph) rather than global cosine similarity (cf. Pearce et al. 2025, who used the same approach to recover phylogenetic distances from Evo 2 DNA embeddings)—provides additional evidence. Replacing cosine similarity with KNN geodesic distance improves AUROC for the 1B model by +0.8% but *hurts* the 250M model by −2.2%. This is consistent with the hypothesis that larger models encode regulatory information in more complex, nonlinear geometric structures within the embedding manifold, rather than as simple linear directions that cosine similarity captures.

Practical implication. The GRN claim in the main text is *zero-shot capability*, not scaling. The 250M embedding cosine result demonstrates that a relatively small model encodes regulatory relationships as a static geometric property of its learned embedding space. The hidden activation results demonstrate that larger models contain equal or greater information, accessible through the transformer layers. Perturbation prediction (main text, Fig. 3) carries the scaling narrative.

F.7 BigWig Threshold Sensitivity Analysis

F.7.1 Rationale

The ENETS2 benchmark uses IDR-thresholded narrowPeak files, representing the top ~2% of genomic positions with statistically significant TF binding. A natural question is: how does model performance vary as a function of binding confidence? Does the model identify genes with any binding, or specifically genes with strong, functional binding?

F.7.2 Method

We obtained fold-change-over-control bigWig signal tracks from the ENCODE portal (`https://www.encodeproject.org/`) for K562 ChIP-seq experiments (hg19 assembly). For each gene, we computed the mean bigWig signal in a ±500 bp window around the TSS (hg19, UCSC refGene annotation). AUROC was computed at different fold-enrichment thresholds: a gene is "positive" if its TSS signal exceeds the threshold, "negative" otherwise.

F.7.3 STAT2 Threshold Curve (K562)

Table S6.8. AUROC at different binding strength thresholds for STAT2, K562 cell line.

BigWig Threshold	Biological Interpretation	*N* Positives	GPL 250M	Tahoe-3B	Winner
1.0 (all signal)	Background/noise	3,097	0.480	0.488	~Random
1.5	Moderate binding	354	0.538	0.539	~Tie
2.0 (strong)	**Functional binding**	**55**	**0.687**	**0.675**	**GPL**
2.5 (very strong)	**Core regulatory targets**	**28**	**0.814**	**0.779**	**GPL**

Both models are near random at the permissive threshold (fold-enrichment $\geq$ 1.0, encompassing ~40% of genes). Performance improves monotonically with stringency, reaching AUROC = 0.814 for GPL at the most stringent threshold. The model identifies genes with *strong, functional* binding, not weak or background-level ChIP-seq signal. GPL outperforms Tahoe-3B precisely at the operating point where biological significance is highest.

F.8 Data Sources and Reproducibility

F.8.1 ENCODE Networks (Benchmark)

- **Network:** ENETS2 proximal-filtered. Source: `http://encodenets.gersteinlab.org/enets2.Proximal_filtered.txt`.
- **Reference:** Gerstein et al. (2012).
- **Local file:** `enets2_proximal_filtered.txt` (26,070 edges, 115 TFs, 9,026 targets).

F.8.2 ENCODE ChIP-seq Peaks (STAT2 Case Study)

- **Source:** ENCODE REST API, `https://www.encodeproject.org/`.
- **File type:** `bed narrowPeak`, optimal IDR thresholded peaks.
- **Cell line:** K562 (ENCODE Tier 1).
- **TFs downloaded:** STAT2, STAT1, GATA1 (plus 62 K562 TFs for multi-TF analysis).
- **Assembly:** hg19 (GRCh37). Critical: ENCODE peak files for these experiments use hg19 coordinates; all gene annotations must match.

F.8.3 ENCODE BigWig Signal Tracks

- **Source:** ENCODE portal, `https://www.encodeproject.org/`.
- **File type:** `bigWig`, fold change over control.
- **Cell line:** K562.
- **TSS signal extraction:** Mean signal in ±500 bp window around TSS using pyBigWig.

F.8.4 Reference Genome and Gene Annotation

- **Genome:** hg19 (GRCh37).
- **TSS annotation:** UCSC refGene annotation for hg19 coordinates.
- **Promoter windows:** TSS ±2.5 kb for ENETS2 edge definition and motif scanning.

F.8.5 JASPAR Motif Matrices

- **STAT1::STAT2 heterodimer:** JASPAR MA0517.1, 15-bp position frequency matrix. Source: `https://jaspar.genereg.net/`. Scan threshold: ≥80% of PWM maximum score (18.04/22.55). Both strands.
- **STAT1 homodimer (GAS element):** JASPAR MA0137.3, 11-bp PFM (consensus TTCNNNGAA). Same scoring procedure.

F.8.6 Model Embeddings

- **GPL 250M:** d_{model} = 512. Gene vocabulary: 36,591.
- **GPL 1B:** d_{model} = 1,024.
- **Baseline models:** Tahoe (70M, 1B, 3B), Geneformer (10M, 104M, 316M), scGPT, Arc STATE SE 600M.

G Subcellular localization: extended analysis

G.1 LDA Methodology

We applied linear discriminant analysis (LDA) to pretrained gene embeddings as a probe for subcellular localization. LDA projects high-dimensional embeddings (512 dimensions for the 250M-parameter model) onto at most $k - 1$ discriminant axes, where k is the number of categories. With 20 valid HPA categories (those with ≥50 annotated genes), LDA produces 19 discriminant axes. We report results on the first four axes (LD1–LD4), which capture the largest between-class variance.

Cross-validation. Five-fold cross-validation was used throughout. Balanced accuracy is reported as the mean across held-out folds (12.3% ± 1.2% for the 250M model). No hyperparameter tuning was performed—LDA has no tunable hyperparameters beyond the number of components.

Shuffled control. To confirm that classification accuracy depends on the learned gene representations rather than LDA's flexibility, we permuted gene-to-embedding mappings—randomly reassigning each gene's embedding vector to a different gene—and repeated the 5-fold CV procedure. Shuffled accuracy drops to chance level (∼5%), confirming that the spatial signal resides in the pretrained embeddings.

Ground truth. Subcellular localization labels come from the Human Protein Atlas (HPA) v23, which assigns each protein-coding gene to one or more compartments based on immunofluorescence imaging. For genes with multiple annotations, we used the primary location (extracellular location if available, otherwise the first listed main location). We retained 20 categories with ≥50 annotated genes, covering 12,503 genes total.

G.2 Three-Modality Comparison: Protein, DNA, and RNA

To contextualize the expression-based localization signal, we ran the identical LDA pipeline on embeddings from three families of pretrained models spanning three biological modalities:

- **Protein sequence:** ESM2 (Lin et al. 2023) at 150M, 650M, 3B, and 15B parameters; ESM-C (Hayes et al. 2024) at 300M and 600M parameters. Embeddings extracted by mean-pooling the last hidden layer across all positions (fp16, max sequence length 1,022 amino acids for ESM2, 2,046 for ESM-C). For protein language models, the last hidden layer is the standard embedding extraction point (Rives et al. 2021). Source: UniProt human reference proteome (UP000005640), longest isoform per gene, 20,584 genes. 11.6% of proteins exceed ESM2's 1,022-token context and were truncated.
- **DNA coding sequence:** Evo 2 (Brixi et al. 2025) at 1B and 7B parameters. Embeddings extracted by mean-pooling an intermediate transformer block (`blocks.19` for the 1B model, `blocks.26` for the 7B model), following the Evo 2 official exon classification notebook (Brixi et al. 2025, GitHub). Evo 2 uses StripedHyena, a hybrid architecture with gated convolutions; the final layer is close to the language modeling head and produces representations optimized for next-token prediction rather than biological semantics. Intermediate layers capture richer biological features, analogous to findings in NLP (Hewitt & Manning 2019) and protein language models (Rao et al. 2019). Source: Ensembl GRCh38 release 113 coding sequences, longest CDS per gene, ambiguous bases (N) replaced with adenine, truncated at 8,192 nucleotides (Evo 2's context length), 19,881 genes. Only 1.1% of coding sequences exceed the 8,192 nt limit.
- **RNA expression:** Our pretrained models at 250M and 1B parameters (512 and 1,024 dimensions). Embeddings are the learned gene embedding vectors from the input layer.

Dimensionality correction. Embedding dimensions range from 512 (GPL-250M) to 5,120 (ESM2-15B). LDA estimates class-conditional covariance matrices; when embedding dimension approaches or exceeds the number of samples (∼12,700 genes), covariance estimation degrades. To ensure fair comparison, we applied StandardScaler followed by PCA to 256 components before LDA for all models. We report balanced accuracy throughout to account for severe class imbalance across the 21 HPA compartments (Nucleoplasm 27.8%, Cytosol 22.4%).

G.2.1 Results: balanced accuracy across modalities

Table 10. Subcellular localization balanced accuracy (5-fold CV, 21 HPA compartments). Chance level: 4.8%.

Model	Type	Params	Balanced Accuracy
Shuffled control	—	—	4.8% ± 0.2%
Pearson co-expression	—	—	9.6% ± 0.5%
GPL 250M (ours)	**RNA**	**250M**	**12.3**% ± **1.2**%
Evo2-1B	DNA	1B	16.4% ± 1.1%
Evo2-7B	DNA	7B	23.1% ± 0.9%
ESM2-150M	Protein	150M	23.1% ± 1.2%
ESM2-650M	Protein	650M	23.9% ± 1.2%
ESM-C 300M	Protein	300M	24.2% ± 0.7%
ESM-C 600M	Protein	600M	24.9% ± 1.0%
ESM2-3B	Protein	3B	25.3% ± 1.3%
ESM2-15B	Protein	15B	**26.7**% ± **1.4**%

G.2.2 Key findings

Protein models plateau at ∼27%. Across six protein sequence models spanning two architectures (ESM2, ESM-C) and two orders of magnitude in parameter count (150M to 15B), balanced accuracy ranges from 23.1% to 26.7%. Scaling is monotonic within ESM2 (23.1% → 23.9% → 25.3% → 26.7%) but shallow—a 100× increase in parameters yields only 3.6 percentage points.

DNA models scale strongly but remain below protein. Evo 2 CDS models improve from 16.4% (1B) to 23.1% (7B), matching the smallest protein model (ESM2-150M). This is expected: the localization signals (signal peptides, NLS, KDEL motifs, transmembrane helices) are encoded in the DNA coding sequence as codons, but the model must first learn the genetic code—the mapping from triplets to amino acids—before it can recognize these motifs. At 7B parameters, the model has largely learned this mapping and approaches protein-level performance.

RNA expression is competitive without any sequence information. Our 250M-parameter model achieves 12.3%—46% of the protein ceiling—from co-expression patterns alone. It has never seen a nucleotide, an amino acid, or a localization signal. The spatial information it recovers comes entirely from observing which genes covary across cellular contexts: genes that localize together are co-expressed together.

Information hierarchy. The three modalities form a coherent hierarchy: protein sequence (direct access to sorting signals) > DNA coding sequence (indirect, via genetic code) > RNA expression (no sequence access, learns from functional consequences). That all three modalities carry localization information—and that a purely observational modality recovers nearly half the signal of direct sequence access—indicates that subcellular localization is deeply coupled to transcriptional regulation.

G.2.3 The 1B expression model

The 1B expression model scores lower than the 250M model on embedding-based LDA. This parallels the GRN finding (§6.1): under ranking-based pretraining, larger models migrate regulatory and spatial knowledge from the embedding matrix into transformer layers. Embedding cosine similarity and LDA—which operate on the embedding layer alone—underestimate what larger models encode. Hidden-state probes at intermediate transformer layers would likely recover stronger localization signal at 1B scale.

G.3 The Ribosome Lifecycle Test — Extended

G.3.1 Cytoplasmic vs nucleolar ribosomes (LD1)

Nucleolar ribosome assembly factors and cytoplasmic ribosomal proteins perform sequential steps of the same biological process yet occupy different subcellular compartments.

Gene group	Genes	Function	Location	Mean LD1
Nucleolar assembly	NCL, FBL, NOP56, NOP58, DKC1, RRP1, UTP14A	Ribosome biogenesis	Nucleolus	+1.42
Cytoplasmic ribosomes	RPL3, RPL4, RPL5, RPL7, RPS3, RPS4X, RPS6, RPS8	Translation	Cytoplasm	+0.24

In raw embedding space: These two groups are 6× more similar to each other than to the genomic background (mean pairwise cosine similarity 0.04 vs 0.006 baseline). The embeddings reflect their functional relationship.

On LD1: They separate to opposite ends of the spatial axis. Cohen's $d = 2.72$, $p = 2.1 \times 10^{-6}$ (two-sample t-test). The spatial signal overrides the functional similarity.

G.3.2 Cytoplasmic vs mitochondrial ribosomes (LD2)

Gene group	Genes	Function	Location	Mean LD2
Cytoplasmic ribosomes	RPL3, RPL4, RPL5, RPL7, RPS3, RPS6, RPS8	Translation	Cytoplasm	+0.21
Mitochondrial ribosomes	MRPL1, MRPL2, MRPL3, MRPL4, MRPS5, MRPS7, MRPS9, MRPS10	Translation	Mitochondria	−4.07

Same molecular function—assembling proteins from mRNA—in different compartments. LD2 separates them by over 4 standard deviations.

Mitochondrial ribosomal proteins descend from the bacterial ancestor that became the mitochondrion through endosymbiosis approximately two billion years ago. Although now encoded in the nuclear genome, these genes retain distinctive expression signatures: they are co-regulated by nuclear respiratory factors (NRF1/2) and PGC-1α as part of the mitochondrial biogenesis program, depend on specialized import machinery (TIM/TOM complexes), and respond to mitochondrial stress signals (the mitochondrial unfolded protein response, UPR^{mt}) distinct from cytoplasmic stress pathways. LD2 captures this evolutionary boundary from expression patterns alone.