Differentiable Retrieval-Augmented Generation for Predicting Cellular Responses to Gene Perturbation

Andrea Giuseppe Di Francesco*
Sapienza University of Rome
ISTI-CNR, Pisa, Italy
difrancesco@diag.uniroma1.it

Andrea Rubbi*
University of Cambridge
Wellcome Sanger Institute
ar2232@cam.ac.uk

Rishabh Jain
University of Cambridge
rj412@cst.cam.ac.uk

Pietro Liò
University of Cambridge
pl219@cam.ac.uk

Abstract

Predicting transcriptional responses to genetic perturbations is fundamental to functional genomics and therapeutic discovery. Recent deep learning models have shown promise in single-cell perturbation response prediction, but they typically generate each response in isolation, without explicitly leveraging related perturbations. We introduce **PT-RAG** (**P**erturbation-aware **T**wo-stage **R**etrieval-**A**ugmented **G**eneration), a plug-in retrieval-and-conditioning module for generative cellular perturbation response. PT-RAG augments an existing perturbation-response backbone with learned access to related perturbation contexts. The key challenge is that relevance is not fixed in this setting: functionally related genes may elicit different effects across cell types. PT-RAG addresses this with a two-stage retrieval mechanism: GenePT-based semantic retrieval first identifies K candidate perturbations, after which a differentiable Gumbel-Softmax selector adaptively selects retrieved contexts conditioned on the control cell state, the query perturbation, and each candidate perturbation. We evaluate PT-RAG on two backbones, a STATE-style generator used as a frozen random reservoir and a fully trained scGPT, across cross-cell-type and cross-perturbation generalization tasks. PT-RAG consistently improves distributional similarity and often overall predictive quality; for example, on scGPT cross-cell-type results, the 2-Wasserstein distance drops by 5.9% relative to scGPT alone. The code to reproduce our experiments is available at https://github.com/difra100/PT-RAG_NIPS.

1 Introduction

Understanding how cells respond to genetic perturbations is a fundamental challenge in systems biology with profound implications for drug discovery, disease modelling, and gene therapy [Norman et al., 2019]. High-throughput Perturb-seq technologies now enable measurement of single-cell transcriptomic responses to thousands of genetic perturbations across diverse cell contexts [Norman et al., 2019, Replogle et al., 2022], generating rich datasets that capture the complex landscape of cellular responses. However, the combinatorial explosion of possible perturbations and cell contexts makes comprehensive experimental characterization infeasible, motivating the development of computational methods to predict perturbation responses *in silico*.

Recent deep learning methods span a broad range of approaches: latent-space perturbation models [Lotfollahi et al., 2019, 2023], optimal transport frameworks [Bunne et al., 2023], graph neural

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

networks for gene-gene interactions [Roohani et al., 2024], and transformer architectures that model cell populations as sequences [Adduri et al., 2025, Klein et al., 2025]. Despite their diversity, all of these methods share a critical architectural limitation: predictions are generated solely from the control cell state and perturbation identity, *without leveraging knowledge about related perturbations* already observed in the training data. This limitation is most acute in two practically important scenarios: predicting responses that are only partially observed during training, and predicting responses to perturbations that have never been experimentally tested.

In natural language processing, Retrieval-Augmented Generation (RAG) [Lewis et al., 2021] has emerged as a powerful paradigm for incorporating external knowledge into generation. In such settings, retrieval can rely on off-the-shelf text or image encoders [Song et al., 2020, Radford et al., 2021], and even non-differentiable retrieval yields strong results because the notion of "relevant context" is well-defined and task-aligned.

RAG beyond language models. Extending RAG to cellular biology is non-trivial for two reasons specific to this domain. (1) **No established retrieval metric**: there is no consensus on how to measure perturbation similarity. One-hot encodings carry no semantic information; embedding-based similarities (e.g., via GenePT [Chen and Zou, 2023]) capture functional descriptions but not cellular effects. (2) **Cell-type agnosticism**: standard retrieval depends only on the query perturbation, providing identical context regardless of whether predictions are made for T-cells, hepatocytes, or leukemia cells, though the same perturbation can have qualitatively different effects across these contexts. These challenges raise a critical question: *can RAG improve generation when the retrieval objective must itself be learned?*

Our contributions.

- We introduce **PT-RAG** (Perturbation-aware Two-stage Retrieval-Augmented Generation), the first differentiable RAG framework for cellular perturbation response prediction. PT-RAG uses a two-stage pipeline that first retrieves K candidate perturbations via cosine similarity in GenePT embedding space, then applies Gumbel-Softmax [Jang et al., 2017, Maddison et al., 2017] selection conditioned on both the target cell state and perturbation identity, enabling end-to-end optimization of what context benefits generation. We demonstrate that naïve retrieval *actively harms* performance, underscoring that differentiability and cell-type-aware conditioning are both necessary.
- We demonstrate that PT-RAG is a **plug-and-play** retrieval module rather than a backbone-specific architecture: in addition to the STATE family, we provide cross-cell-type transfer experiments on the scGPT perturbation generator [Cui et al., 2023] (see Section 5.1.1).
- We introduce a **cross-perturbation generalisation** protocol—a challenging split in which 60% of perturbations are held out for testing, that tests the most practically relevant scenario: predicting responses to genes never experimentally perturbed. Together with cross-cell-type generalisation, the two protocols provide complementary evidence for when and why retrieval helps in cellular biology.
- We provide quantitative evidence that PT-RAG learns cell-type-specific retrieval patterns ($\sim$19% perturbation overlap across cell types for the same query gene via Jaccard similarity analysis), and deliver statistically rigorous comparisons across both evaluation settings.

2 Related Works

Perturbation response prediction. Existing perturbation prediction methods [Lotfollahi et al., 2019, 2023, Bunne et al., 2023, Roohani et al., 2024, Adduri et al., 2025, Klein et al., 2025] generate responses from the cell state and perturbation identity, but do not retrieve knowledge from related perturbations. Namely, scGen [Lotfollahi et al., 2019] and CPA [Lotfollahi et al., 2023] learn perturbation effects in latent space, CellOT [Bunne et al., 2023] frames prediction as neural optimal transport, STATE [Adduri et al., 2025] models cell populations as sequences with distributional losses, and CellFlow [Klein et al., 2025] adds conditional flow matching. scGPT [Cui et al., 2023] provides a strong transformer backbone that we extend with RAG, while GEARS [Roohani et al., 2024] uses a GO co-annotation graph to propagate perturbation effects. Our work extends the existing perturbation modelling paradigm from a generation-only perspective to a retrieval-augmented generation perspective, enabling contextual perturbation information to be used at prediction time. Extended comparisons appear in Appendix E.

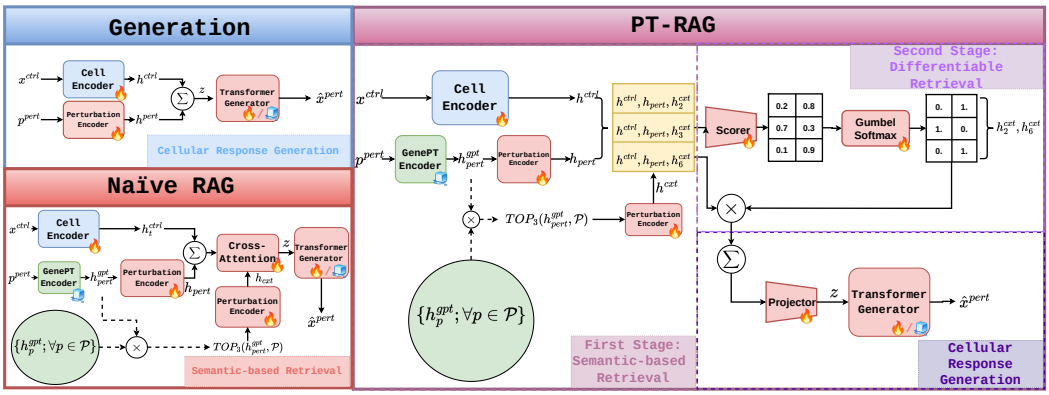


Figure 1: **Comparison of architectures.** *Left*: Generation baseline and Naïve RAG (non-differentiable retrieval, dotted lines; frozen components indicated by ice cube icon). *Right*: PT-RAG uses two-stage differentiable retrieval with Gumbel-Softmax selection conditioned on h^{ctrl}, h^{pert}, and candidate embeddings h_k^{cxt}. The Transformer Generator is frozen in the STATE-based experiments and trained in the scGPT-based ones.

Differentiable RAG. Standard RAG [Lewis et al., 2021] relies on fixed retrieval, while stochastic and differentiable variants [Zamani and Bendersky, 2024, Gao et al., 2025] learn retrieval decisions end to end. Stochastic RAG [Zamani and Bendersky, 2024] and D-RAG [Gao et al., 2025] show the value of optimized retrieval, but they assume pretrained generators and well-defined similarity metrics. PT-RAG instead learns relevance from scratch in a cell-type-dependent, non-textual generation setting.

RAG in biology. GeneRAG [Lin et al., 2024], scRAG [Yu et al., 2025], and agenticRAG for single-cell analysis [Nouri et al., 2025], all use retrieval to supply annotations or protein-centric context. None of these methods targets the generation of cellular response *distributions* conditioned on perturbation and cell state. PT-RAG bridges this gap by applying retrieval directly to perturbation-response generation in a learned, task-conditioned, cell-state-aware manner. Extended discussion appears in Appendix E.

3 Methods

We consider the task of predicting cellular responses to single-gene perturbations. Let $\mathcal{X} \subseteq \mathbb{R}^G$ be the space of gene expression profiles over G genes. Given a population of control cells $\{x_i^{ctrl}\}_{i=1}^N \subseteq \mathcal{X}$ and a perturbation $p^{pert} \in \mathcal{P}$ (where $\mathcal{P}$ is the set of perturbations), the goal is to predict the distribution of perturbed cells $\{\hat{x}_i^{pert}\}_{i=1}^N$.

In this section we describe three approaches in increasing order of complexity: Generation baseline, Naïve RAG, and PT-RAG. Figure 1 provides a visual comparison of all architectures.

3.1 Perturbation Representations with GenePT

Unlike prior work that uses one-hot perturbation encodings [Adduri et al., 2025], we leverage GenePT [Chen and Zou, 2023] embeddings that capture semantic gene relationships. For each gene g, GenePT provides $h_g^{gpt} \in \mathbb{R}^{d_{gpt}}$ derived from GPT-3.5 encodings of the gene's NCBI description [Brown et al., 2014] ($d_{gpt} = 1536$). We construct a perturbation database $\mathcal{D} = \{h_p^{gpt} \mid p \in \mathcal{P}\}$ for all $|\mathcal{P}| \approx 2009$ perturbations and normalize embeddings to unit norm before similarity computation.

3.2 Generation Baseline

Following STATE [Adduri et al., 2025], the Generation baseline uses:

$$h^{ctrl} = \text{CellEncoder}(x^{ctrl}), \qquad h^{pert} = \text{PertEncoder}(h_p^{gpt}), \tag{1}$$

combined as $z = h^{ctrl} + h^{pert}$ and decoded by a Transformer Generator: $\hat{x}^{pert} = \text{TransformerGenerator}(z)$.

The cell encoder is a trainable linear layer on the 2,000 HVG values. In the STATE experiments, the transformer generator and output projection are randomly initialised and frozen (reservoir [Shen et al., 2021]); PertEncoder is a trainable single-layer MLP. Predictions are made in isolation without leveraging related perturbation knowledge. This decomposition is intentionally backbone-agnostic: to attach PT-RAG, a generator must expose (i) a control-state representation h^{ctrl}, (ii) a perturbation representation h^{pert}, and (iii) a differentiable point at which the retrieved context can condition the prediction of $\hat{x}^{pert}$. The retrieval logic itself is unchanged across backbones; only a small backbone-specific adapter is needed to connect it. In the STATE-based models, the context enters at the cell level, as the generator input, whereas in our scGPT instantiation [Cui et al., 2023] it is added to the gene-token representations before the expression decoder (Appendix C.3).

3.3 Naïve RAG

Given a perturbation query h_p^{gpt}, Naïve RAG retrieves its top-K most similar perturbations via cosine similarity on GenePT embeddings: $\mathcal{R}_{p^{pert}} = \text{TOP}_K(h_p^{gpt}, \mathcal{D}) = \{p_{(1)}, \ldots, p_{(K)}\}$, and integrates them via Cross-Attention:

$$z = \text{CrossAttention}\big(q = h^{ctrl} + h^{pert},\ k = h^{cxt},\ v = h^{cxt}\big), \tag{2}$$

where $h^{cxt} \in \mathbb{R}^{K \times d_h}$ is the matrix of the retrieved embeddings encoded with PertEncoder. Retrieval is **non-differentiable** and depends *only* on h_p^{gpt}, providing identical context regardless of cell type.

3.4 PT-RAG: Perturbation-aware Two-stage Retrieval-Augmented Generation

PT-RAG addresses the limitations of Naïve RAG through a two-stage differentiable retrieval pipeline.

3.4.1 Stage 1: Semantic Candidate Retrieval

Identical to Naïve RAG, Stage 1 non-differentiably selects K candidates:

$$\mathcal{R}_{p^{pert}} = \text{TOP}_K(h_p^{gpt},\ \mathcal{D} \setminus \{h_p^{gpt}\}) = \{p_{(1)}, \ldots, p_{(K)}\}. \tag{3}$$

The query perturbation is excluded from the pool to prevent information leakage. This stage efficiently prunes the search space from $|\mathcal{P}|$ to K semantically relevant candidates.

3.4.2 Stage 2: Differentiable Cell-Type-Aware Selection

Each retrieved candidate $p_{(k)}$ is encoded as:

$$h_k^{cxt} = \text{PertEncoder}(h_{p_{(k)}}^{gpt}) \in \mathbb{R}^{d_h}, \quad k = 1, \ldots, K. \tag{4}$$

We form a **triplet** capturing the three-way relationship between cell state, target perturbation, and candidate:

$$c_k = [h^{ctrl};\ h^{pert};\ h_k^{cxt}] \in \mathbb{R}^{3d_h}, \tag{5}$$

and score each candidate with a two-layer MLP:

$$s_k = \text{MLP}_{\text{score}}(\text{LayerNorm}(c_k)) \in \mathbb{R}^2, \tag{6}$$

where s_k outputs `exclude` and `include` logits. We apply Straight-Through Gumbel-Softmax [Jang et al., 2017, Maddison et al., 2017] to make K **independent binary decisions** rather than a single selection among K candidates:

$$w_k = \text{GumbelSoftmax}(s_k,\ \tau)[\texttt{include}] \in \{0, 1\}, \quad k = 1, \ldots, K. \tag{7}$$

This allows multiple candidates to be selected simultaneously (or none). Selected contexts are projected and aggregated:

$$z = \sum_{k=1}^{K} w_k \cdot \text{MLP}_{\text{proj}}(c_k), \tag{8}$$

and decoded: $\hat{x}^{pert} = \text{TransformerGenerator}(z)$.

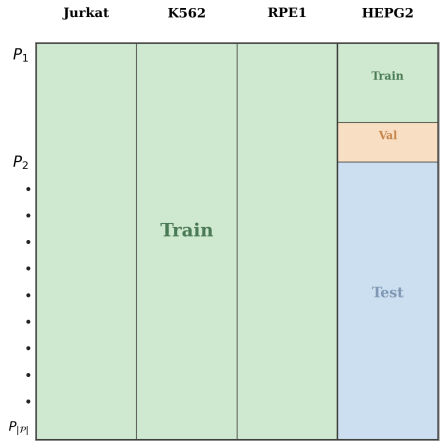


Figure 2: Cross-cell-type setting in Replogle-Nadig.

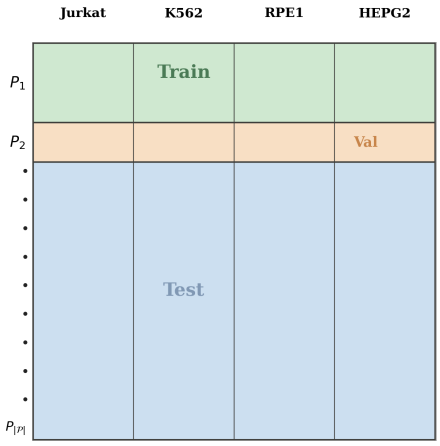


Figure 3: Cross-perturbation setting in Replogle-Nadig.

3.4.3 Training Objective

We train all STATE-based models with the following objective (scGPT-based models use a mean-squared-error reconstruction loss in place of $\mathcal{L}_{\text{dist}}$; Appendix C.3).

$$\mathcal{L} = \mathcal{L}_{\text{dist}} + \lambda_{\text{sparse}} \mathcal{L}_{\text{sparse}}, \tag{9}$$

where $\mathcal{L}_{\text{dist}} = \text{Energy}(\hat{x}^{pert}, x^{pert})$ is the energy distance [Rizzo and Székely, 2016] and $\mathcal{L}_{\text{sparse}} = \frac{1}{K}\sum_{k=1}^{K} w_k$ penalizes dense masks, preventing mode collapse to always-retrieve-all. The values of λ_{sparse} and K used in each setting are reported in Appendix C.2; sensitivity analyses in Appendix F.6 confirm robustness to this choice.

Table 1 summarizes the conceptual differences across all three methods.

Table 1: **Method comparison.** All methods except PT-RAG use non-differentiable or no retrieval; PT-RAG uses two-stage differentiable selection.

	Generation	**Naïve RAG**	**PT-RAG**
Retrieval	✗	✓	✓
Cell-type-aware Retrieval	✗	✗	✓

4 Experimental Setup

4.1 Dataset

We use the Replogle-Nadig Perturb-seq dataset and evaluate two complementary generalization settings: cross-cell-type and cross-perturbation. In the cross-cell-type setting, we hold out each cell type in turn: about two-thirds of its perturbations are used as few-shot examples during training, and the rest are held out for validation and testing. In the cross-perturbation setting, all cell types are visible, and the perturbations themselves are split into train, validation, and test sets, with 60% held out for testing and a dynamic retrieval pool that expands across stages.

Figures 2 and 3 summarize the two settings. The full dataset description, split sizes, retrieval-pool protocol, and task details are provided in Appendix D.

4.2 Baselines and Models

We compare four approaches for the main STATE-based experiments: (1) **STATE** [Adduri et al., 2025]: one-hot perturbation encoding, no retrieval; (2) **STATE+GenePT**: GenePT embeddings re-

Table 2: **Cross-cell-type generalization.** Mean values across 1,635 test perturbations from four cell types. Significance vs. PT-RAG: † ($p_{\text{FDR}} < 0.01$), †† ($p_{\text{FDR}} < 0.05$), † † † ($p_{\text{FDR}} < 0.1$). Bold: best. Results with standard deviations in Table 6.

Metric	STATE	STATE+GenePT	Naïve RAG	PT-RAG
Gene-level expression correlations				
Pearson DEG ↑	$0.624^{\dagger}$	0.631	$0.396^{\dagger}$	**0.633**
Spearman DEG ↑	$0.403^{\dagger}$	0.411	$0.307^{\dagger}$	**0.412**
Expression reconstruction accuracy				
$\text{MSE}_{\text{PCA50}}$ ↓	8.43	8.42	$12.64^{\dagger}$	**8.39**
Distributional similarity (PCA space)				
W_1 ↓	$35.70^{\dagger}$	$35.53^{\dagger\dagger\dagger}$	$48.48^{\dagger}$	**35.41**
W_2 ↓	$646.1^{\dagger}$	$638.7^{\dagger\dagger}$	$1189.5^{\dagger}$	**633.7**
Energy ↓	$9.41^{\dagger\dagger\dagger}$	9.40	$14.18^{\dagger}$	**9.33**

place one-hot encodings, no retrieval; (3) **Naïve RAG**: top-$K = 32$ GenePT cosine retrieval, cross-attention integration, non-differentiable; (4) **PT-RAG** (ours): two-stage differentiable retrieval. In all STATE-based experiments, the generator is a frozen random reservoir (Section 3.2), whereas scGPT models are fully trained. Our STATE baselines therefore differ from the published STATE model and should not be read as a direct reproduction of it. To evaluate backbone transferability, we additionally run a cross-cell-type study with an scGPT generator [Cui et al., 2023] augmented by GenePT and PT-RAG (see Section 5.1.1). Full architecture and training details are in Appendix C. In the cross-perturbation setting, to avoid data leakage, the retrieval pool is stage-specific: $\mathcal{D}_{train}$ during training, $\mathcal{D}_{val}$ during validation, and all 2,009 perturbations at test time (excluding the query). We do not include the one-hot STATE baseline in this setting because one-hot perturbation codes are inherently transductive and cannot represent unseen perturbations in a cold-start regime. We also exclude Naïve RAG and compare PT-RAG against the strongest non-retrieval transferable baseline (STATE+GenePT).

4.3 Evaluation Metrics

According to common practices to assess gene expression generation quality [Rubbi et al., 2026], we use the following metrics (definitions in Appendix B): (1) **Gene-level expression correlations** (Pearson, Spearman on differentially expressed genes [DEGs]); (2) **Expression reconstruction accuracy** ($\text{MSE}_{\text{PCA50}}$); (3) **Distributional similarity in PCA space** (W_1, W_2, Energy distance). Statistical significance is assessed via the Mann-Whitney U test with Benjamini-Hochberg FDR correction (details in Appendix G).

5 Results

5.1 Cross-Cell-Type Generalization

Table 2 presents results across 1,635 test perturbations.

Naïve RAG actively hurts performance. Despite incorporating retrieved context, Naïve RAG substantially underperforms STATE across all metrics (Pearson: 0.396 vs. 0.624; W_2: 1189.5 vs. 646.1). This failure is a primary finding of this paper: in this domain, retrieval without differentiable, cell-type-aware selection is not merely unhelpful—it is actively harmful. Ablations varying $K \in \{2, 5, 10, 32\}$ (Figures 10–11) confirm that this failure is not due to insufficient retrieval breadth: even $K = 32$ Naïve RAG falls far short of STATE, with non-monotonic performance curves suggesting that larger pools introduce more noise rather than more signal.

STATE+GenePT shows consistent but modest gains. Replacing one-hot encodings with GenePT embeddings yields small but consistent improvements over STATE (Pearson: 0.631 vs. 0.624), demonstrating the value of semantic gene representations. The remaining gap to PT-RAG is concentrated in distributional metrics (W_1, W_2), suggesting that cell-type-specific context selection adds population-level fidelity beyond what semantic embeddings alone provide.

Table 3: **Cross-cell-type generalization: Results on scGPT.** Mean values across 1,635 test perturbations from four cell types. Significance vs. PT-RAG: † ($p_{\mathrm{FDR}} < 0.01$), †† ($p_{\mathrm{FDR}} < 0.05$), † † † ($p_{\mathrm{FDR}} < 0.1$). Bold: best. Results with standard deviations in Table 7.

Metric	scGPT	scGPT+GenePT	scGPT+PT-RAG
Gene-level expression correlations			
Pearson DEG ↑	**0.9209**	0.9167	0.9128
Spearman DEG ↑	0.9500 †	**0.9509** †	0.9368
Expression reconstruction accuracy			
MSE_{PCA50} ↓	18.6440 †	19.8907 †	**17.3212**
Distributional similarity (PCA space)			
W_1 ↓	33.7992 †	34.6746 †	**32.7828**
W_2 ↓	578.6169 †	609.7348 †	**544.6977**
Energy ↓	23.5341 †	24.3923 †	**22.3410**

PT-RAG. PT-RAG demonstrates statistically significant improvements over STATE in gene-level correlations (Pearson: $p_{\mathrm{FDR}} = 2.4\times10^{-8}$; Spearman: $p_{\mathrm{FDR}} = 4.9\times10^{-10}$), and distributional similarity (W_1: $p_{\mathrm{FDR}} = 3.4\times10^{-6}$; W_2: $p_{\mathrm{FDR}} = 5.5\times10^{-8}$). Improvements over STATE+GenePT are more modest and concentrated in Wasserstein distances (W_2: $p_{\mathrm{FDR}} = 0.041$), consistent with the interpretation that retrieval adds cell-type-specific context selection on top of semantic encoding rather than semantic encoding itself. Notably, PT-RAG does not consistently outperform STATE+GenePT on K562 (Table 12), suggesting that retrieval benefits may be cell-type-dependent.

Overall, these results indicate that retrieved GenePT embeddings capture substantial functional relationships, and that end-to-end differentiable, cell-state-aware retrieval (PT-RAG) improves both gene-level correlations and distributional alignment relative to non-retrieval and naive retrieval baselines.

5.1.1 Transfer to scGPT backbone

The scGPT transfer section tests whether PT-RAG behaves as a backbone-agnostic retrieval module rather than a backbone-specific trick. The retrieval logic is unchanged: Stage 1 still retrieves GenePT candidates by cosine similarity and Stage 2 still makes cell-state-aware include/exclude decisions with Gumbel-Softmax; only the generator backbone changes. For readability in the main text, we report means only; full mean $\pm$ standard deviation values are provided in Appendix Table 7.

scGPT transfer: retrieval improves distributional fidelity while preserving backbone transferability. The scGPT results mirror the main STATE findings on the distributional side: PT-RAG is best on all three geometry metrics (W_1, W_2, Energy) and on reconstruction ($\mathrm{MSE}_{\mathrm{PCA50}}$), with statistically significant differences versus both baselines (Table 3). Relative to scGPT alone, PT-RAG reduces $\mathrm{MSE}_{\mathrm{PCA50}}$ by 7.1%, W_1 by 3.0%, W_2 by 5.9% (from 578.6 to 544.7) and Energy by 5.1%; relative to scGPT+GenePT, the W_2 reduction is 10.7%. These gains come with slightly lower gene-level correlations (Pearson -0.008, Spearman -0.013 versus scGPT; the Spearman difference is significant).

GenePT alone can degrade scGPT, while PT-RAG recovers and surpasses in most metrics. Compared with vanilla scGPT (one-hot perturbation encoding), scGPT+GenePT is worse on Pearson, $\mathrm{MSE}_{\mathrm{PCA50}}$, W_1, W_2, and Energy (with only a small Spearman gain). PT-RAG then recovers from this drop and outperforms both scGPT and scGPT+GenePT on reconstruction and all distributional metrics, at the cost of slightly lower gene-level correlations. This pattern supports the same conclusion as in STATE: semantic embeddings are useful, but they need cell-state-aware differentiable selection to consistently improve generation quality.

5.2 Cross-Perturbation Generalization

Table 4 reports results averaged across three folds. This setting is harder than cross-cell-type because 60% of all perturbations are unseen during training, so the model must recover from a much smaller direct supervision signal and rely more heavily on the retrieval pool. The table therefore serves as

Table 4: **Cross-perturbation generalization.** Results averaged across three folds. Significance vs. PT-RAG: † ($p_{\mathrm{FDR}} < 0.01$), †† ($p_{\mathrm{FDR}} < 0.05$), ††† ($p_{\mathrm{FDR}} < 0.1$). Bold: best. Results with standard deviations in Table 8.

Metric	STATE+GenePT	PT-RAG
Gene-level expression correlations		
Pearson DEG ↑	**0.6377**	0.6314
Spearman DEG ↑	**0.4293**	0.4224
Expression reconstruction accuracy		
$\mathrm{MSE}_{\mathrm{PCA50}}$ ↓	$1.1477^{\dagger}$	**1.1316**
Distributional similarity (PCA space)		
W_1 ↓	$0.8043^{\dagger}$	**0.8016**
W_2 ↓	$0.6436^{\dagger}$	**0.6367**
Energy ↓	$0.3385^{\dagger}$	**0.3357**

a stress test of whether PT-RAG still improves the geometry of the predicted response distribution when the perturbation split is much more severe. For readability in the main text, we report fold means only; full mean ± standard deviation values are provided in Appendix Table 8.

Cross-perturbation: PT-RAG consistently improves most metrics, with significant gains on reconstruction and distributional geometry. This setting is more challenging than cross-cell-type because 60% of perturbations are never seen during training. In this regime, PT-RAG improves 4/6 reported metrics versus STATE+GenePT, and these gains are statistically significant on $\mathrm{MSE}_{\mathrm{PCA50}}$, W_1, W_2, and Energy (all marked † in Table 4). The strongest pattern is again distributional: PT-RAG yields lower Wasserstein and Energy distances, indicating better recovery of population-level response structure for unseen genes.

Correlation behavior is complementary, not contradictory. STATE+GenePT remains slightly higher on Pearson and Spearman DEG, while PT-RAG is better on reconstruction and distributional metrics. This is consistent with retrieval acting primarily as a geometric/population-level correction mechanism rather than a pure rank-correlation optimizer.

Why one-hot baselines are excluded here. A one-hot perturbation representation cannot encode perturbations absent from training: for unseen genes, the model has no transferable perturbation embedding and therefore no meaningful cold-start initialization. This is exactly why cross-perturbation evaluation is centered on transferable representations (GenePT) and why PT-RAG is evaluated against STATE+GenePT in this setting.

The dynamic retrieval pool is a distinctive feature of this setting. At test time, the model has access to all the 2,009 characterized perturbations (excluding the query), whereas during training only the train split is retrievable. A control in which the test-time pool is restricted to the training perturbations gives essentially the same results (Appendix F.7), so the gains are not explained by the larger pool available at test time.

5.3 Generalization to a Different Perturbation Modality

The experiments above use a single dataset family based on CRISPRi knockdown. To test whether PT-RAG transfers to a different perturbation modality, we evaluate it on the Perturb-seq dataset of Norman et al. [2019], in which genes are *activated* with CRISPRa in `K562` cells. Since the dataset contains a single cell type, only the cross-perturbation setting applies: the 105 single-gene perturbations are split into 32 train, 10 validation and 63 test perturbations, with the same stage-dependent retrieval pool as in Section D.3. We use the STATE-based reservoir setting and report one fold; details are in Appendix D.4.

PT-RAG improves all metrics on Norman. Table 5 shows that PT-RAG outperforms STATE+GenePT on every reported metric, and does so in each of the 63 test perturbations. In contrast to Replogle-Nadig, where the gains are concentrated in distributional metrics, the largest improvements here are in gene-level correlations (Pearson $+0.087$, Spearman $+0.080$), while $\mathrm{MSE}_{\mathrm{PCA50}}$, W_1, W_2 and Energy decrease by 2.1–3.3%. The differences are statistically significant on all metrics except $\mathrm{MSE}_{\mathrm{PCA50}}$. This suggests that the lower DEG correlations observed in

Table 5: **Cross-perturbation generalization on Norman et al. (CRISPRa).** Mean values across the 63 test perturbations of one fold. Significance vs. PT-RAG: † ($p_{\text{FDR}} < 0.01$), †† ($p_{\text{FDR}} < 0.05$). **Bold**: best.

Metric	STATE+GenePT	PT-RAG
Gene-level expression correlations		
Pearson DEG ↑	0.2985†	**0.3856**
Spearman DEG ↑	0.2760†	**0.3564**
Expression reconstruction accuracy		
$\text{MSE}_{\text{PCA50}}$ ↓	5.4482	**5.2871**
Distributional similarity (PCA space)		
W_1 ↓	1.3706†	**1.3339**
W_2 ↓	3.0658††	**2.9649**
Energy ↓	0.7985††	**0.7815**

the Replogle-Nadig cross-perturbation setting are specific to that setting rather than intrinsic to retrieval. We note that this experiment uses a small perturbation set and a single fold, so it should be read as supporting evidence rather than as a full replication.

5.4 Cell-Type-Specific Retrieval Patterns

To validate that PT-RAG's differentiable retrieval genuinely conditions on cellular context, we analyze the overlap in selected perturbations across cell types using Jaccard similarity.

For each of the 33 genes common to all four cell types in the cross-cell-type test set, we extract the top-10 most frequently selected perturbations by the Gumbel-Softmax mechanism in each cell type and compute pairwise Jaccard similarity. Figure 4 shows the resulting matrix.

Low overlap confirms cell-type-specific retrieval. Off-diagonal Jaccard values average 0.191, indicating that ∼19% of selected perturbations overlap between cell types for the same query gene. This low overlap confirms that PT-RAG actively conditions on cell state rather than relying solely on the query perturbation—a capability that Naïve RAG fundamentally lacks.

Biologically coherent selection. Despite variation across cell types, PT-RAG maintains functional coherence. For WARS (tryptophanyl-tRNA synthetase), it retrieves other aminoacyl-tRNA synthetases across all cell types, but with different specific members (Jurkat: EARS2, DARS, VARS; HepG2: SARS2, GART, TARS; K562: FARSB, KARS, FARS2; RPE1: KARS, GART, TARS, QARS). This is consistent with known cell-type-specific differences in protein synthesis demands, though we note these are suggestive observations rather than rigorous biological claims. Additional retrieval examples appear in Appendix F.1.

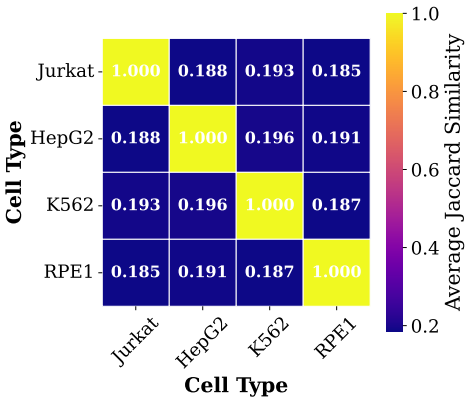


Figure 4: Jaccard similarity of top-10 retrieved perturbations across cell types. Off-diagonal values (range 0.185–0.196, mean 0.191) confirm that only ∼19% of selected perturbations overlap between cell types for the same query gene.

6 Discussion

Cross-cell-type results. Table 2 gives the cleanest picture in the paper: Naïve RAG is consistently worse than the no-retrieval baseline, STATE+GenePT gives a small improvement over STATE, and PT-RAG is best on every metric reported there. These results suggest that retrieval helps when it is learned end to end and when the selector can choose contexts that match the current cell state.

scGPT transfer. The scGPT transfer table shows two complementary facts. First, PT-RAG is not tied to one generator: the same two-stage retrieval policy transfers to scGPT without changing retrieval formulation. Second, the gains are concentrated where retrieval should matter most: reconstruction and distributional geometry. In particular, W_2 drops from 578.6 (scGPT) to 544.7 (scGPT+PT-RAG), with consistent improvements in W_1 and Energy as well. The table also highlights that replacing one-hot perturbation encoding with GenePT alone can reduce performance in this backbone; PT-RAG recovers this drop and outperforms both baselines on most metrics.

Cross-perturbation results. The cross-perturbation experiments provide complementary evidence: it tests retrieval in the strictest cold-start regime, where unseen perturbations dominate the test split. In this setting, the STATE one-hot baseline is not applicable, so comparison against STATE+GenePT isolates the added value of retrieval over the strongest transferable non-retrieval baseline. PT-RAG improves most reported metrics and is significant on reconstruction and all distributional rows, which is exactly the regime where context selection should help: preserving the geometry of unseen perturbed populations. The Norman experiment (Section 5.3) shows the same benefit under a different perturbation modality, with gains also in gene-level correlations.

Why the method works. The tables point to a consistent interpretation. GenePT already gives the model a useful semantic prior, but Naïve RAG shows that retrieval by itself is not enough; the retrieved contexts have to be filtered by a differentiable, cell-aware selector. PT-RAG improves the parts of the prediction problem that depend on population structure, which is why the retrieval module is best understood as a control mechanism over context rather than as a replacement for the generator.

Limitations. PT-RAG adds computation relative to the baselines because retrieval and selection are learned on top of the generator, so the method is not free. Our STATE-based experiments use a frozen, randomly initialised generator; we do not claim that the reported gains carry over to a fully trained generator, which we leave for future work. In the cross-perturbation setting, the retrieval pool at test time contains the embeddings of all perturbations, including those never seen in training; we regard this as part of the protocol rather than as an advantage of the method. Moreover, Stage 1 retrieval is cell-type agnostic, so a relevant perturbation that is not among the top-K GenePT neighbours cannot be recovered by Stage 2 (see the controls in Appendix F.7). The current study is also limited to single-gene perturbations, which keeps the biological setting focused but leaves combinatorial perturbations and other modalities that would require a completely new analysis on context modelling for future work. Finally, the retrieval examples are biologically suggestive rather than causal, so they should be read as model behavior rather than mechanistic proof.

7 Conclusion

We have presented PT-RAG as a retrieval module for perturbation biology that is both differentiable and reusable across backbones. The cross-cell-type results show the main claim most cleanly: learned retrieval beats Naïve RAG and improves on the retrieval-free baseline, with the strongest gains appearing where population geometry matters.

The scGPT experiment reinforces the plug-and-play framing, because the same retrieval rule can be attached to another generator without changing the retrieval logic. The cross-perturbation results then show how the same method behaves under a harder split in which direct perturbation supervision is much thinner.

That setting matters because it isolates whether retrieval can still help when the model has to lean more heavily on related contexts rather than on direct examples of the query perturbation.

Taken together, we conclude that PT-RAG is effective when retrieval is learned, cell-aware, and transferred as a module rather than baked into one backbone.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and the area chair for their constructive feedback and insightful discussions, which helped us improve this work. We thank Muhammad Shoaib Malik for an independent reproduction effort that helped us clarify the experimental setting.

Funding. A.G.D.F. was partially supported by the European Union's Horizon Europe Research and Innovation programme under the RenAI project (GA No. 101291378), and by the ENRESFOOD project, funded under the FutureFoodS partnership. The project is co-funded by the Austrian Science Fund (FWF; 10.55776/KIN4661525), the Dutch Ministry of Agriculture, Fisheries, Food Security and Nature (BO-43-219-026), the Italian Ministry of University and Research (MUR), the German Federal Ministry of Research, Technology and Space (031B1717), and the European Union's Horizon Europe research and innovation programme via FutureFoodS, the European Partnership for a Sustainable Future of Food Systems (GA No. 101136361).

Competing interests. The authors declare no competing interests.

References

Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-Yanai, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L. Bonnar, Marco Jost, Thomas M. Norman, and Jonathan S. Weissman. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28, 2022. doi: https://doi.org/10.1016/j.cell.2022.05.013. URL https://www.sciencedirect.com/science/article/pii/S0092867422005979.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Brahim, Junyue Cao, Fabian J Theis, et al. Compositional perturbation autoencoder for single-cell response modeling. *Molecular Systems Biology*, 19(6):e11517, 2023.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Saez-Rodriguez Del Castillo, Mitchell Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11): 1759–1768, 2023.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, 2024.

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, Chiara Ricci-Tam, Christopher Carpenter, Vishvak Subramanyam, Aidan Winters, Sravya Tirukovular, Jeremy Sullivan, Brian S. Plosky, Basak Eraslan, Nicholas D. Youngblut, Jure Leskovec, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Alexander Dobin, Dave P. Burke, Hani Goodarzi, and Yusuf H. Roohani. Predicting cellular responses to perturbation across diverse contexts with STATE. *bioRxiv*, 2025. doi: 10.1101/2025.06.26.661135. URL https://www.biorxiv.org/content/10.1101/2025.06.26.661135v1. Preprint.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrowskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, Nadezhda Azbukina, Fátima Sanchís-Calleja, Theo Uscidda, Artur Szalata, Manuel Gander, Aviv Regev, Barbara Treutlein, J. Gray Camp, and Fabian J. Theis. CellFlow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, 2025. doi: 10.1101/2025.04.11.648220. URL https://www.biorxiv.org/content/10.1101/2025.04.11.648220v1. Preprint.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks, 2021. URL https://arxiv.org/abs/2005.11401.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: Masked and permuted pre-training for language understanding, 2020. URL https://arxiv.org/abs/2004.09297.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Yiqun T. Chen and James Zou. GenePT: A simple but hard-to-beat foundation model for genes and cells built from ChatGPT. *bioRxiv*, 2023. doi: 10.1101/2023.10.16.562533. URL https://www.biorxiv.org/content/early/2023/10/19/2023.10.16.562533.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax, 2017. URL https://arxiv.org/abs/1611.01144.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables, 2017. URL https://arxiv.org/abs/1611.00712.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, 2023.

Hamed Zamani and Michael Bendersky. Stochastic RAG: End-to-end retrieval-augmented generation through expected utility maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pages 2641–2646. Association for Computing Machinery, 2024. doi: 10.1145/3626772.3657923. URL https://doi.org/10.1145/3626772.3657923.

Guangze Gao, Zixuan Li, Chunfeng Yuan, Jiawei Li, Wu Jianzhuo, Yuehao Zhang, Xiaolong Jin, Bing Li, and Weiming Hu. D-RAG: Differentiable retrieval-augmented generation for knowledge graph question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35398–35417. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1793. URL https://aclanthology.org/2025.emnlp-main.1793/.

Xinyi Lin, Gelei Deng, Yuekang Li, Jingquan Ge, Joshua Wing Kei Ho, and Yi Liu. GeneRAG: Enhancing large language models with gene-related task by retrieval-augmented generation. *bioRxiv*, 2024. doi: 10.1101/2024.06.24.600176. URL https://www.biorxiv.org/content/early/2024/06/28/2024.06.24.600176.

Zhiyin Yu, Chao Zheng, Chong Chen, Xian-Sheng Hua, and Xiao Luo. scRAG: Hybrid retrieval-augmented generation for LLM-based cross-tissue single-cell annotation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 954–970. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.53. URL https://aclanthology.org/2025.findings-acl.53/.

Nima Nouri, Ronen Artzi, and Virginia Savova. An agentic AI framework for ingestion and standardization of single-cell RNA-seq data analysis. *bioRxiv*, 2025. doi: 10.1101/2025.07.31.667880. URL https://www.biorxiv.org/content/early/2025/08/01/2025.07.31.667880.

Garth Brown, Vichet Hem, Kenneth Katz, Michael Ovetsky, Craig Wallin, Olga Ermolaeva, Igor Tolstoy, Tatiana Tatusova, Kim Pruitt, Donna Maglott, and Terence Murphy. Gene: A gene-centered information resource at NCBI. *Nucleic Acids Research*, 43, 2014. doi: 10.1093/nar/gku1055.

Sheng Shen, Alexei Baevski, Ari Morcos, Kurt Keutzer, Michael Auli, and Douwe Kiela. Reservoir transformers. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4294–4309, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.331. URL https://aclanthology.org/2021.acl-long.331/.

Maria L Rizzo and Gábor J Székely. Energy distance. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(1):27–38, 2016.

Andrea Rubbi, Andrea Giuseppe Di Francesco, Mohammad Lotfollahi, and Pietro Liò. A standardized framework for evaluating gene expression generative models, 2026. URL https://arxiv.org/abs/2603.11244.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. doi: 10.1023/A:1026543900054.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Ajay Nadig, Joseph M. Replogle, Angela N. Pogson, Steven A McCarroll, Jonathan S. Weissman, Elise B. Robinson, and Luke J. O'Connor. Transcriptome-wide characterization of genetic perturbations. *bioRxiv*, 2024. doi: 10.1101/2024.07.03.601903. URL https://www.biorxiv.org/content/early/2024/07/03/2024.07.03.601903.

Francesca Luchetti, A Gregorini, Stefano Papa, Sabrina Burattini, Barbara Canonico, Maria Valentini, and E Falcieri. The K562 chronic myeloid leukemia cell line undergoes apoptosis in response to interferon-α. *Haematologica*, 83:974–980, 1998.

Robert Abraham and Arthur Weiss. Jurkat T cells and development of the T-cell receptor paradigm. *Nature Reviews Immunology*, 4:301–308, 2004. doi: 10.1038/nri1330.

Elora Vanoni and Emeline Nandrot. The retinal pigment epithelium: Cells that know the beat! *Advances in Experimental Medicine and Biology*, 1415:539–545, 2023. doi: 10.1007/978-3-031-27681-1_79.

Stefania Moscato, Francesca Ronca, Daniela Campani, and Serena Danti. Poly(vinyl alcohol)/gelatin hydrogels cultured with HepG2 cells as a 3D model of hepatocellular carcinoma: A morphological study. *Journal of Functional Biomaterials*, 6:16–32, 2015. doi: 10.3390/jfb6010016.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017. URL https://arxiv.org/abs/1704.01212.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965. doi: 10.1093/biomet/52.3-4.591.

Nadim Nachar. The mann-whitney U: A test for assessing whether two independent samples come from the same distribution. *Tutorials in Quantitative Methods for Psychology*, 4, 2008. doi: 10.20982/tqmp.04.1.p013.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57:289–300, 1995. doi: 10.2307/2346101.

Appendix Overview

- **Appendix A**: Gumbel-Softmax mathematical background.
- **Appendix B**: Evaluation metric definitions.
- **Appendix C**: Full implementation and hyperparameter details.
- **Appendix D**: Description about datasets, and task, namely cross-cell-type and cross-perturbation, and the additional Norman dataset.
- **Appendix E**: Extended related work.
- **Appendix F**: Extended results: cell-type-specific retrieval examples, full tables with standard deviations, per-cell-type breakdowns, sensitivity analyses, additional controls, and computational costs.
- **Appendix G**: Statistical test details.
- **Appendix H**: LLM usage statement.

A Additional Preliminaries: Gumbel-Softmax

A.1 The Gumbel Distribution

The Gumbel$(0, 1)$ distribution has PDF:

$$f(x) = e^{-x-e^{-x}}, \tag{10}$$

illustrated in Figure 5.

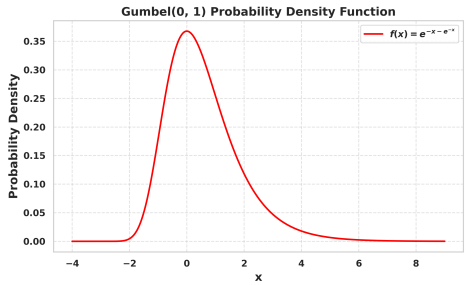


Figure 5: PDF of Gumbel$(0, 1)$: $f(x) = e^{-x-e^{-x}}$.

A.2 The Gumbel-Softmax Trick

Given logits $\pi = (\pi_1, \ldots, \pi_n)$, the Gumbel-Softmax [Jang et al., 2017, Maddison et al., 2017] provides a continuous, differentiable approximation to categorical sampling:

$$y_i = \frac{\exp((\log \pi_i + g_i)/\tau)}{\sum_j \exp((\log \pi_j + g_j)/\tau)}, \quad g_i \overset{\text{iid}}{\sim} \text{Gumbel}(0, 1), \tag{11}$$

where $\tau > 0$ is temperature. As $\tau \to 0$, the distribution approaches a one-hot (categorical) sample.

A.3 Straight-Through Estimator

For hard binary decisions, we use the Straight-Through Gumbel-Softmax estimator. Forward pass applies $\arg\max$ to obtain a hard one-hot; backward pass computes gradients through the soft probabilities:

$$\mathbf{y} = \mathbf{y}^{\text{hard}} + \big(\mathbf{y}^{\text{soft}} - \text{sg}(\mathbf{y}^{\text{soft}})\big), \tag{12}$$

where $\mathbf{y}^{\text{hard}} = \text{one_hot}(\arg\max_i y_i^{\text{soft}})$ and $\text{sg}(\cdot)$ is stop-gradient. This enables end-to-end training of models with discrete selection.

PT-RAG formulation. The Gumbel-Softmax is applied independently to each of the K candidates (binary include/exclude), not as a softmax over K candidates. This allows multiple candidates to be selected simultaneously or none at all, giving the scoring MLP full flexibility.

B Evaluation Metrics

B.1 Gene-Level Expression Correlations

For each test condition, DEGs are identified by comparing the observed perturbed cells with the control cells: a gene is called differentially expressed if its Welch's t-test p-value is at most 0.1, and its absolute $\log_2$ fold change is at least 0.05. Correlations are then computed on these genes:

- **Pearson correlation**: linear relationship between predicted and observed expression changes across DEGs.
- **Spearman correlation**: rank-order correlation of expression changes; robust to outliers.

B.2 Expression Reconstruction Accuracy

- $\mathrm{MSE}_{\mathrm{PCA50}}$: mean squared error in a 50-dimensional PCA space, fit on the observed expression profiles used for evaluation.

B.3 Distributional Similarity

Both predicted and true populations are projected to 50-dimensional PCA space before computing:

- **Wasserstein distances** W_1**,** W_2 [Rubner et al., 2000, Cuturi, 2013]: optimal transport distances, sensitive to both shape and support.
- **Energy distance** [Rizzo and Székely, 2016]: $\mathcal{E}(P, Q) = 2\mathbb{E}[\|X-Y\|] - \mathbb{E}[\|X-X'\|] - \mathbb{E}[\|Y-Y'\|]$; primary training objective.

C Implementation Details

C.1 Shared Architecture

All STATE-based models use a Llama-style [Touvron et al., 2023] transformer generator (8 layers, hidden size 128, 12 attention heads, feed-forward size 2,784) operating on cell populations of sequence length 64, with batch size 64.

Reservoir setting. In all STATE-based experiments the transformer generator and its linear output projection are *randomly initialised and kept frozen*; no pretrained weights are used. The generator therefore acts as a fixed random reservoir, in the spirit of reservoir computing and reservoir transformers [Shen et al., 2021]. Only the components around it are trained: the cell encoder, the perturbation encoder, the retrieval modules (where present), and the readout that maps the generator output to gene expression.

Cell Encoder. A trainable linear layer maps expression profiles over $G = 2000$ highly variable genes to 128-dimensional representations.

Perturbation Encoder. Trainable single-layer MLP: $\mathbb{R}^{1536} \to \mathbb{R}^{128}$.

Readout. A trainable MLP (hidden sizes 1024, 1024, 512) maps the generator output to the G expression values used for evaluation.

Parameter counts. Of the 20.97M parameters of PT-RAG, 5.21M are trainable and 15.76M are frozen (11.41M transformer layers, 0.26M output projection, and a 4.10M token-embedding table that is unused because inputs are provided as embeddings). The corresponding trainable counts are 5.18M (STATE), 5.11M (STATE+GenePT) and 5.36M (Naïve RAG); the frozen part is identical across methods.

C.2 PT-RAG-Specific Components

- **Scoring MLP** ($\mathrm{MLP}_{\mathrm{score}}$): two-layer network, hidden dim 128; input $[h^{ctrl}; h^{pert}; h_k^{cxt}] \in \mathbb{R}^{384}$; output $\mathbb{R}^2$.
- **Projection MLP** ($\mathrm{MLP}_{\mathrm{proj}}$): one-layer network, hidden dim 128.

- Gumbel temperature $\tau = 0.5$. Cross-cell-type (STATE): $K = 32$, $\lambda_{\text{sparse}} = 0.1$. Cross-perturbation (STATE): $K = 128$, $\lambda_{\text{sparse}} = 1.0$; the controls in Appendix F.7 use a model with $K = 32$. Norman: $K = 16$, $\lambda_{\text{sparse}} = 1.0$.

C.3 scGPT Backbone

To test transfer beyond STATE, we implement PT-RAG on top of an scGPT perturbation generator [Cui et al., 2023] (12 transformer layers, embedding size 512, 8 heads). Each cell is represented as a sequence of up to 512 gene tokens; every token combines a gene embedding, an encoding of the continuous control expression value, and a perturbation-flag embedding. A decoder predicts the perturbed expression of each gene from the transformer output.

Initialisation and training. Gene embeddings, the value encoder and the transformer are initialised from the public scGPT checkpoint where parameter names match; the attention projection weights do not match and are trained from scratch. *All* parameters are trained (no component is frozen), with learning rate 5×10^{-5}, batch size 64 and 25,000 steps. scGPT models are trained with a mean-squared-error reconstruction loss, to which PT-RAG adds the sparsity term of the training objective.

scGPT+GenePT. The GenePT embedding of the query perturbation is projected to the model dimension, $h^{pert} = \text{MLP}(h_p^{gpt}) \in \mathbb{R}^{512}$, and added to the transformer output at every gene position before decoding.

scGPT+PT-RAG. The retrieval module is the same as in the STATE instantiation. Stage 1 retrieves the top-K candidates by GenePT cosine similarity. For Stage 2, the control-cell representation h^{ctrl} is read from the transformer output (first sequence position), each candidate is encoded with the same GenePT projection, and the triplets $[h^{ctrl}; h^{pert}; h_k^{cxt}]$ are scored with a two-layer MLP followed by Straight-Through Gumbel-Softmax, as in the STATE instantiation. The selected projected triplets are summed and the resulting context vector is added to the transformer output at every gene position, on top of h^{pert}, before the expression decoder. Only the interface differs from STATE: the context conditions gene-token representations after the transformer, rather than replacing the generator input. We use $K = 128$ and $\lambda_{\text{sparse}} = 1.0$.

C.4 Training Configuration

- STATE-based models: Adam optimizer, learning rate 10^{-3}; up to 50,000 steps, validation every 2,000 steps.
- scGPT-based models: learning rate 5×10^{-5}, 25,000 steps (Appendix C.3).
- Hardware: NVIDIA A100 (80 GB). PT-RAG convergence: ≈8–10 hours per target cell type.

C.5 Cross-Perturbation Protocol Details

Folds are constructed by random partition of the 2,009 perturbations into 670 train / 133 validation / 1,206 test perturbations (approximately 33% / 7% / 60%). Three independent folds are generated; final results are averaged across folds.

GenePT embeddings of the perturbations that become available in the pool at validation and test time are used for retrieval indexing (Stage 1).

D Dataset and Tasks

D.1 Dataset

We evaluate on the Replogle-Nadig dataset [Replogle et al., 2022, Nadig et al., 2024], a large-scale Perturb-seq study with 643,413 single-cell transcriptomic profiles. The data span four human cell lines: `K562` (chronic myeloid leukemia) [Luchetti et al., 1998], `Jurkat` (T-cell lymphocyte) [Abraham and Weiss, 2004], `RPE1` (retinal pigment epithelial-1) [Vanoni and Nandrot, 2023], and `HepG2` (hepatocellular carcinoma) [Moscato et al., 2015]. The dataset contains 2,023 unique single-gene perturbations and a non-targeting control; we restrict retrieval and evaluation to the 2,009 perturbations with an available GenePT embedding. Not every perturbation is profiled in every cell line: the

number of perturbations per cell line ranges from 1,340 (HepG2) to 1,537 (Jurkat). Each cell is represented by $G = 2{,}000$ highly variable genes. An additional dataset used to test a different perturbation modality is described in Section D.4.

D.2 Evaluation Setting 1: Cross-Cell-Type Generalization

Each cell type serves as a held-out target in turn, using the split files released with STATE [Adduri et al., 2025]. Models are trained on all perturbation responses from the three remaining cell types. For the target cell type, about two thirds of its perturbations (900–1,014, depending on the cell type) are provided as few-shot examples during training; the remaining perturbations are held out, with 60–78 used for validation and 380–449 (about 28%) for testing. Perturbations held out in the target cell type remain available in the source cell types in which they were profiled. After restricting to perturbations with a GenePT embedding, the test sets contain 375 (HepG2), 401 (K562), 443 (Jurkat) and 416 (RPE1) perturbations, for a total of $n = 1{,}635$.

This setting tests generalization across *biological contexts*: given perturbation data from related cell types, can the model predict responses in a partially observed cell type? It is the primary setting for evaluating cell-type-awareness in PT-RAG.

D.3 Evaluation Setting 2: Cross-Perturbation Generalization

This setting tests generalization across the *perturbation space* itself: predicting cellular responses to perturbations never seen during training.

Protocol. All four cell types are always visible during training, validation, and testing, so no cell type is held out. The 2,009 perturbations are randomly split into **670 train**, **133 validation**, and **1,206 test** perturbations (approximately 33% / 7% / 60%). This ratio is challenging by design, simulating realistic experimental campaigns in which most genes have not yet been perturbed. Because a perturbation is evaluated separately in each cell line where it was profiled, each fold yields about 3,430 test cell-type–perturbation pairs.

The retrieval pool is dynamic and stage-dependent (the query perturbation is always excluded):

- **Training**: only train-split perturbations are available for retrieval ($|\mathcal{D}_{\text{train}}| = 669$).
- **Validation**: train and validation perturbations are available ($|\mathcal{D}_{\text{val}}| = 802$).
- **Testing**: all perturbations are available for retrieval, including test-split perturbations ($|\mathcal{D}_{\text{test}}| = 2{,}008$).

The pool contains the GenePT embeddings of these perturbations, not their measured expression responses. This dynamic pool reflects the scenario where retrieval can draw on the description of any gene of interest at inference time. Three independent folds are generated and results averaged.

The two task setups are visualized in Figures 2, 3 in the main text; here we keep the full protocol details.

Relationship to cross-cell-type setting. The two settings are complementary. Cross-cell-type asks: *can retrieval help when the perturbation is known but the cell type is only partially observed?* Cross-perturbation asks: *can retrieval help when the cell type is known but the perturbation is unseen?* The cross-perturbation setting is the more direct test of the value of retrieval: when a query perturbation has no direct training signal, the representations of functionally related perturbations seen during training may provide the main source of perturbation-specific information. It also represents a practically relevant experimental scenario: in discovery campaigns, one typically has transcriptomic data across cell types but wants to predict the effect of genes that have never been perturbed.

D.4 Additional Dataset: Norman et al. [2019]

To test PT-RAG beyond the Replogle-Nadig data, we use the Perturb-seq dataset of Norman et al. [2019], which differs in perturbation modality: genes are *activated* with CRISPRa, whereas Replogle-Nadig uses CRISPRi knockdown. The dataset contains a single cell line (K562). We keep the single-gene perturbations and the control cells, giving 69,570 cells and 105 single-gene perturbations. Expression counts are normalized to 10^4 counts per cell and log-transformed, and genes

are aligned to the gene set of the Replogle-Nadig data (6,350 of its 6,642 genes are present in the Norman data; missing genes are set to zero).

Since there is only one cell type, only the cross-perturbation setting applies. We use the same protocol as in Section D.3: the 105 perturbations are randomly split into 32 train, 10 validation and 63 test perturbations (approximately 30% / 10% / 60%). One perturbation has no GenePT embedding and is excluded from retrieval, so the retrieval pool contains 31 perturbations during training, 41 during validation and 104 at test time (the query is always excluded). Because the training pool is small, we use $K = 16$ candidates and $\lambda_{\text{sparse}} = 1.0$, and train for 10,000 steps; all other settings are as in the Replogle-Nadig experiments. Results are reported for one fold.

E Extended Related Work

E.1 Single-Cell Perturbation Response Modeling

Computational prediction of single-cell perturbation responses has advanced rapidly in recent years. scGen [Lotfollahi et al., 2019] learns additive perturbation vectors in a VAE latent space, while CPA [Lotfollahi et al., 2023] introduces compositional modeling with covariate disentanglement. CellOT [Bunne et al., 2023] casts perturbation response prediction as neural optimal transport between control and perturbed distributions, and STATE [Adduri et al., 2025] models cell populations as sequences with distributional losses such as energy distance and Sinkhorn divergence. CellFlow [Klein et al., 2025] extends this family with conditional flow matching for trajectory prediction. scGPT [Cui et al., 2023] provides a strong transformer backbone for single-cell modeling and is a natural candidate to extend with RAG for perturbation prediction, which is exactly the transfer setting studied in the main text.

A complementary line of work leverages gene-gene interaction graphs. GEARS [Roohani et al., 2024] constructs gene regulatory graphs from GO co-annotation with Jaccard-weighted edges and uses graph neural networks via neural message passing [Gilmer et al., 2017] to propagate perturbation effects, enabling combinatorial knockout prediction.

Across these methods, the shared limitation is that predictions are generated from cell state and perturbation identity alone, without explicitly retrieving knowledge from related perturbations. Our framework extends STATE [Adduri et al., 2025] and scGPT [Cui et al., 2023] with retrieval-augmented generation so contextual perturbation information can be used at prediction time rather than compressed into a direct perturbation representation. This makes PT-RAG a retrieval module for perturbation biology rather than a replacement for the underlying generator.

E.2 Retrieval-Augmented Generation

RAG [Lewis et al., 2021] combines parametric knowledge stored in model weights with non-parametric knowledge retrieved from external sources. In its standard form, retrieval is fixed and query-based: the generator is conditioned on retrieved content, but the retrieval step itself is not learned end to end. Stochastic RAG [Zamani and Bendersky, 2024] and D-RAG [Gao et al., 2025] show that optimized retrieval can improve performance, but they assume pretrained generators and well-defined similarity metrics. PT-RAG instead learns relevance from scratch in a cell-type-dependent, non-textual generation setting where both the retrieved content and the output are distributions over cells rather than text.

RAG has also expanded beyond text to images, video, audio, and graphs, showing that the retrieval principle is not inherently language-bound. The key difference in our setting is that the retrieved items are perturbation embeddings and the generator output is a predicted cellular population, which requires task-conditioned retrieval rather than lexical or document matching.

E.3 RAG in Biology

Within biology, most RAG systems retrieve literature, annotations, or structured context for downstream tasks. GeneRAG [Lin et al., 2024] and scRAG [Yu et al., 2025] retrieve textual context for biological question answering, and related systems such as agenticRAG for single-cell analysis and RetrievalAugmentedProteinEncoder focus on annotation-centric or protein-centric retrieval. These

approaches demonstrate that retrieval is useful in biology, but none addresses generation of cellular response distributions conditioned on perturbation and cell state.

The closest conceptual overlap with PT-RAG is retrieval-augmented encoding, where retrieved context is injected into a learned representation before prediction. That perspective helps explain why our method transfers cleanly to scGPT: the same retrieval policy can be attached to a transformer backbone as a plug-in module. PT-RAG is therefore the first RAG framework for cellular response modeling that jointly handles learned retrieval, cell-state conditioning, and non-textual generation.

Overall, the appendix literature supports the same conclusion as the main paper: existing perturbation-response models and existing biological RAG systems each solve part of the problem, but PT-RAG is the first to combine them in a differentiable, cell-aware retrieval-and-generation framework.

E.4 Differentiable Retrieval in Other Domains

Stochastic RAG [Zamani and Bendersky, 2024] formulates retrieval as sampling without replacement using Gumbel-Top-K, demonstrating benefits of end-to-end optimization on diverse NLP benchmarks. D-RAG [Gao et al., 2025] applies differentiable retrieval to knowledge graph QA, jointly optimizing a subgraph retriever and LLM-based generator via Gumbel-Softmax reparameterization. Both works operate in domains with pretrained generators and a relatively clear notion of relevance. PT-RAG adapts the differentiable retrieval idea to a domain where these assumptions fail: no pretrained retriever exists, relevant context is cell-type-dependent, and the generator produces cell distributions rather than text.

F Extended Results

F.1 Cell-Type-Specific Retrieval Examples

We present top-5 most frequently selected perturbations across cell types for five illustrative query genes.

WARS (Tryptophanyl-tRNA Synthetase). Jurkat: EARS2, DARS, SEPSECS, CTU2, VARS; HepG2: SARS2, GART, KARS, EARS2, TARS; K562: FARSB, KARS, FARS2, EPRS, DARS; RPE1: KARS, GART, TARS, QARS, AARS2. All are aminoacyl-tRNA synthetases, reflecting functional coherence with the query gene's role in translation; specific members vary by cell type, consistent with differential protein synthesis demands across T-cells, hepatocytes, myeloid, and epithelial contexts.

DDX27 (DEAD-Box RNA Helicase). Jurkat: DHX15, GEMIN4, DHX16, DDX19A, DDX23; HepG2: EDC4, DDX56, DHX36, GEMIN4, DDX1; K562: DDX11, DDX1, DDX6, DHX36, DHX33; RPE1: EDC4, DDX10, DDX56, DHX29, DDX19A. Selection of distinct DDX/DHX family members suggests cell-type-specific RNA processing pathway utilization; GEMIN4 is shared between Jurkat and HepG2, potentially reflecting shared splicing demands.

MRPL39 (Mitochondrial Ribosomal Protein L39). Jurkat: MRPL23, MRPL50, MRPL51, MRPL37, BRIP1; HepG2: RPL13, MRPL20, MRPL17, MRPL4, MRPL33; K562: MRPL36, MRPL20, MRPL51, MRPL54, MRPL4; RPE1: MRPL10, MRPL3, MRPL37, MRPL51, MRPL35. HepG2's inclusion of cytoplasmic RPL13 may reflect the high metabolic demands of hepatocytes coordinating mitochondrial and cytoplasmic translation.

RPS2 (Ribosomal Protein S2). Jurkat: RPS17, RPS28, RPL35, RPS3A, RPS15; HepG2: RPL35, RPS10, RPS16, RPS17, RPL4; K562: RPS24, RPL30, RPS17, RPS9, RPL4; RPE1: RPS3A, RPS5, RPS28, RPL9, RPS10. The mixture of large (RPL) and small (RPS) subunit proteins varies by cell type, potentially reflecting cell-type-specific ribosome stoichiometry.

TPRKB (TP53RK Binding Protein). Jurkat: PGAM5, ING3, PTK2, PKMYT1, DBF4; HepG2: TTK, PLK1, ING3, PGAM5, GAK; K562: DBF4, PLK1, BRK1, PTK2, PRKRA; RPE1: CDK2, GTF3C4, GAK, BRK1, PLK1. Cell-cycle kinase preferences differ across cell types (PKMYT1 in Jurkat, CDK2 in RPE1), suggestive of context-dependent cell cycle regulatory requirements.

Summary. These examples demonstrate that PT-RAG's differentiable retrieval learns biologically plausible, cell-type-specific patterns: functional coherence (retrieving genes from the same family

Table 6: **Cross-cell-type: full results with standard deviations.** $n = 1{,}635$ (375 HepG2, 416 RPE1, 443 Jurkat, 401 K562).

Metric	STATE	STATE+GenePT	Naïve RAG	PT-RAG
Pearson DEG ↑	0.624 ± 0.048	0.631 ± 0.051	0.396 ± 0.063	$\mathbf{0.633 \pm 0.048}$
Spearman DEG ↑	0.403 ± 0.046	0.411 ± 0.052	0.307 ± 0.041	$\mathbf{0.412 \pm 0.051}$
$\text{MSE}_{\text{PCA50}}$ ↓	8.43 ± 0.93	8.42 ± 0.88	12.64 ± 1.04	$\mathbf{8.39 \pm 0.87}$
W_1 ↓	35.70 ± 1.76	35.53 ± 1.68	48.48 ± 1.71	$\mathbf{35.41 \pm 1.62}$
W_2 ↓	646.1 ± 63.6	638.7 ± 60.6	1189.5 ± 83.3	$\mathbf{633.7 \pm 58.4}$
Energy ↓	9.41 ± 1.18	9.40 ± 1.15	14.18 ± 1.17	$\mathbf{9.33 \pm 1.15}$

Table 7: **Cross-cell-type generalization on scGPT: full results with standard deviations.** Mean values across 1,635 test perturbations from four cell types. Significance vs. PT-RAG: † ($p_{\text{FDR}} < 0.01$), †† ($p_{\text{FDR}} < 0.05$), † † † ($p_{\text{FDR}} < 0.1$). Bold: best.

Metric	scGPT	scGPT+GenePT	scGPT+PT-RAG
Gene-level expression correlations			
Pearson DEG ↑	**0.9209 ± 0.0158**	0.9167 ± 0.0149	0.9128 ± 0.0146
Spearman DEG ↑	0.9500 ± 0.0341	**0.9509 ± 0.0306**	0.9368 ± 0.0299
Expression reconstruction accuracy			
MSE_{PCA50} ↓	18.6440 ± 2.0080 †	19.8907 ± 2.8476 †	**17.3212 ± 2.1105**
Distributional similarity (PCA space)			
W_1 ↓	33.7992 ± 1.6254 †	34.6746 ± 2.2780 †	**32.7828 ± 1.8324**
W_2 ↓	578.6169 ± 54.8356 †	609.7348 ± 80.0043 †	**544.6977 ± 60.1570**
Energy ↓	23.5341 ± 1.7914 †	24.3923 ± 2.0678 †	**22.3410 ± 1.7828**

or pathway) is maintained while specific context selection adapts to the cellular background. We note these observations are suggestive and not a rigorous biological validation.

F.2 Cross-Cell-Type Results with Standard Deviations

Table 6 presents complete cross-cell-type results.

F.3 Cross-Cell-Type Results on scGPT with Standard Deviations

Table 7 provides the full cross-cell-type scGPT results (mean ± standard deviation), corresponding to Table 3 in the main text.

F.4 Cross-Perturbation Results with Standard Deviations

Table 8 provides the full cross-perturbation results (mean ± standard deviation), corresponding to Table 4 in the main text.

F.5 Per-Cell-Type Results

In this section, we want to provide the reader with information at the cell-level granularity, namely, how the model performs cell-perturbation pairs for each different training. We have results where the held-out cell was any among `HepG2`, `RPE1`, `Jurkat`, and `K562`. Such results are in Tables 9, 10, 11, 12. We see how the patterns of Table 6, reflects the results from the specific cell-lines.

F.6 Sensitivity Studies

Sparsity regularization. Figures 6 and 7 analyze $\lambda_{\text{sparse}} \in \{0, 0.01, 0.10, 1.00\}$ on HepG2 ($K = 32$). Without sparsity ($\lambda = 0$), the model retrieves ≈32 candidates on average, yielding Pearson DEG 0.134 (vs. 0.604 with $\lambda = 0.1$), Spearman -0.025, and RMSE 0.567. With any non-zero

Table 8: **Cross-perturbation generalization: full results with standard deviations.** Results averaged across three folds. Significance vs. PT-RAG: † ($p_{\text{FDR}} < 0.01$), †† ($p_{\text{FDR}} < 0.05$), † † † ($p_{\text{FDR}} < 0.1$). Bold: best.

Metric	STATE+GenePT	PT-RAG
Gene-level expression correlations		
Pearson DEG ↑	$\mathbf{0.6377 \pm 0.0367}$	0.6314 ± 0.0394
Spearman DEG ↑	$\mathbf{0.4293 \pm 0.0344}$	0.4224 ± 0.0381
Expression reconstruction accuracy		
$\text{MSE}_{\text{PCA50}}$ ↓	$1.1477 \pm 0.2243^{\dagger}$	$\mathbf{1.1316 \pm 0.2150}$
Distributional similarity (PCA space)		
W_1 ↓	$0.8043 \pm 0.0582^{\dagger}$	$\mathbf{0.8016 \pm 0.0578}$
W_2 ↓	$0.6436 \pm 0.1134^{\dagger}$	$\mathbf{0.6367 \pm 0.1093}$
Energy ↓	$0.3385 \pm 0.0459^{\dagger}$	$\mathbf{0.3357 \pm 0.0442}$

Table 9: Results on `HepG2` ($n = 375$).

Metric	STATE	STATE+GenePT	Naïve RAG	PT-RAG
Pearson DEG ↑	0.596 ± 0.051	0.595 ± 0.055	0.351 ± 0.054	$\mathbf{0.604 \pm 0.049}$
Spearman DEG ↑	0.398 ± 0.044	0.394 ± 0.058	0.289 ± 0.036	$\mathbf{0.401 \pm 0.049}$
$\text{MSE}_{\text{PCA50}}$ ↓	8.89 ± 0.32	8.86 ± 0.32	13.35 ± 0.77	$\mathbf{8.83 \pm 0.31}$
W_1 ↓	36.30 ± 1.34	35.53 ± 1.20	48.68 ± 1.31	$\mathbf{35.41 \pm 1.18}$
W_2 ↓	668.5 ± 49.5	637.7 ± 43.3	1198.5 ± 64.5	$\mathbf{631.8 \pm 42.3}$
Energy ↓	9.68 ± 0.69	9.54 ± 0.67	14.25 ± 0.90	$\mathbf{9.48 \pm 0.67}$

λ, performance stabilizes: Pearson 0.594–0.604, Spearman 0.386–0.401, RMSE ≈0.470, average selected perturbations 4.9–12.5.

Retrieval size K. Figures 8 and 9 compare $K \in \{16, 32\}$ across sparsity levels on HepG2. Both settings yield comparable performance across all sparsity levels, demonstrating that PT-RAG is robust to K within this range.

Naïve RAG retrieval size ablation. Figures 10 and 11 evaluate Naïve RAG with $K \in \{2, 5, 10, 32\}$ vs. PT-RAG ($K = 32$) on HepG2. At $K = 32$, Naïve RAG reaches Pearson 0.351 vs. PT-RAG's 0.604 (gap: 42%). Spearman is non-monotonic with K (0.220 at $K = 2$, dipping to 0.170 at $K = 5$, recovering to 0.289 at $K = 32$), and RMSE degrades from $K = 2$ to $K = 5$ before recovering. This non-monotonic behavior confirms that the failure is not due to insufficient retrieval breadth but to the lack of differentiable, cell-type-aware selection.

F.7 Additional Controls

This section reports controls that isolate where the gains of PT-RAG come from. Unless stated otherwise, they are run in the cross-perturbation setting on a single fold (3,452 test cell-type–perturbation pairs) with a PT-RAG model using $K = 32$. Since they use one fold and a different K, their values should be compared with each other and with the STATE+GenePT row of the same table, not with the three-fold averages of Table 4.

Retrieval pool. In the cross-perturbation protocol, PT-RAG can retrieve from all perturbations at test time, whereas only the training perturbations are available during training (Section D.3). To check whether the larger test-time pool explains the gains, we evaluate the same trained model with the pool restricted to the train and validation perturbations, and to the training perturbations only. All metrics are essentially unchanged, so the improvements over STATE+GenePT on reconstruction and distributional metrics do not come from the larger pool.

Contribution of the selector. To separate the learned selection from the training of the perturbation encoder, we initialise the perturbation encoder from the STATE+GenePT model and keep it frozen, training only the scorer and the projector. This improves Pearson correlation, $\text{MSE}_{\text{PCA50}}$, W_1 and W_2 over STATE+GenePT, but not Energy, and it does not reach the distributional gains

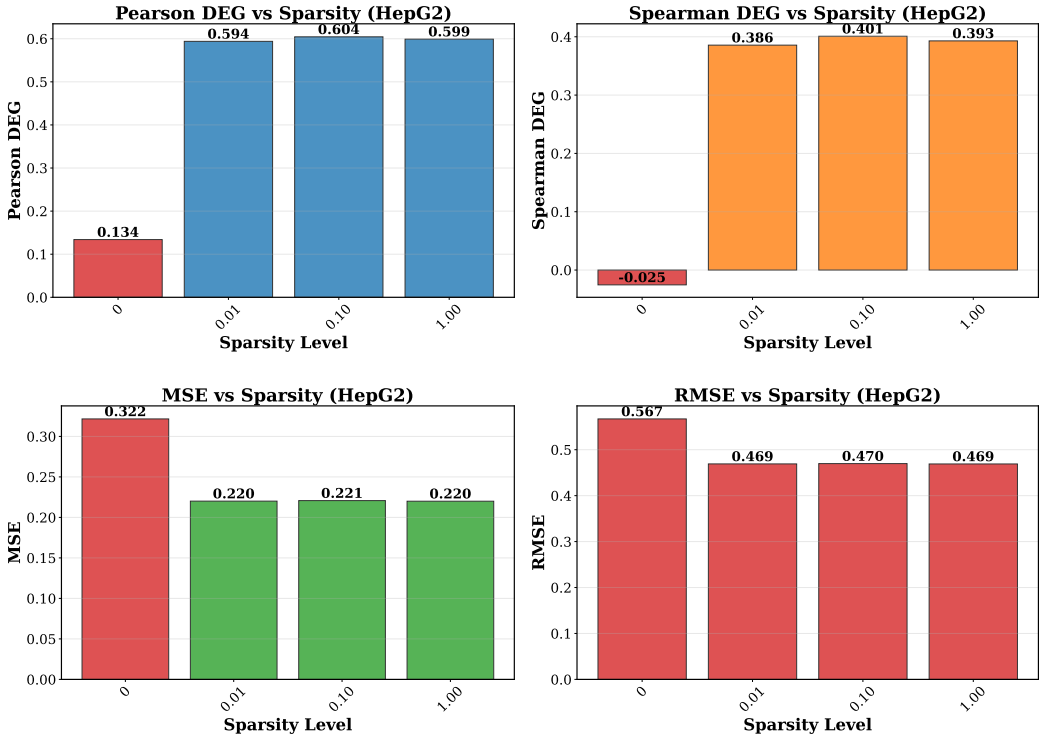


Figure 6: **Sparsity regularization sensitivity (gene-level and reconstruction metrics).** HepG2, $K = 32$.

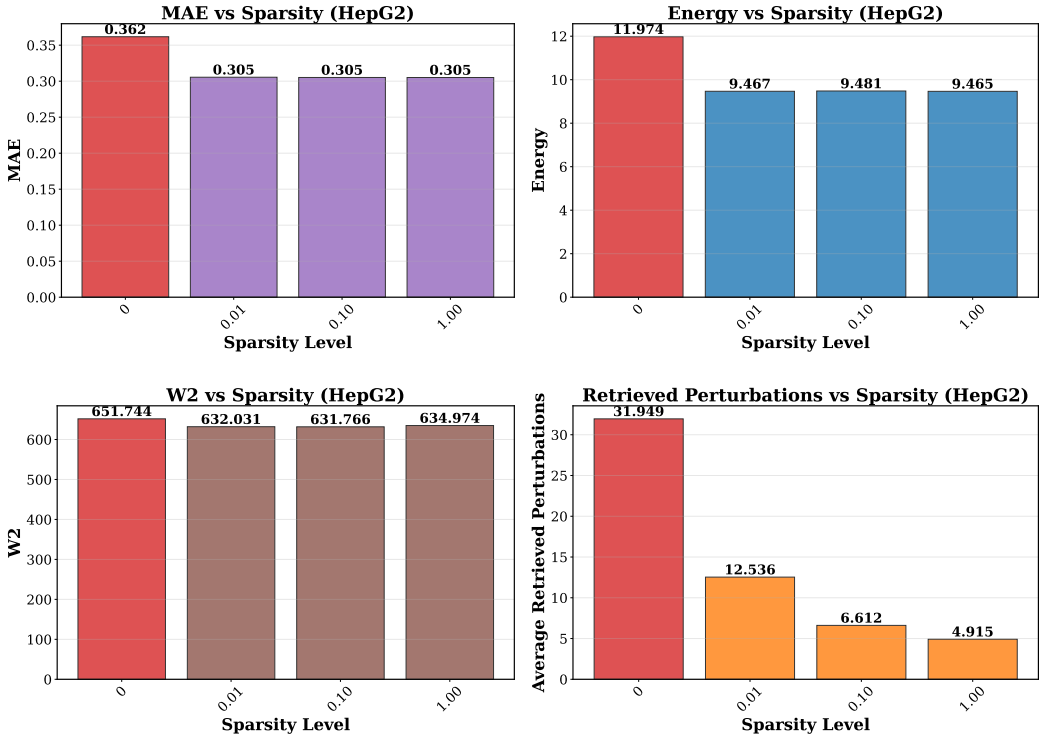


Figure 7: **Sparsity regularization sensitivity (distributional metrics and retrieval count).** HepG2, $K = 32$.

Table 10: Results on RPE1 ($n = 416$).

Metric	STATE	STATE+GenePT	Naïve RAG	PT-RAG
Pearson DEG ↑	0.621 ± 0.043	0.633 ± 0.043	0.450 ± 0.052	$\mathbf{0.640 \pm 0.042}$
Spearman DEG ↑	0.419 ± 0.035	0.432 ± 0.033	0.342 ± 0.027	$\mathbf{0.438 \pm 0.033}$
$\text{MSE}_{\text{PCA50}}$ ↓	9.56 ± 0.35	9.48 ± 0.32	13.29 ± 0.80	$\mathbf{9.44 \pm 0.33}$
W_1 ↓	36.75 ± 1.25	36.62 ± 1.24	49.31 ± 1.46	$\mathbf{36.53 \pm 1.22}$
W_2 ↓	682.5 ± 47.1	676.3 ± 46.4	1227.5 ± 72.9	$\mathbf{672.8 \pm 45.8}$
Energy ↓	10.17 ± 0.84	10.14 ± 0.84	14.56 ± 0.81	$\mathbf{10.08 \pm 0.84}$

Table 11: Results on Jurkat ($n = 443$).

Metric	STATE	STATE+GenePT	Naïve RAG	PT-RAG
Pearson DEG ↑	0.636 ± 0.048	$\mathbf{0.652 \pm 0.049}$	0.398 ± 0.056	0.649 ± 0.048
Spearman DEG ↑	0.392 ± 0.056	0.424 ± 0.058	0.318 ± 0.033	$\mathbf{0.424 \pm 0.054}$
$\text{MSE}_{\text{PCA50}}$ ↓	$\mathbf{7.35 \pm 0.41}$	7.36 ± 0.41	11.46 ± 0.62	7.36 ± 0.41
W_1 ↓	35.94 ± 1.52	36.10 ± 1.50	49.06 ± 1.50	$\mathbf{35.82 \pm 1.48}$
W_2 ↓	654.4 ± 56.7	660.1 ± 56.1	1219.6 ± 75.4	$\mathbf{649.9 \pm 55.1}$
Energy ↓	8.15 ± 1.01	8.23 ± 1.02	13.12 ± 1.10	$\mathbf{8.14 \pm 1.01}$

of the jointly trained model. Both the learned selection and its joint training with the perturbation encoder therefore contribute.

Robustness to noisy candidates. At test time, we replace 60% of the Stage-1 candidates with easy negatives, i.e. the perturbations least similar to the query. The selector does not filter them out: about 59% of the selected candidates come from the corrupted portion. The distributional metrics degrade but remain better than those of STATE+GenePT. Candidate quality therefore matters, and severe Stage-1 noise is a real failure mode, although it does not make the prediction collapse.

Candidate-set size. A larger K reduces the risk that Stage 1 misses a relevant perturbation. In the cross-cell-type setting, increasing K from 32 to 128 gives comparable results (Table 14): Pearson, W_1 and W_2 are nearly identical, Spearman is slightly lower, and $\text{MSE}_{\text{PCA50}}$ and Energy are slightly better.

F.8 Computational Costs

Table 15 compares computational requirements on the cross-cell type experiments. In Table 15, the parameter counts are all in the same range, so the main difference is the extra retrieval and selection computation introduced by PT-RAG rather than a large increase in model size. This is consistent with the design of the method: PT-RAG adds a retrieval module on top of the generator, but does not replace the backbone with a much larger architecture.

For the cross-perturbation setting, the corresponding cost comparison is shown in Table 16. Table 16 is the more informative comparison for the transfer setting: STATE+GenePT and PT-RAG again have comparable parameter counts, but PT-RAG requires substantially more FLOPs because it has to score and filter retrieved contexts over the full pool of roughly 2,009 perturbations. In other words, the extra cost is not coming from a larger backbone, but from computing relevance over a much larger perturbation set, which is exactly the price of making retrieval cell-state-aware and dynamic in the cross-perturbation regime.

For the scGPT transfer experiments, Table 17 shows the computational overhead of adding GenePT and PT-RAG to the scGPT backbone. Here the parameter increase is minimal: scGPT+PT-RAG adds only ≈ 2.4M parameters compared to vanilla scGPT, but the FLOPs remain nearly identical (65.49B per cell vs. 64.88B), indicating that GenePT retrieval and differentiable selection impose negligible computational burden in the scGPT setting.

Table 12: Results on K562 ($n = 401$).

Metric	STATE	STATE+GenePT	Naïve RAG	PT-RAG
Pearson DEG ↑	$\mathbf{0.639 \pm 0.037}$	0.638 ± 0.037	0.381 ± 0.046	0.633 ± 0.041
Spearman DEG ↑	$\mathbf{0.406 \pm 0.040}$	0.391 ± 0.044	0.274 ± 0.029	0.383 ± 0.047
$\text{MSE}_{\text{PCA50}}$ ↓	$\mathbf{8.02 \pm 0.33}$	8.07 ± 0.32	12.61 ± 0.59	8.04 ± 0.33
W_1 ↓	33.77 ± 1.21	$\mathbf{33.76 \pm 1.18}$	46.77 ± 1.23	33.80 ± 1.15
W_2 ↓	578.1 ± 42.4	$\mathbf{576.9 \pm 41.3}$	1108.5 ± 58.9	577.1 ± 40.2
Energy ↓	$\mathbf{9.73 \pm 0.91}$	9.80 ± 0.92	14.89 ± 0.97	9.74 ± 0.91

Table 13: **Additional controls, cross-perturbation (one fold, $K = 32$).** See the text for the description of each row.

Model	Pearson ↑	Spearman ↑	$\text{MSE}_{\text{PCA50}}$ ↓	W_1 ↓	W_2 ↓	Energy ↓
STATE+GenePT (no retrieval)	0.6409	0.4314	1.3056	0.8166	0.7173	0.3618
Retrieval pool available at test time						
PT-RAG, full pool (2,008)	0.6242	0.4139	1.2556	0.8073	0.6939	0.3511
PT-RAG, train+val pool (802)	0.6237	0.4135	1.2552	0.8072	0.6937	0.3511
PT-RAG, train-only pool (669)	0.6237	0.4135	1.2551	0.8072	0.6936	0.3510
Contribution of the selector and robustness						
Frozen encoder + selector	0.6432	0.4284	1.2880	0.8157	0.7104	0.3628
PT-RAG, 60% easy negatives	0.6274	0.4171	1.2675	0.8096	0.6990	0.3541

G Statistical Test Details

Normality is assessed via Shapiro-Wilk test [Shapiro and Wilk, 1965]; all metric distributions depart significantly from normality ($p < 0.05$). We apply the Mann-Whitney U test [Nachar, 2008] with Benjamini-Hochberg FDR correction [Benjamini and Hochberg, 1995].

For the cross-perturbation setting, statistical tests were run analogously with FDR correction applied across the same metric set and model pairs.

H LLM Usage

Large language models were used as writing assistants to improve clarity and presentation of the manuscript and software documentation. All scientific ideas, methods, experiments, and conclusions are solely those of the authors, and all LLM-assisted content was reviewed and edited for correctness prior to inclusion.

Table 14: **Candidate-set size, cross-cell-type** (pooled over the four cell types, $n = 1{,}635$).

Metric	$K = 32$	$K = 128$
Pearson DEG ↑	0.6326	0.6314
Spearman DEG ↑	0.4121	0.4072
$\text{MSE}_{\text{PCA50}}$ ↓	8.392	8.368
W_1 ↓	35.411	35.406
W_2 ↓	633.74	633.67
Energy ↓	9.333	9.308

Table 15: **Computational costs on Cross-Cell type generalization** (STATE). FLOPs per batch (64 cells).

Model	Parameters	FLOPs/Batch	FLOPs/Cell
STATE	20,936,480	1.67B	34.9M
STATE+GenePT	20,874,016	1.67B	34.8M
Naïve RAG	21,120,288	1.67B	34.8M
STATE+PT-RAG	20,973,602	2.86B	59.9M

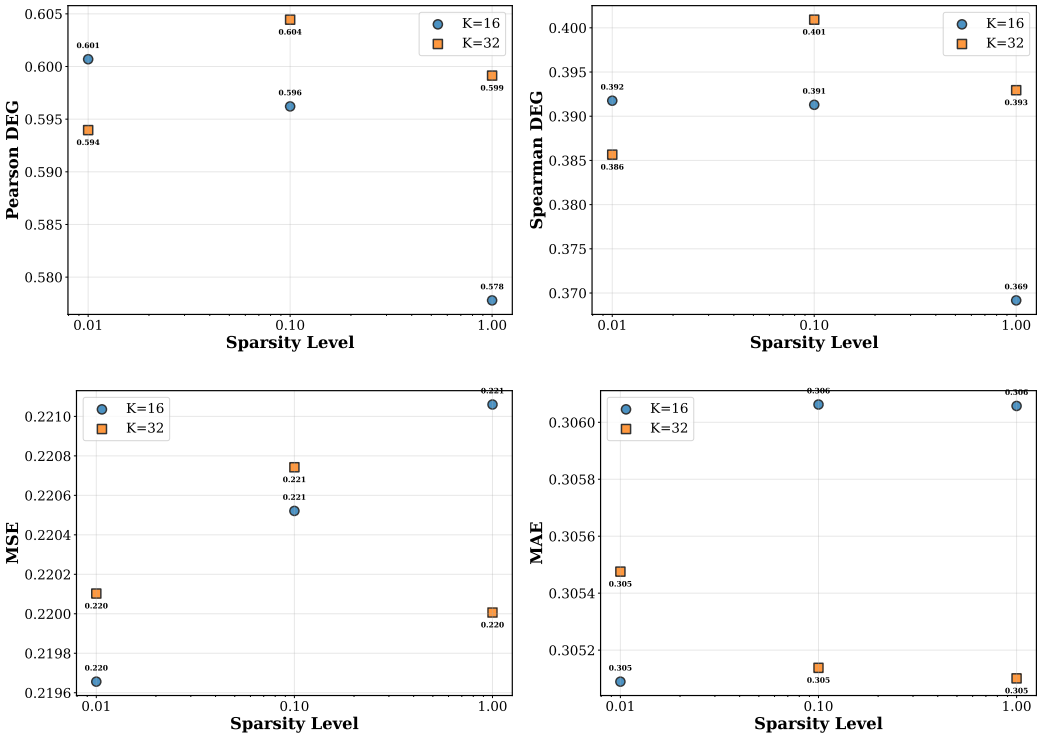


Figure 8: $K = 16$ **vs.** $K = 32$ **(gene-level and reconstruction metrics).** HepG2.

Table 16: **Computational costs on Cross-Perturbation generalization.** FLOPs per batch (64 cells).

Model	Parameters	FLOPs/Batch (64 cells)	FLOPs/Cell
STATE+GenePT	20,874,016	2.23B	34.87M
PT-RAG $K = 128$, sparsity $= 1.0$	20,973,602	7.46B	116.61M

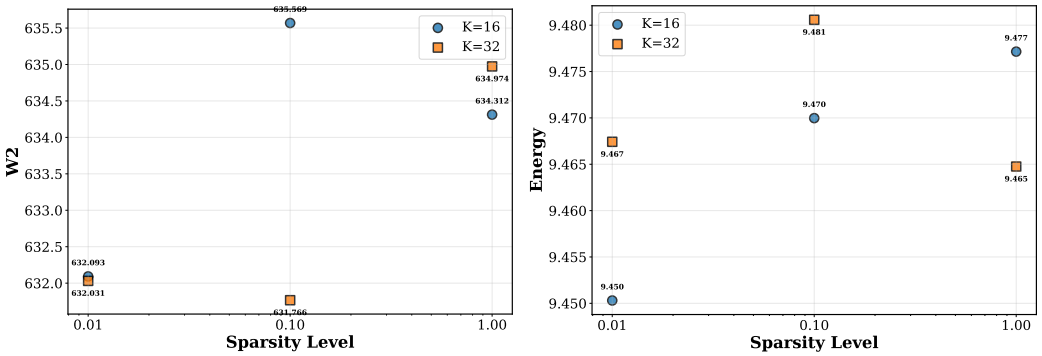


Figure 9: $K = 16$ **vs.** $K = 32$ **(distributional metrics).** HepG2.

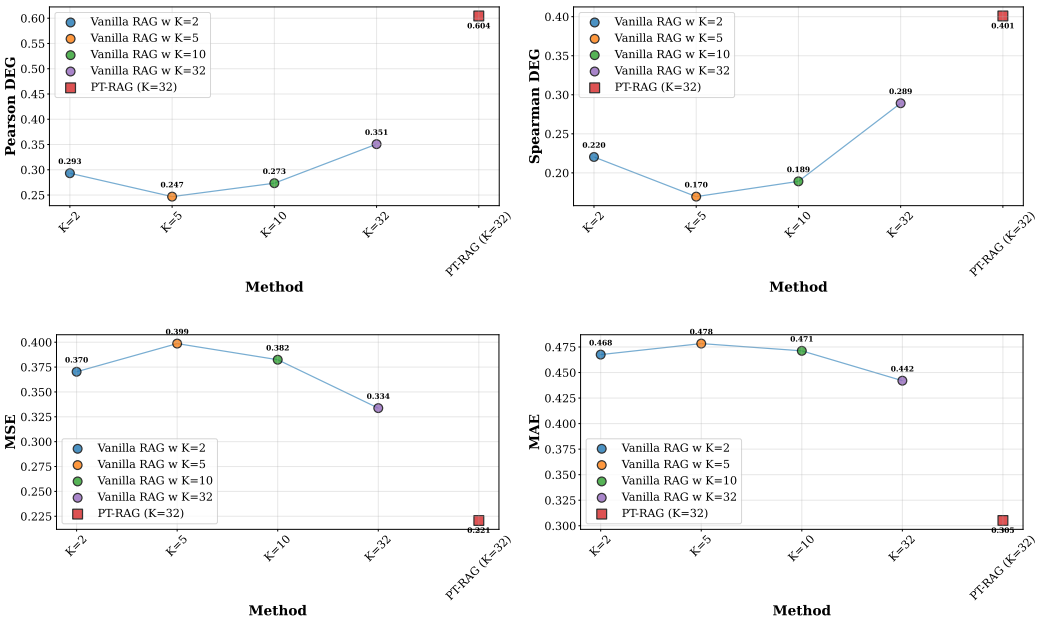


Figure 10: **Naïve RAG** $K \in \{2, 5, 10, 32\}$ **vs. PT-RAG** ($K = 32$)**.** Gene-level and reconstruction. HepG2.

Table 17: **Computational costs on Cross-Cell type generalization** (scGPT). FLOPs per batch (64 cells).

Model	Parameters	FLOPs/Batch (64 cells)	FLOPs/Cell
scGPT	51,334,658	4.15T	64.88B
scGPT+GenePT	52,121,602	4.15T	64.88B
scGPT+PT-RAG ($K = 128$, sparsity $= 1.0$)	53,699,588	4.19T	65.49B

Table 18: **FDR-corrected p-values for PT-RAG vs. baselines (cross-cell-type).** $n = 1{,}635$.

Metric	vs. STATE	vs. STATE+GenePT
Pearson DEG	2.44×10^{-8} †	0.590
Spearman DEG	4.89×10^{-10} †	0.785
$\text{MSE}_{\text{PCA50}}$	0.577	0.590
W_1	3.41×10^{-6} †	$0.082^{\dagger\dagger\dagger}$
W_2	5.47×10^{-8} †	$0.041^{\dagger\dagger}$
Energy	$0.082^{\dagger\dagger\dagger}$	0.186

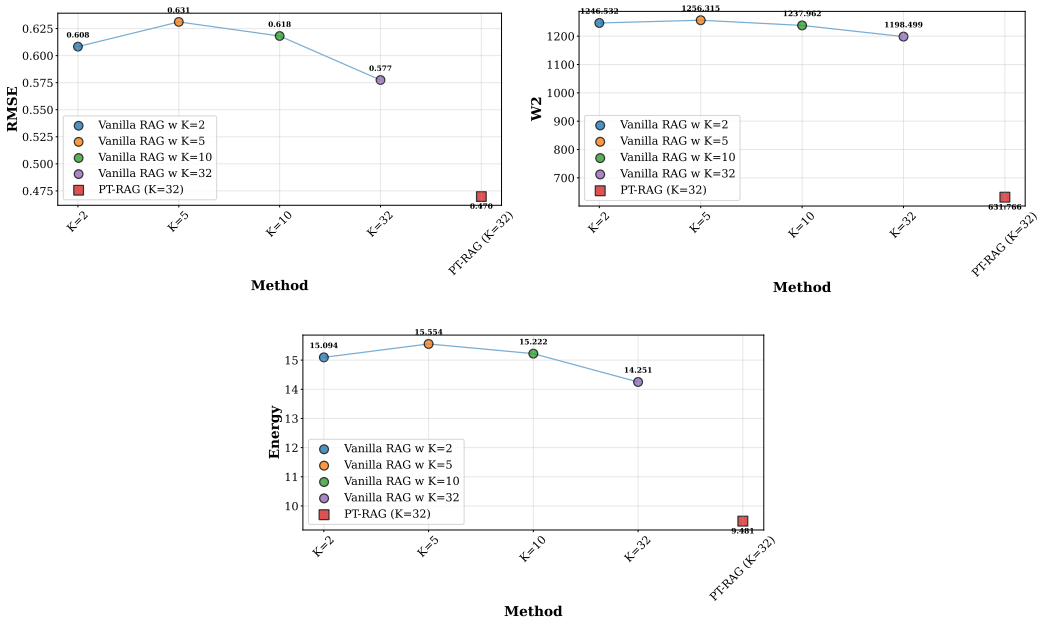


Figure 11: **Naïve RAG** $K \in \{2, 5, 10, 32\}$ **vs. PT-RAG (**$K = 32$**).** Distributional metrics. HepG2.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the paper's core claims: PT-RAG is a differentiable, cell-type-aware retrieval module; it improves cross-cell-type prediction over Naïve RAG and retrieval-free baselines; and it works as a plug-and-play module to STATE (as a reservoir) and to fully-trained scGPT. The scope and caveats are also reflected in the Discussion and Conclusion.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses concrete limitations in the Discussion section, including the reservoir setting, extra compute introduced by retrieval, the restriction to single-gene perturbations, and the fact that the retrieval examples are suggestive rather than causal.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not contain formal theoretical results, theorems, or proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper discloses the dataset, split protocol, model structure, retrieval pipeline, optimizer, learning rate, batch size, and evaluation metrics needed to reproduce the main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The manuscript shares code and reproduction details through a GitHub repository, and the appendix gives the experimental setup in enough detail for faithful reproduction of the main experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the train/test splits, retrieval-pool protocol, optimizer, learning rate, batch size, training duration, and model components in the main text and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The paper reports fold-averaged results, standard deviations, and FDR-corrected significance tests, and it explains the statistical procedures used for the reported comparisons.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper gives the relevant compute details for reproduction, including the GPU type, batch size, training steps, approximate runtime per target cell type, and the computational cost table.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses standard public perturbation-biology datasets and methods and does not indicate any deviation from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on methodological gains and does not include a dedicated discussion of negative societal impacts or mitigation strategies.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not introduce a high-risk model or dataset that requires special safeguards beyond standard code release and normal scientific use.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the original creators of the datasets, models, and methods through citations and respects their intended scientific use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new standalone asset package beyond the manuscript and the code.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human-subjects research, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are used only for writing and editing assistance, not as a non-standard component of the core method, so an LLM-usage declaration is not required for the scientific contribution.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.