Sequential Optimal Experimental Design of Perturbation Screens Guided by Multi-modal Priors

Kexin Huang[1,2], Romain Lopez[1,3], Jan-Christian Hütter[1], Takamasa Kudo[1], Antonio Rios[1], and Aviv Regev[1]

[1]Research & Early Development, Genentech
[2]Department of Computer Science, Stanford University
[3]Department of Genetics, Stanford University

Abstract

Understanding a cell's expression response to genetic perturbations helps to address important challenges in biology and medicine, including the function of gene circuits, discovery of therapeutic targets and cell reprogramming and engineering. In recent years, Perturb-seq, pooled genetic screens with single cell RNA-seq (scRNA-seq) readouts, has emerged as a common method to collect such data. However, irrespective of technological advances, because combinations of gene perturbations can have unpredictable, non-additive effects, the number of experimental configurations far exceeds experimental capacity, and for certain cases, the number of available cells. While recent machine learning models, trained on existing Perturb-seq data sets, can predict perturbation outcomes with some degree of accuracy, they are currently limited by sub-optimal training set selection and the small number of cell contexts of training data, leading to poor predictions for unexplored parts of perturbation space. As biologists deploy Perturb-seq across diverse biological systems, there is an enormous need for algorithms to guide iterative experiments while exploring the large space of possible perturbations and their combinations. Here, we propose a sequential approach for designing Perturb-seq experiments that uses the model to strategically select the most informative perturbations at each step for subsequent experiments. This enables a significantly more efficient exploration of the perturbation space, while predicting the effect of the rest of the unseen perturbations with high-fidelity. Analysis of a previous large-scale Perturb-seq experiment reveals that our setting is severely restricted by the number of examples and rounds, falling into a non-conventional active learning regime called "active learning on a budget". Motivated by this insight, we develop IterPert, a novel active learning method that exploits rich and multi-modal prior knowledge in order to efficiently guide the selection of subsequent perturbations. Using prior knowledge for this task is novel, and crucial for successful active learning on a budget. We validate IterPert using in-silico benchmarking of active learning, constructed from a large-scale CRISPRi Perturb-seq data set. We find that IterPert outperforms other active learning strategies by reaching comparable accuracy at only a third of the number of perturbations profiled as the next best method. Overall, our results demonstrate the potential of sequentially designing perturbation screens through IterPert.

1 Introduction

The expression response of a cell to a genetic perturbation reveals fundamental insights into cell and gene function [1]. Perturb-seq is a relatively recent technology for pooled genetic screens with a single-cell RNA seq (scRNA-seq) readout of the expression response to a perturbation [2, 3, 4]. Perturb-seq provides insights into gene regulatory machinery [5], helps identify target genes for therapeutic intervention [6], and can facilitate the engineering cells with a specific target state [7, 8]. Recent technical advances have enhanced the scope, scale and efficiency of Perturb-Seq [9, 4, 10]. However, because of the plethora of biological contexts, across cell types, states and stimuli, and the need to test combinations of perturbations (due to the possibility of non-additive genetic interactions), the number of required experiments explodes combinatorially. With trillions of potential experimental configurations or more, it becomes unrealistic to conduct all of them directly [8, 11].

Recently, researchers proposed machine learning models to predict perturbation outcomes [12, 13, 14]. Such models are trained on existing Perturb-seq datasets [10, 2, 8, 11] and then predict expression outcomes of unseen perturbations, of single genes or their combinations. While promising, these models suffer from a selection bias caused by the design of the original experiment used for training, in terms of selected perturbations and biological conditions. In particular, the training data are often profiled to answer a specific biological question, but not to maximize the predictive accuracy of the machine learning model across a large pool of unprofiled perturbations.

In this work, we present a novel paradigm for exploring a perturbation space by executing a sequence of Perturb-seq experiments. At the core of this paradigm lies a sequential optimal design procedure that interleaves the machine learning model and the wet-lab, where the Perturb-seq assay is performed. At each step of the sequence, we acquire data and use it to re-train the machine learning model. Then, we apply an optimal design strategy to select a batch of perturbation experiments that will most benefit the model to predict all of the unprofiled perturbations. The key idea is to sample the perturbation space intelligently by considering perturbations that are most informative and representative to the model, while accounting for diversity. Using this strategy, we can run as few perturbation experiments as possible, while obtaining a model that has sufficiently explored the perturbation space.

This idea is well-studied in the machine learning literature, and is the topic of active learning [15]. Active learning has been used in practice across many domains, such as document classification [16], medical imaging [17] and speech recognition [18]. However, we noticed that effective active learning approaches necessitate a substantial initial set of labeled examples (i.e., in our case, profiled perturbations), complemented by numerous batches that collectively result in tens of thousands of labeled data points [19, 20]. In contrast, the constraints of iterative Perturb-seq in the lab make such conditions unattainable, both in terms of cost and time (as shown in our economic analysis in Section 3.1). In this "budgeted" regime, it has been reported that random selection outperforms most active learning strategies [21, 22, 23].

We therefore propose a new strategy called IterPert (Iterative Perturb-seq) that tackles the active learning on a budget setting for Perturb-seq data. Motivated by a data-driven analysis, our key observation is that when on a budget, it may be beneficial to combine the evidence from the data with publicly available sources of prior-knowledge, especially in the first few rounds. Such examples of prior knowledge include Perturb-seq data from related systems, large scale genetic screens with other modalities, such as genome-scale optical pooled screens [24, 25], and data on physical molecular interactions, such as protein complexes [26, 27]. This prior information spans multiple modalities such as networks, text, image, and 3D structure, which may be challenging to exploit during active learning. We overcome this by defining reproducing kernel Hilbert spaces on each of the modalities, and applying a kernel fusion strategy [28] to combine information from multiple sources.

To compare IterPert against other commonly used methods, we conducted an extensive empirical study using a large-scale single-gene CRISPRi Perturb-seq dataset collected in a cancer cell line (K562 cells) [11] and benchmarked 8 recent active learning strategies. IterPert achieved similar accuracy as the best active learning strategy but with three times fewer perturbations profiled as the training data. IterPert also showed robust

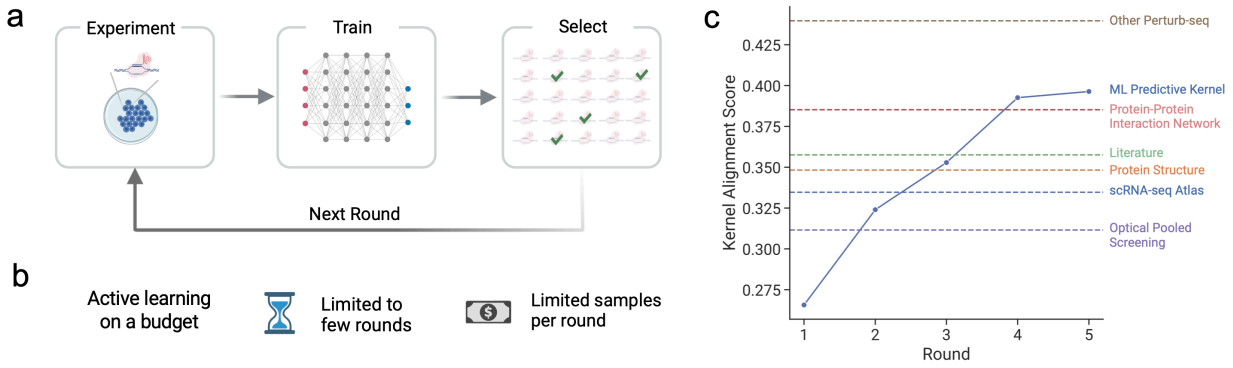

Figure 1: **Sequential design of Perturb-seq experiments. a.** Illustration of the iterative Perturb-seq procedure. In each round, a batch of perturbations is selected and the corresponding experiments are conducted. Then, a machine learning model is updated with these newly-profiled perturbations. An active learning strategy uses the model's predictions to select the set of perturbations for the next round. Through this iteration, the goal is to reach high accuracy with a minimal number of experiments. **b.** Illustration of "active learning on a budget". Active learning for Perturb-seq is highly restricted to much fewer profiled perturbations (i.e., labeled examples) compared to a conventional active learning setting. This motivates the development of a specialized method for this setting. **c.** Exploratory data analysis shows that the model kernel suffers from poor representation when few perturbations have been profiled (low budget). However, other data sources, described in Section 3.3, contain rich and complementary information that can be potentially transferred to the model kernel, motivating IterPert.

performance in both essential genes screens and genome-scale screens, and when considering batch effects across iterations.

To summarize, our contributions are (1) proposing a sequential experimental design approach to Perturb-seq profiling for efficient exploration of a perturbation space; (2) identifying the algorithmic problem of active learning on a budget in this setting; (3) proposing a new active learning strategy that incorporates prior information and obtains a speedup of more than three times over the best baseline strategy.

2 Background

Perturb-seq prediction model. We consider a predictive model f_θ with parameters θ that maps a set of perturbations $\mathcal{P} = (P_1, \cdots, P_M)$ to the post-perturbed expression outcome $\hat{\mathbf{y}} \in \mathbb{R}^L$, where L denotes the number of genes with measured expression levels. We denote the set of available Perturb-seq training data as $\mathcal{D}_{\text{train}} = \mathcal{X}_{\text{train}} \times \mathcal{Y}_{\text{train}}$, where $\mathcal{X}_{\text{train}} = \{\mathcal{P}_i\}_{i=1}^{N_{\text{train}}}$ and $\mathcal{Y}_{\text{train}} = \{\mathbf{y}_i\}_{i=1}^{N_{\text{train}}}$, respectively.

Several models have been designed for this specific task [12, 13, 29, 30], and our proposed framework can be adapted for any of those (refer to Section 5). However, in the remainder of this paper we focus on adopting the current state-of-the-art model GEARS [12] as the prediction model for active learning. GEARS is a deep learning model customized for perturbation prediction that uses graph neural networks (GNN) to incorporate gene ontology and gene co-expression graphs to learn perturbation embeddings from data. GEARS uses a focal loss as the objective function during training in order to assign higher weight to differentially expressed genes: $\mathcal{L} = \frac{1}{|\mathcal{D}|}\sum_{u=1}^{|\mathcal{D}|}\frac{1}{L}\sum_{v=1}^{L}(\mathbf{y}_v^u - \hat{\mathbf{y}}_v^u)^{2+\gamma}$, where $\gamma = 2$ and $\hat{\mathbf{y}}_v^u, \mathbf{y}_v^u$ are the predicted and true expression level of gene v after perturbation u, respectively.

Batch-mode pool-based active learning. Except for the specific low-budget setting, the active learning problem we are interested in has been well-studied in the literature [15] and corresponds to batch-mode pool-based active learning. It can be formulated as follows: We consider an initial labeled training set $\mathcal{D}_{\text{train}}^{(0)}$ and an unlabeled pool set $\mathcal{X}_{\text{pool}}^{(0)}$. In each subsequent round i, we first train a model f_θ on $\mathcal{D}_{\text{train}}^{(i-1)}$. Then, an active learning

selection strategy g takes in (1) a pre-specified batch size N_{batch}, (2) the training set $\mathcal{D}_{\text{train}}^{(i-1)}$, (3) the unlabeled pool set $\mathcal{X}_{\text{pool}}^{(i-1)}$, and (4) the model f_θ and selects a batch $\mathcal{X}_{\text{batch}} \subset \mathcal{X}_{\text{pool}}^{(i-1)}$. We then acquire the labels $\mathcal{Y}_{\text{batch}}$ for $\mathcal{X}_{\text{batch}}$ (i.e., for our biological setting, we run the perturbation experiment). Finally, we update the labeled set $\mathcal{D}_{\text{train}}^{(i)} = \mathcal{D}_{\text{train}}^{(i-1)} \cup \mathcal{X}_{\text{batch}}$ and pooled set $\mathcal{X}_{\text{pool}}^{(i)} = \mathcal{X}_{\text{pool}}^{(i-1)} \setminus \mathcal{X}_{\text{batch}}$. We proceed with the next round until a total of R rounds is reached.

Recently, the algorithmic framework by Holzmüller et al. [31] unified a large number of existing methods for this task. Their approach relies on reproducing kernel Hilbert spaces (RKHS), and computations on kernel matrices. Specifically, it consists of three steps. (1) *Base kernel calculation.* We construct a positive semi-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ to capture how the predictions from f_θ change with respect to $\mathcal{X}$. A common choice is to build a finite-dimensional feature map $\phi : \mathcal{X} \to \mathbb{R}^d$ with $k(x, x') = \langle \phi(x), \phi(x') \rangle$. Typical examples are the full gradient kernel, obtained for $\phi_{\text{grad}}(x) = \nabla_\theta f_\theta(x)$, as well as the last layer kernel $\phi_{\text{ll}}(x) = \nabla_{\mathbf{W}^{(L)}} f_\theta(x)$, where $\mathbf{W}^{(L)}$ is denoted as the last layer parameter of the model f_θ. We note that because there are L gene expression levels to predict for each perturbation, we are interested in a multi-task prediction problem. Yet, we may still operate within this framework. Indeed, even though the gradient vector $\nabla_\theta f_\theta$ becomes a Jacobian matrix $\text{Jac} f_\theta(\theta)$, we may just identify the matrix space to a finite-dimensional vector space. (2) *Kernel transformation.* While the base kernel defines the relation among inputs, it often requires an additional kernel transformation step for better performance, such as min-max normalization. (3) *Selection rule.* Lastly, given the transformed kernel, a selection method is invoked. The overall principle is to select informative and representative points that account for diversity. Since our proposed strategy modifies neither the kernel transformation nor the selection rules, we refer the readers to [31] for a more detailed review.

3 Method

3.1 Sequential design of Perturb-seq experiment

We now describe the unique challenges that may arise while designing an active learning strategy for the sequential design of Perturb-seq experiments.

Problem definition. We consider an initial Perturb-seq readout $\mathcal{D}_{\text{train}}^{(0)}$ and a pool of unperturbed genes $\mathcal{X}_{\text{pool}}^{(0)}$. In each round i, we train a perturbation prediction model f_θ using available data $\mathcal{D}_{\text{train}}^{(i-1)}$. Then, an active learning selection strategy g selects a batch $\mathcal{X}_{\text{batch}} \in \mathcal{X}_{\text{pool}}^{(i-1)}$. We then conduct a wet-lab Perturb-seq experiment on these selected perturbations and obtain a batch of new readouts $\mathcal{Y}_{\text{batch}}$, and proceed with the next round. The goal is find a selection strategy g that minimizes the model's prediction error $\mathcal{L}^{(i)}$ on all the perturbations. For evaluation purposes, we evaluate the performance on a hold-out set of perturbations $\mathcal{D}_{\text{test}}$ at each round i.

Experimental setup. We focus on a CRISPRi Perturb-seq screen on cells from the K562 cell line undergoing 2,058 single-gene perturbations (essential genes as defined in [11]). We construct a benchmark to simulate the real-world active learning loop. First, we randomly select a hold out set of 205 perturbations for evaluation. This randomly selected hold out set gauges a model's capacity to predict the entire perturbation space. Next, we set the number of rounds $R = 5$ and the number of perturbations that can be performed in each round $N_{\text{batch}} = 100$. For the sake of a fair comparison, we fix a random initial set of 100 perturbations for all methods. We measure the error at round i using the GEARS training loss at the hold out test set.

Economic analysis reveals active learning on a budget setting. Our problem is drastically different from the conventional active learning setting in several ways. First, previous works focus on single-output classification/regression tasks [32, 31], while the outcome of a Perturb-seq experiment is high-dimensional. This means that the predictive model may be harder to learn and thus may require more data.

Second, in a typical setting, the initial labeled set $|\mathcal{D}_{\text{train}}|$ is large enough for model training [32], followed by a large number of labels queried per round. However, this large number of labeled data is unattainable for

Perturb-seq data, because each perturbation is associated with a high cost. Perturb-seq's cost (currently dominated by the cost of scRNA-seq) for one perturbation can be estimated as the price of processing and sequencing a cell (varies across techniques, for droplet-based microfluidics, ~0.5$ [33]) times the number of guides per perturbation (~2) times the number of cells per guide (~30) in addition to the pro-rated cost of labor, instruments, and quality control. Thus, a single perturbation is currently estimated to cost more than $30, making the number of perturbations intrinsically limited per round. Indeed, most Perturb-seq experiments reported to date are in the order of hundreds of perturbations [34, 8, 2, 10], largely driven by cost, as the experiment scales readily to genome-scale in the lab.

Third, previous works assume that many rounds of data acquisition can be performed. For example, the recent GeneDisco active learning challenge uses up to 40 rounds [35]. In contrast, each round of a Perturb-seq experiment is time-consuming. With some variation due to differences in the experimental platform, on average, each round of Perturb-seq takes at least a month (1 week for oligonucleotides synthesis, 1 week for library cloning, 1 week for titering, and 1 week for experiments). Thus, the number of rounds R should be small since the total time grows linearly with R for sequential Perturb-seq design. 40 rounds correspond to more than 3 years of implementation and is thus not realistic with current assay capabilities. This is why in our setting, we use $R = 5$.

Overall, we are operating in a different regime that we summarize as active learning on a budget [32]. We demonstrate below that this has significant impact on the design of the active learning strategy.

3.2 Data-driven motivation for incorporating prior knowledge

We hypothesize that the setting of active learning on a budget will affect the performance of any active learning strategy significantly because of the estimation of the kernel matrix, and therefore also the estimated relationships between perturbations, may be highly biased. We next present an analysis to support this hypothesis.

Testing alignment of kernels. We develop a simple test to gauge the quality of any kernel for downstream utilization in an active learning strategy. Since we have access to ground truth data (i.e., the outcome of all perturbations), we construct a ground truth kernel k_{truth} with $\phi_{\text{truth}}(x) = \mathbf{y}$, where $\mathbf{y}$ denotes the experimental result. We expect the kernel matrix $\tilde{\mathbf{K}}$ to reflect pairwise perturbation relationships, up to experimental noise. Therefore, the kernel matrix $\mathbf{K}$ derived from the predictive model f_θ should ideally be aligned as closely to the ground truth kernel as possible. To measure the alignment between the query kernel $\mathbf{K}$ and the ground truth kernel $\tilde{\mathbf{K}}$, we use the kernel alignment score $\text{KA}(\mathbf{K}, \tilde{\mathbf{K}})$ [36] defined as the cosine similarity between the two matrices (using the inner product canonically induced by Frobenius norm).

Poor alignment of the predictive model kernel when on a budget. We apply a baseline active learning algorithm for five rounds, where, at each round, we randomly query 100 new perturbations (random selection rule). At each of these five rounds, we also retrieve the perturbation prediction model, compute the kernel matrices and calculate the alignment score with ground truth. To make these kernel matrices comparable across rounds, we calculate them on the list of perturbations in the pool set of the last round. We observe that as the number of profiled perturbations increases, the model kernel alignment score also increases (Figure 1c). However, the alignment scores are low during the first few rounds, suggesting that the kernel matrix does not accurately represent the similarities between perturbations. This will lead to suboptimal selections since selection rules solely rely on the kernel to make selections.

Prior knowledge contains auxiliary information of perturbation relationships. To tackle the insufficiency of the model kernel, we hypothesize that we can leverage abundant information about perturbation relationships stored in other sources of prior knowledge. To support this hypothesis, we collect a list of such sources and derive kernels that represent perturbation similarities (details about the sources and kernel derivations appear in Section 3.3). Using the same kernel alignment metric (Figure 1c), we observe that kernels derived from prior information have better alignment with the ground truth kernel compared to the model predictive

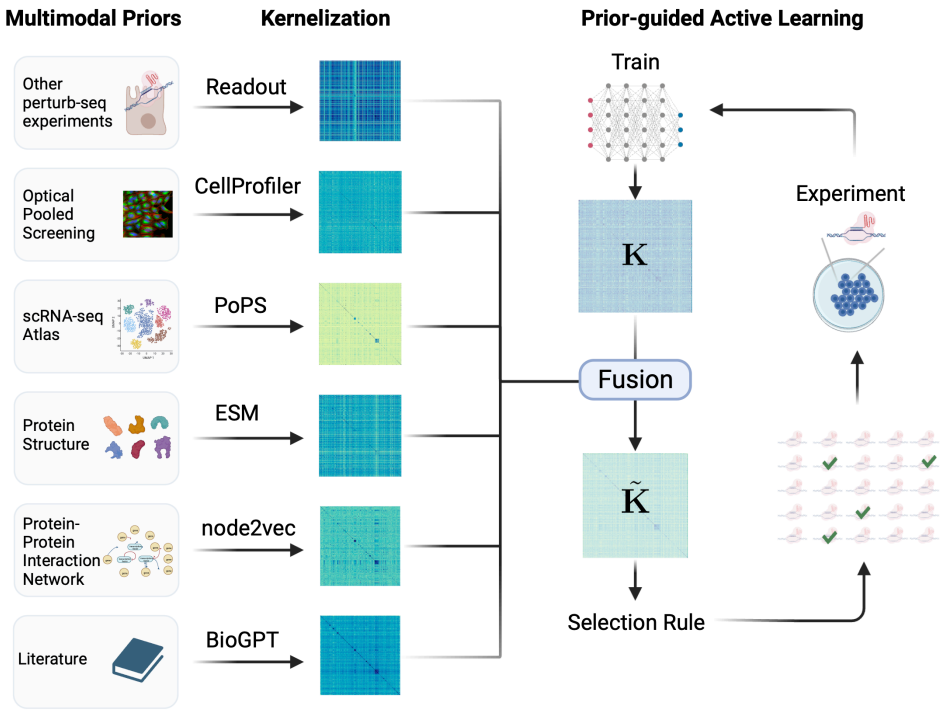

Figure 2: **Illustration of IterPert.** Driven by the exploratory data analysis in Section 3.2, we introduce IterPert, an active learning selection approach that integrates a wide range of multi-modal prior knowledge to tackle the problem of active learning on a budget for Perturb-seq data. Our primary technique involves enhancing the model kernel when faced with budget constraints. We achieve this by transforming each source of multi-modal prior knowledge into a reproducing kernel Hilbert space through diverse featurization methods for each modality, explained in Section 3.3 These kernels are then fused to refine the model kernel, ensuring a more precise characterization of perturbation relations. We then apply standard selection rules to this enhanced kernel, giving rise to IterPert.

kernel, especially in the first 2 rounds, suggesting rich information that could be complementary to the model kernel. This motivates us to design a method that integrates prior knowledge into active learning strategies.

3.3 IterPert: a multi-modal prior-guided active learning strategy

Overview. Motivated by the exploratory analysis in Section 3.2, we propose IterPert, an active learning strategy that incorporates diverse sources of prior knowledge to complement the model kernel when on a budget. The key step of our method consists in defining a kernel on each source of prior knowledge and combining those kernels with the model kernel to capture the relations between perturbations more accurately. This new prior-fused kernel is then followed by standard selection rules to form an active learning strategy.

Kernelized multi-modal prior information. Prior knowledge may come from diverse modalities, such as images, texts, and networks; therefore, how to employ these sources of prior knowledge for active learning is not straightforward. The information needed for active learning is not the raw prior knowledge, but the relations between the perturbed genes captured in the prior knowledge (e.g., using a kernel matrix). Thus, we propose to define a kernel $k(x, x') = \langle \phi_{\text{prior}}(x), \phi_{\text{prior}}(x') \rangle$ for each source of prior knowledge. Here, we introduce 6 distinct categories of prior knowledge, explain how to engineer a feature map ϕ, and provide insight on why each one should intuitively help map the perturbation space. The detailed preprocessing for each source can be found in Appendix A.

(1) Additional Perturb-seq data. Multiple Perturb-seq experiments have been conducted across several cell con-

texts [34]. Perturb-seq data from other cell contexts or experiments contain useful prior information since certain relations between perturbations might be either context-agnostic or at least transferable to the cell context of interest. For each perturbation x, $\phi(x)$ is defined as the mean of pseudo-bulk expression change from the non-targeting control cells from the Perturb-seq readouts.

(2) Optical pooled screens (OPS). OPS [24, 25] data consists of cell morphological images associated with a genetic perturbation in each individual cell in a pool. Intuitively, perturbations that elicit similar morphological phenotypes could also have similar expression phenotypes. For each perturbation x, $\phi(x)$ is the imaging features from CellProfiler [37], an image processing software that extracts morphology profiles.

(3) scRNA-seq atlas. Genes that are co-expressed together likely belong to similar pathways, and perturbations in the same pathway tend to have similar expression effects. Thus, gene co-expression data could be useful for the prediction task. For each perturbation x, $\phi(x)$ is the list of normalized gene expression measurements for gene x across a collection of scRNA-seq experiments [38]. The kernel matrix derived from this feature map corresponds to the co-expression matrix.

(4) Protein structures. If the proteins encoded by the perturbed genes have similar structures, they are more likely to have similar functions, and similar perturbation outcomes [26]. For each perturbation x, we obtain its protein coding sequence, and then feed it into a recent protein language model (15B ESM model [27]) to obtain structural features $\phi(x)$.

(5) Protein-protein interaction network (PPI). A PPI network connects proteins that physically interact with each other [39]. Intuitively, a physical interaction between two proteins suggest that they might participate in a shared biomolecular pathway or complex. Thus, perturbations of genes coding for physically interacting proteins might lead to similar effects [40]. For each perturbation x, $\phi(x)$ is a node embedding of x in the PPI network, such as node2vec [41].

(6) Literature. Perturbations that are mentioned in similar contexts in the literature are more likely to have similar functions and phenotypes. To encode this, for each perturbation x, $\phi(x)$, we feed the corresponding gene name to a recent large language model that is fine-tuned on biological literature (*e.g.* BioGPT [42]) and use the text embedding as the feature map.

Kernel fusion. The kernelization step enables integration across kernels with diverse modalities, since it converts different modalities into one — a kernel matrix of the same size. Now, we study how to fuse all the prior kernels with the model kernel. Given the set of prior kernels $\{k_1, \cdots, k_m\}$ and their kernel matrices $\{\mathbf{K}_1, \cdots, \mathbf{K}_m\}$, we update the model kernel matrix $\mathbf{K}$ at each round to obtain the kernel matrix $\hat{\mathbf{K}}$ for our active learning procedure as follows:

$$\hat{\mathbf{K}} = \text{FUSION}(\mathbf{K}, \mathbf{K}_1, \cdots, \mathbf{K}_m)$$

Since the different prior kernels have different feature map dimensions, and the kernel corresponds to taking the dot product between feature maps, the scale differs significantly across kernels. Thus, to avoid one kernel with a large scale overriding the others, we apply a min-max scale normalization to each kernel.

We experiment with multiple strategies for the FUSION operator, including element-wise operators, such as mean, max, and product, and adaptive kernel aggregation methods, such as the kernel alignment weighted operator and the kernel regression operator. A discussion and performance study of these fusion operator appears in Appendix B. Interestingly, the mean operator $\hat{\mathbf{K}} = \frac{1}{m+1}(\mathbf{K} + \mathbf{K}_1 + \cdots + \mathbf{K}_m)$ has the best empirical performance. Additionally, this approach has the theoretical advantage of guaranteeing that the fused kernel is positive and semi-definite (PSD), which is a required property for several downstream selection rules [31]. Note that the mean operator also has an interpretation in the feature maps space, where it is equivalent to the concatenation of all the feature maps.

Selection rule. IterPert only modifies the base kernel and is agnostic to the selection rule. For the sake of simplicity, in our experiments, we apply IterPert with only one popular rule called greedy distance maximization. This method greedily select points with maximum distance to all previously selected points [43]. Particularly,

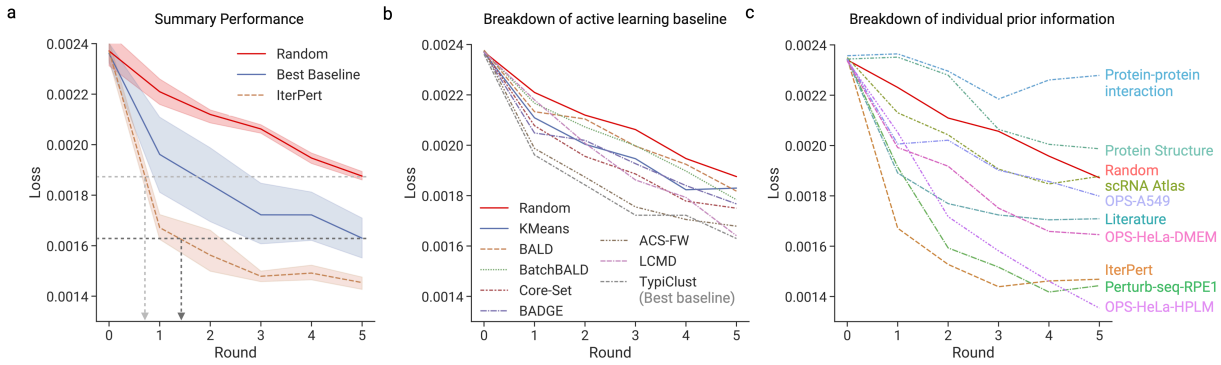

Figure 3: **a.** IterPert achieves significant speedup of model learning compared to the best baseline and random selection. Focal loss (training objective of the base model, y axis) across active learning rounds (x axis). We conduct 10 random runs where the solid line is the average and the error bar is the 95% confidence interval of the mean. **b.** Detailed breakdown of state-of-the-art active learning baselines. The best baseline is TypiClust [32]. Plot as in panel a, with the solid line denoting the average across 10 runs. **c.** Detailed breakdown of individual prior-augmented active learning. The solid line is the average across 5 runs. Error bars are not visualized in panels b and c for visual clarity and can be found in Appendix C.

given the prior-fused kernel $\hat{k}$, for a perturbation i in the pool set and any point j in the selected set, it first calculates the distance $d_{ij} = \sqrt{\hat{k}(x_i, x_i) + \hat{k}(x_j, x_j) - 2\hat{k}(x_i, x_j)}$. This is equivalent to taking the squared distance in the feature map space. Next, it selects point i^* greedily as

$$i^* = \text{argmax}_{i \in \mathfrak{X}_{\text{rem}}} \min_{j \in \mathfrak{X}_{\text{sel}}} d_{ij}, \tag{1}$$

where $\mathfrak{X}_{\text{sel}}$ is the union of the training set and the points already selected, and $\mathfrak{X}_{\text{rem}}$ is the pool set excluding the already selected points.

4 Experiment

We conduct experiments to demonstrate IterPert's advantage over state-of-the-art active learning strategies in efficiently designing Perturb-seq experiments. We also conduct systematic ablation studies to delineate the contribution of each prior information source. We evaluate the performance of our benchmarked methods in various settings, including an extension to a larger pool size by leveraging a genome-scale Perturb-seq screen and also accounting for batch effects across rounds.

Benchmarking state-of-the-art active learning methods. We first benchmark the set of active learning methods (Figure 3b) available from the open-source repository released by Holzmüller et al. [31]. We observed that all active learning methods have better performance than uniform/random sampling. The best-performing method was TypiClust [32], which is a recent active learning on a budget method that prioritizes typical examples instead of uncertain examples and shows significant improvement over random selection. This corroborates our hypothesis that the problem of sequential Perturb-seq experimental design corresponds to the setting of active learning on a budget.

IterPert achieves significant improvement over the best baseline. We report the performance of IterPert against the best active learning baseline and random sampling in Figure 3a. Importantly, IterPert uses roughly one round to reach the same accuracy as five rounds of uniform sampling, reflecting a greater than 5-fold speedup. Similarly, IterPert uses roughly 1.5 rounds (through linear extrapolation) to reach the same accuracy as five rounds of uniform sampling, reflecting a more than 3-fold speedup. We also observe similar improvements in other biologically meaningful metrics, such as the mean squared error (MSE) of predicted expression profiles calculated on the top 20 differentially expressed genes in each perturbation (Figure 4a) and the

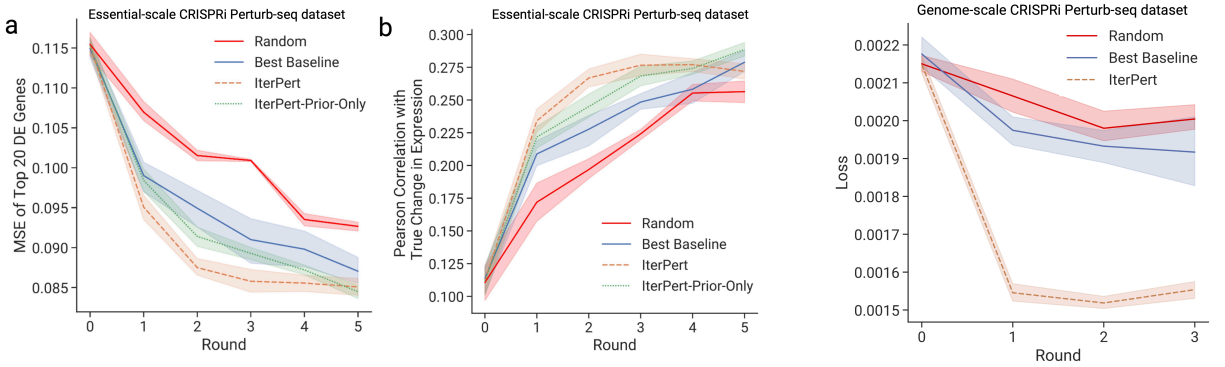

Figure 4: Biologically meaningful metrics as used in [12]. Metrics (y axis) across active learning rounds (x axis). Each method is averaged across 10 runs and error bar is the 95% CI of the mean. **a.** MSE of top 20 differentially expressed genes per perturbation. **b.** Pearson correlation coefficient between the predicted and true expression changes (centered on non-targeting controls). IterPert-Prior-Only is an ablation of IterPert where we remove the model kernel. Best baseline is TypiClust [32].

Figure 5: Performance for genome-scale K562 CRISPRi screen. Focal loss (y axis) across active learning rounds (x axis). Solid line: mean over 10 runs, error bar: 95% CI of the mean. Best baseline is TypiClust [32]. Results for other baselines can be found in Appendix D.

Pearson correlation coefficient over changes in gene expression (Figure 4b). This showcases that IterPert is an efficient method for designing Perturb-seq experiments. Moreover, the first round had the steepest increase of accuracy for IterPert, confirming our data analysis in Section 3.2 on the usefulness of prior knowledge.

Dissecting IterPert performance across multi-modal priors. To further understand the origin of the performance improvement of IterPert, we conduct several ablations. First, we report performance when using a single prior kernel, so that we may understand which prior source contributes most to the performance of the method (Figure 3c). Aggregation of all priors outperformed any individual prior alone. This showcases synergies across the diverse sources of prior knowledge. Comparing across priors, the best-performing prior is the Perturb-seq data in RPE1 cells, highlighting that there is transferable information across Perturb-seq experiments, even from different cell contexts. Optical pooled screens were also strongly informative, demonstrating that cell morphology carries shared information with Perturb-seq outcomes. Notably, different cell contexts and treatment/phenotypes of OPS lead to different improvement levels. The HeLa cell line seems to have a larger contribution to model performance increase than an OPS in the A549 cell line. Other prior knowledge sources, such as literature and a scRNA-seq atlas, also show an improvement, while PPI has limited contribution, maybe due to noise. Overall, perturbation-specific priors have richer signals compared to general gene-based priors. We also conduct an ablation where we remove the model kernel (Figure 4a,b). We observe a performance degradation, highlighting the synergy between prior knowledge and the model kernel.

Extension to genome-scale experiment. The pool set of the essential genes in K562 dataset is relatively small (<2,000 perturbations). For many real-world applications of Perturb-seq, one may want to select from a larger pool set size, for example, in genome-scale screens or in combinatorial screens. To gauge the improvement in larger setups, we conducted another experiment by leveraging the genome-scale K562 CRISPRi perturb-seq screen from [11]. This dataset has 9,748 single-gene perturbations and thus corresponds to a much larger pool of possible perturbations. We set $N_{\text{batch}} = 300$ and performed $R = 3$ rounds in total. We report the performance in Figure 5. We find that IterPert consistently displays a significant efficiency improvement over both the random and best active learning baselines, especially in the first round.

Accounting for batch effects across rounds. One important consideration when developing an active learning strategy for Perturb-seq data is that there are batch effects across rounds (Figure 6a), which could bias the

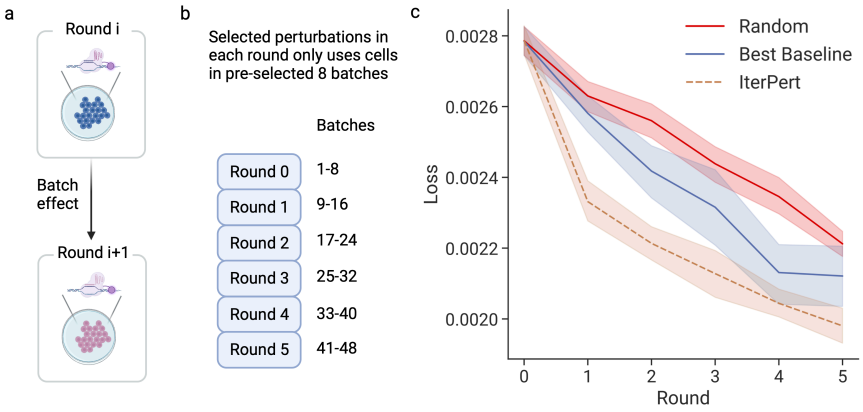

Figure 6: **a.** Batch effects exist across active learning rounds, which could bias model training and selection. **b.** Illustration of simulation evaluation settings where we restrict the cells from the selected perturbations in each round to certain batches (lanes) such that different rounds use cells from different batches. **c.** Active learning performance in the batch effect setting. Each method is averaged across 10 runs and error bar is 95% CI. Best baseline is TypiClust [32].

predictive model and selection strategy. To evaluate the our method's robustness to this, we simulate batch effects by leveraging the batch information in the dataset, which consists of 48 batches (lanes) (Figure 6b). In particular, we restrict the cells for $\mathcal{X}_{\text{batch}}$ in each round to come from different batches (8 batches for each round), to ensure that the model experiences some batch effects. We conduct the same experiment and report the performance in Figure 6c. We observe that the absolute value of the loss is worse than in the previously explored settings without batch effects. This may mainly be due to the fact that 6 times fewer cells are available for training. In this challenging setting, we still observe that IterPert has a more efficient selection strategy compared to the best baseline and uniform sampling.

5 Related works

Active learning. We highlight recent advancements that we consider as baselines and refer the readers to surveys [15, 31, 44] for a more comprehensive overview. BALD [45] selects instances where the model's predictions exhibit the most disagreement across possible parameter configurations, focusing on uncertainty. BatchBALD [46] is an extension of BALD and it selects batches of data points to maximize joint information and reduce redundancy in batch selection. Core-Set [43] identifies a subset of data that summarizes the entire dataset, aiming for comparable performance with fewer training examples. BADGE [47] chooses data points based on diverse gradient embeddings, capturing instances that offer varied learning experiences. ACS-FW [48] uses the Frank-Wolfe optimization algorithm to select instances from the pool set whose conic combinations best represent the entire set to promote representativeness. LCMD [31] first finds the largest cluster for representativeness and then enforces diversity by picking the maximum distance point within this cluster.

Active learning on a budget. Active learning on a budget has been studied in [21, 22, 23]. They showed that in this setting, random selection outperforms most deep active learning strategies. This phenomenon is often explained by the poor ability of neural models to capture uncertainty on a small budget. The recently proposed method TypiClust [32] prioritizes typical examples instead of uncertain examples and shows significant improvement over random selection. We consider it as our baseline. Note that with IterPert, we do not propose a new selection rule but instead use prior information to adjust the estimation of the perturbation space. We show that IterPert has significant improvement over TypiClust, but we leave the problem of integrating IterPert with TypiClust as future work.

Perturbation prediction models. CellOracle[29] relies on gene regulatory network inference and conducts linear network propagation of perturbation signals to make predictions. CPA[30] uses a non-linear compositional autoencoder to predict effects but it is restricted to predicting seen perturbations. GEARS [12] is a deep learning model customized for perturbation prediction. It is based on a GNN perturbation and cell encoder with a deep composition layer that simulates multi-gene perturbations on cells, and it features a loss function focusing on differentially expressed genes. Recently, single-cell foundation models have gained popularity and claim to excel at perturbation outcome prediction. Notably, scGPT [13] uses a generative pre-training objective over a massive scRNA-seq atlas and is finetuned on perturbation prediction tasks. However, it requires the perturbed genes to be detected in the scRNA-seq experiment, which is not the case for many perturbations in our data set. Although we use GEARS in this work, the approach is general and applicable to other models.

Active learning for genomics experimental design. Sequential optimal design is increasingly popular in high throughput genomics assays. The main task is to identify genes that maximize an endpoint such as cell proliferation [49, 50]. Note that this setting is highly different from ours, because there, the goal is to identify a data point in the data distribution with the highest response (Bayesian optimization). In contrast, we are interested in selecting points that enable a machine learning model to reduce the overall loss across the data distribution (active learning). The more related work GeneDisco [35] is a benchmark for the sequential design of genetic perturbation experiments, proposing both Bayesian optimization and active learning tasks. The key difference in our work is that we focus on active learning for expensive Perturb-seq, where the response is high-dimensional expression profiles, while GeneDisco focuses on functional genomics CRISPR assays with a single scalar readout. This leads to different base prediction models and a different active learning setting than the one discussed in Section 3.1. Also note that we have included the active learning methods benchmarked in GeneDisco (BADGE, KMeans, BALD) in our baselines, and in this study, our proposed method IterPert has significantly better performance.

6 Discussion

We introduced an iterative Perturb-seq procedure for efficient design of perturbation experiments. We highlighted the challenges of active learning on a budget constraints and evaluated current active learning techniques. Motivated by an initial data analysis, we presented IterPert, a new active learning strategy that incorporates multi-modal priors, achieving over three times the speed of the best baseline.

While IterPert shows promise in designing efficient Perturb-seq experiments, it still faces limitations, and further work is necessary for its practical implementation. For instance, while we strive to simulate a realistic setting *in-silico*, several points of divergence could occur in practice. One such divergence is experimental batch effects, which could be more significant than those considered in our setting. Moreover, while our method is very useful for mapping genome-scale single-gene perturbations, further work is needed to extend this approach to multi-gene (combinatorial) perturbations that are currently intractable to experimentally interrogate in an exhaustive way. Extending the framework to multi-gene perturbations requires higher-order kernels or the use of tensor product spaces, which presents an interesting methodological challenge that we leave for future work. Similarly, extensions to chemical perturbations or optical readouts are also exciting future avenues. More specific to our prior-guided strategy, while our empirical study finds that mean fusion works the best, it is not context-specific. Ideally, different combinations of prior information could be automatically picked in different cell contexts. Lastly, with the increasing interest in models to predict the outcome of perturbations, we expect more base prediction models to become available. While our proposed active learning strategy is compatible with any of these, future work remains to investigate IterPert performance with these methods.

Overall, we believe that the sequential design of Perturb-seq could drastically reduce the experimental cost of understanding a complex space of perturbations, thanks to its sample efficiency, and could help answer central biological questions, such as the effect of multi-gene perturbations.

Acknowledgements and Funding Information

We thank Xinming Tu, Jerry Wang, and Rebecca Boiarsky for feedback throughout the duration of this project that greatly improved this work. We warmly thank Jure Leskovec for valuable discussions and feedback for improving the manuscript. We also thank members of the Regev Lab and the Biological Research | AI development (BRAID) department at Genentech for providing constructive feedback on earlier versions of the results presented in this work.

Romain Lopez, Jan-Christian Hütter, Takamasa Kudo, Antonio Rios, and Aviv Regev are employees of Genentech, and may have equity in Roche. Aviv Regev is a co-founder and equity holder of Celsius Therapeutics and an equity holder in Immunitas. She was an SAB member of Thermo Fisher Scientific, Syros Pharmaceuticals, Neogene Therapeutics, and Asimov until July 31st, 2020.

Code Availability

The raw source code is available at https://github.com/Genentech/iterative-perturb-seq and is released under the Apache 2.0 license. The notebooks to reproduce each figure are provided in https://github.com/Genentech/iterative-perturb-seq/tree/master/reproduce_repo. The python package is available at iterpert. We implemented the source code in PyTorch. The base machine learning model is adapted from https://github.com/snap-stanford/GEARS. The active learning strategy framework is adapted from https://github.com/dholzmueller/bmdal_reg.

References

[1] Hiroaki Kitano. Systems biology: a brief overview. *Science*, 295(5560):1662–1664, 2002.

[2] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *Cell*, 167(7):1853–1866, 2016.

[3] Daniel Schraivogel, Andreas R Gschwind, Jennifer H Milbank, Daniel R Leonce, Petra Jakob, Lukas Mathur, Jan O Korbel, Christoph A Merten, Lars Velten, and Lars M Steinmetz. Targeted perturb-seq enables genome-scale genetic screens in single cells. *Nature Methods*, 17(6):629–635, 2020.

[4] Oana Ursu, James T Neal, Emily Shea, Pratiksha I Thakore, Livnat Jerby-Arnon, Lan Nguyen, Danielle Dionne, Celeste Diaz, Julia Bauman, Mariam Mounir Mosaad, et al. Massively parallel phenotyping of coding variants in cancer with perturb-seq. *Nature Biotechnology*, 40(6):896–905, 2022.

[5] Gavin R Schnitzler, Helen Kang, Vivian S Lee-Kim, Rosa X Ma, Tony Zeng, Ramcharan S Angom, Shi Fang, Shamsudheen Karuthedath Vellarikkal, Ronghao Zhou, Katherine Guo, et al. Mapping the convergence of genes for coronary artery disease onto endothelial cell programs. *bioRxiv*, pages 2022–11, 2022.

[6] Matthew R Nelson, Hannah Tipney, Jeffery L Painter, Judong Shen, Paola Nicoletti, Yufeng Shen, Aris Floratos, Pak Chung Sham, Mulin Jun Li, Junwen Wang, et al. The support of human genetic evidence for approved drug indications. *Nature Genetics*, 47(8):856–860, 2015.

[7] Yosef Buganim, Dina A Faddah, and Rudolf Jaenisch. Mechanisms and models of somatic cell reprogramming. *Nature Reviews Genetics*, 14(6):427–439, 2013.

[8] 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.

[9] Douglas Yao, Loic Binan, Jon Bezney, Brooke Simonton, Jahanara Freedman, Chris J Frangieh, Kushal K Dey, Kathryn Geiger-Schuller, Basak Eraslan, Alexander Gusev, et al. Compressed perturb-seq: highly efficient screens for regulatory circuits using random composite perturbations. *bioRxiv*, pages 2023–01, 2023.

[10] Britt Adamson, Thomas M Norman, Marco Jost, Min Y Cho, James K Nuñez, Yuwen Chen, Jacqueline E Villalta, Luke A Gilbert, Max A Horlbeck, Marco Y Hein, et al. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882, 2016.

[11] 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, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

[12] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, pages 1–9, 2023.

[13] 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*, pages 2023–04, 2023.

[14] Jing Gong, Minsheng Hao, Xin Zeng, Chiming Liu, Jianzhu Ma, Xingyi Cheng, Taifeng Wang, Xuegong Zhang, and Le Song. xTrimoGene: An efficient and scalable representation learner for single-cell rna-seq data. *bioRxiv*, pages 2023–03, 2023.

[15] Burr Settles. Active learning literature survey. 2009.

[16] Greg Schohn and David Cohn. Less is more: Active learning with support vector machines. In *International Conference on Machine Learning*, 2000.

[17] Vishwesh Nath, Dong Yang, Bennett A Landman, Daguang Xu, and Holger R Roth. Diminishing uncertainty within the training pool: Active learning for medical image segmentation. *IEEE Transactions on Medical Imaging*, 40(10):2534–2547, 2020.

[18] Giuseppe Riccardi and Dilek Hakkani-Tur. Active learning: Theory and applications to automatic speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(4):504–511, 2005.

[19] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[20] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[21] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(4):631–644, 2019.

[22] Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. Rethinking deep active learning: Using unlabeled data at model training. In *2020 25th International Conference on Pattern Recognition*, pages 1220–1227. IEEE, 2021.

[23] Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. Parting with illusions about deep active learning. *arXiv:1912.05361*, 2019.

[24] David Feldman, Avtar Singh, Jonathan L Schmid-Burgk, Rebecca J Carlson, Anja Mezger, Anthony J Garrity, Feng Zhang, and Paul C Blainey. Optical pooled screens in human cells. *Cell*, 179(3):787–799, 2019.

[25] Meraj Ramezani, Julia Bauman, Avtar Singh, Erin Weisbart, John Yong, Maria Lozada, Gregory P Way, Sanam L Kavari, Celeste Diaz, Marzieh Haghighi, et al. A genome-wide atlas of human cell morphology. *bioRxiv*, 2023.

[26] Madalina Giurgiu, Julian Reinhard, Barbara Brauner, Irmtraud Dunger-Kaltenbach, Gisela Fobo, Goar Frishman, Corinna Montrone, and Andreas Ruepp. Corum: the comprehensive resource of mammalian protein complexes—2019. *Nucleic Acids Research*, 47(D1):D559–D563, 2019.

[27] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15): e2016239118, 2021.

[28] Gert RG Lanckriet, Minghua Deng, Nello Cristianini, Michael I Jordan, and William Stafford Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Biocomputing*, pages 300–311. World Scientific, 2003.

[29] Kenji Kamimoto, Blerta Stringa, Christy M Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel, and Samantha A Morris. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614(7949):742–751, 2023.

[30] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, page e11517, 2023.

[31] David Holzmüller, Viktor Zaverkin, Johannes Kästner, and Ingo Steinwart. A framework and benchmark for deep batch active learning for regression. *Journal of Machine Learning Research*, 24(164):1–81, 2023.

[32] Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *International Conference on Machine Learning*, 2022.

[33] Dezhi Huang, Naya Ma, Xinlei Li, Yang Gou, Yishuo Duan, Bangdong Liu, Jing Xia, Xianlan Zhao, Xiaoqi Wang, Qiong Li, et al. Advances in single-cell rna sequencing and its applications in cancer research. *Journal of Hematology & Oncology*, 16(1):1–48, 2023.

[34] Tessa Durakis Green, Stefan Peidli, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Jake P Taylor-King, Debora Susan Marks, Augustin Luna, Nils Blüthgen, et al. scperturb: Information resource for harmonized single-cell perturbation data. In *Advances in Neural Information Processing Systems Workshop on Learning Meaningful Representations of Life*, 2022.

[35] Arash Mehrjou, Ashkan Soleymani, Andrew Jesson, Pascal Notin, Yarin Gal, Stefan Bauer, and Patrick Schwab. Genedisco: A benchmark for experimental design in drug discovery. *International Conference on Learning Representations*, 2022.

[36] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.

[37] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7:1–11, 2006.

[38] Elle M Weeks, Jacob C Ulirsch, Nathan Y Cheng, Brian L Trippe, Rebecca S Fine, Jenkai Miao, Tejal A Patwardhan, Masahiro Kanai, Joseph Nasser, Charles P Fulco, et al. Leveraging polygenic enrichments of gene features to predict genes underlying complex traits and diseases. *Nature Genetics*, pages 1–10, 2023.

[39] Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. String: a database of predicted functional associations between proteins. *Nucleic Acids Research*, 31(1): 258–261, 2003.

[40] Kathryn R Geiger-Schuller, Basak Eraslan, Olena Kuksenko, Kushal K Dey, Karthik Jagadeesh, Pratiksha I Thakore, Ozge Karayel, Andrea R Yung, Anugraha Rajagopalan, Ana M Meireles, et al. Systematically characterizing the roles of e3-ligase family members in inflammatory responses with massively parallel perturb-seq. *bioRxiv*, pages 2023–01, 2023.

[41] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.

[42] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409, 2022.

[43] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *International Conference on Learning Representations*, 2018.

[44] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys*, 54(9):1–40, 2021.

[45] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv:1112.5745*, 2011.

[46] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[47] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *International Conference on Learning Representations*, 2020.

[48] Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. *Advances in Neural Information Processing Systems*, 32, 2019.

[49] Aldo Pacchiano, Drausin Wulsin, Robert A Barton, and Luis Voloch. Neural design for genetic perturbation experiments. *International Conference on Learning Representations*, 2023.

[50] Clare Lyle, Arash Mehrjou, Pascal Notin, Andrew Jesson, Stefan Bauer, Yarin Gal, and Patrick Schwab. DiscoBAX: Discovery of optimal intervention sets in genomic experiment design. In *International Conference on Machine Learning*, 2023.

Appendix

In Appendix A, we describe pre-processing steps for each prior source we leverage. In Appendix B, we describe different fusion operators to fuse across prior kernels and report their empirical performance. In Appendix C, we further provide plots of the main experiments that include all error bars since they are withheld for the sake of visibility in the main text. In Appendix D, we discuss the experiments on the genome-scale Perturb-seq data and report the obtained performance metrics.

A Data processing on multi-modal priors

1. Additional Perturb-seq data: we use the essential-wide RPE1 cell line CRISPRi dataset from the same paper [11] as the K562 dataset. Particularly, for each perturbation, we obtain the NTC centered pseudobulk expression profile and use that as the feature embedding. For genome-scale experiment, since we do not have another cell line with genome-scale perturbations, we remove this prior source.

2. Optical pooled screens: [25] conducts a genome-wide optical pooled screen and calculated CellProfiler features for each perturbation. We retrieve each perturbation embedding from https://github.com/broadinstitute/2022_PERISCOPE#downloading-profiles. Notably, we use the median aggregation version. For A549, we used 20200805_A549_WG_Screen_guide_normalized_median_merged_ALLBATCHES_ALLWELLS.csv.gz. For HeLa, we used both DMEM 20210422_6W_CP257_guide_normalized_median_merged_ALLBATCHES____DMEM____ALLWELLS.csv and HPLM 20210422_6W_CP257_guide_normalized_median_merged_ALLBATCHES____HPLM____ALLWELLS.csv.

3. scRNA-seq atlas: we used processed scRNA profiles aggregated from multiple scRNA-seq experiments in [38] (https://github.com/FinucaneLab/pops).

4. Protein structures: we retrieve the protein coding sequence of the corresponding gene perturbation from uniprot and then feed each into ESM-2 15 billion parameter model(https://huggingface.co/facebook/esm2_t48_15B_UR50D) and the output [CLS] token embedding is used as the protein embedding.

5. Protein-protein interaction network: we used the PPI knowledge network from https://arxiv.org/abs/2306.04766, and apply node2vec (https://github.com/eliorc/node2vec) to obtain each gene embedding.

6. Literature: we feed the gene name of each perturbation into BioGPT (https://huggingface.co/microsoft/BioGPT-Large) and we use the [CLS] token embedding as the gene embedding.

B Fusion operator

We experiment with multiple strategies for the FUSION operator, including element-wise operators:

1. Mean operator: $\hat{\mathbf{K}} = \frac{1}{m+1}(\mathbf{K} + \mathbf{K}_1 + \cdots + \mathbf{K}_m)$

2. Max operator: $\hat{\mathbf{K}} = \mathrm{MAX}(\mathbf{K}, \mathbf{K}_1, \cdots, \mathbf{K}_m)$

3. Product operator: $\hat{\mathbf{K}} = \mathbf{K} \times \mathbf{K}_1 \times \cdots \times \mathbf{K}_m$

We also experiment with adaptive kernel aggregation methods. Given the subset of kernel matrix with ground truth at round i called $\mathbf{K}^{(i)}_{\text{truth}}$, we can estimate the kernel alignment scores $KA(\mathbf{K}, \mathbf{K}^{(i)}_{\text{truth}})$, where KA [36] is defined as the cosine similarity between the two matrices (using the inner product canonically induced by Frobenius norm). The *kernel alignment weighted operator* is then defined as

$$\hat{\mathbf{K}} = KA(\mathbf{K}, \mathbf{K}^{(i)}_{\text{truth}}) * \mathbf{K} + KA(\mathbf{K}_1, \mathbf{K}^{(i)}_{\text{truth}}) * \mathbf{K}_1 + \cdots + KA(\mathbf{K}_m, \mathbf{K}^{(i)}_{\text{truth}}) * \mathbf{K}_m.$$

Another learnable operator is to estimate the weights $\alpha, \alpha_1, \cdots, \alpha_m$ by solving a linear regression problem to fit the ground truth sub-kernel from prior sub-kernel using validation dataset at each round i:

$$\mathbf{K}^{(i)}_{\text{truth}} \approx \alpha * \mathbf{K}^{(i)} + \alpha_1 * \mathbf{K}^{(i)}_1 + \cdots + \alpha_m * \mathbf{K}^{(i)}_m,$$

and then use the weights to update the entire kernel

$$\hat{\mathbf{K}} = \alpha * \mathbf{K} + \alpha_1 * \mathbf{K}_1 + \cdots + \alpha_m * \mathbf{K}_m.$$

We report performance comparisons of these different operators in Figure 7. We observe that the mean operator has the best empirical performance. We hypothesize that the reason for this that while the learnable operators can capture context-specific relations among the kernels, their estimation is biased due to the limited size of available data for each round. We also experimented with non-linear integration of kernels, but they easily led to overfitting. In the end, we adopt the mean fusion operator.

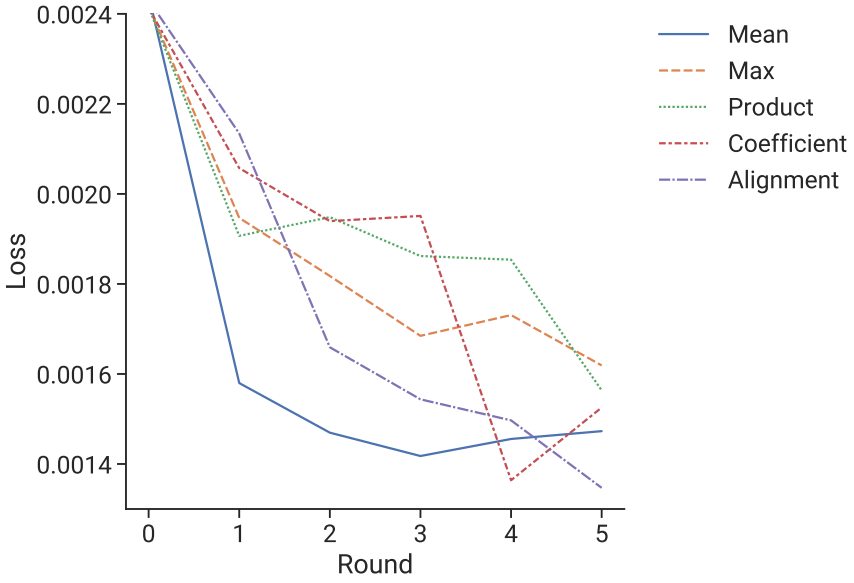

Figure 7: Performance comparison on a seed different from the main experiments across different fusion operators. Mean operator has the best empirical performance.

C Error bars for baselines

Error bars are omitted in Figures 3b and 3c in the main paper to make the plots easier to read. We here report the error bar for Figure 3b (breakdown of active learning baselines) in Figure 8 and the error bar for Figure 3c (breakdown of individual prior information) in Figure 9.

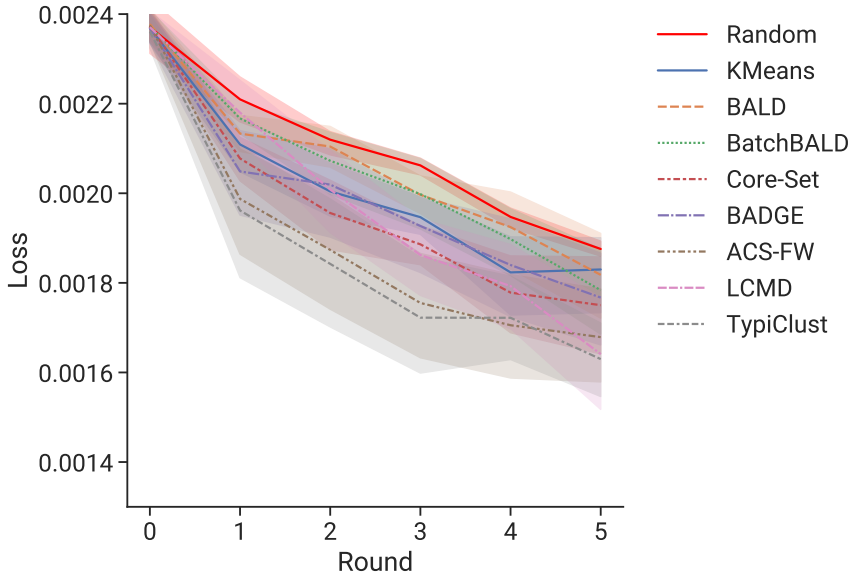

Figure 8: Performance comparison across different baselines with error bar corresponding to 95% confidence interval.

D Baseline performance for genome-scale perturbation screen

In Figure 10, we report the performance of all the baseline state-of-the-art active learning strategies on the genome-scale perturbation screen that were omitted in Figure 5.

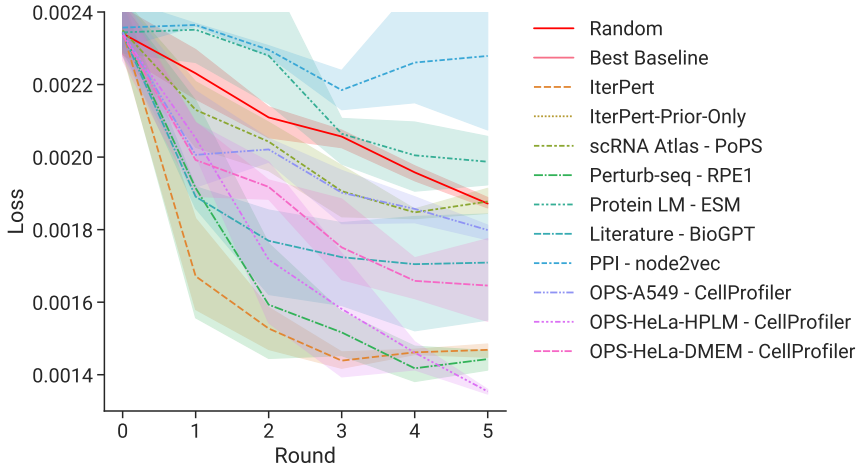

Figure 9: Performance comparison across different prior information with error bar corresponding to 95% confidence interval.

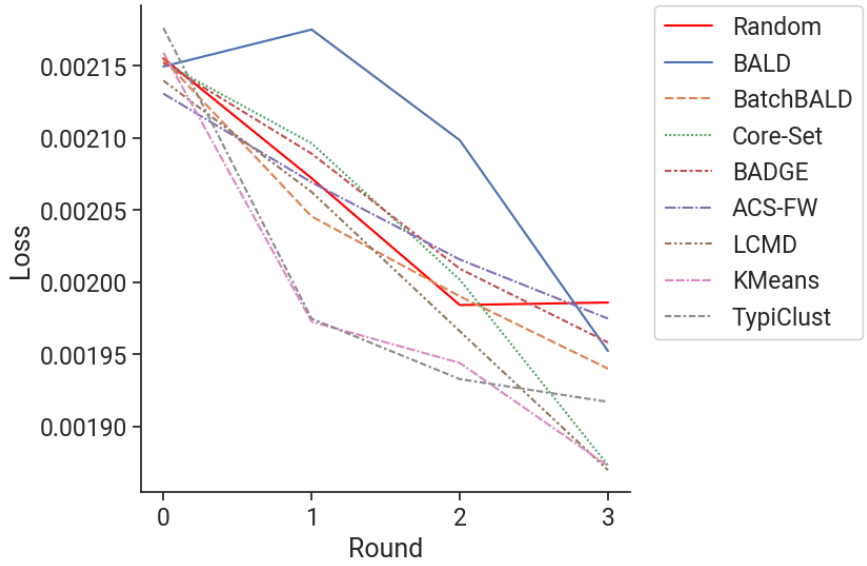

Figure 10: Performance comparison across baseline methods for genome-scale K562 screens with error bar corresponding to 95% confidence interval.