Lifting Biomolecular Data Acquisition

Andrei Slabodkin[*1], **Mattia G. Gollub**[*1], **Kerry Dobbs**[*1], **Xiao-Bing Cui**[*1], **Fang Zhang**[1], **Kristina Gurung**[1], **Elizabeth B. Wood**[†1], **Eli N. Weinstein**[†1,2]

Abstract

One strategy to scale up ML-driven science is to increase wet lab experiments' information density. We present a method based on a neural extension of compressed sensing to function space. We measure the activity of multiple different molecules simultaneously, rather than individually. Then, we deconvolute the molecule-activity map during model training. Co-design of wet lab experiments and learning algorithms provably leads to large gains in information density. We demonstrate with an application to cell therapies, testing multiple protein designs per droplet in a high-throughput assay.

1 Introduction

Machine learning holds dramatic potential to automate and accelerate science. Algorithms can be used to design and learn from complex laboratory experiments, and to extract more information from finite experimental resources (Lindley, 1956; Chaloner & Verdinelli, 1995; Rainforth et al., 2024). As integration deepens, it is unlikely that the optimal experimental designs are those that follow conventional human interpretable protocols. Rather, machine-readable experiments could carry higher information density.

Here, we study the co-design of algorithms and experiments in the context of high throughput biology. We consider experiments that synthesize and test the activity of different biomolecules – we focus on proteins, though the methods are more general. The aim is to learn a mapping from molecule to activity. Such maps are critical for biological understanding, molecular design and disease diagnosis (Frazer et al., 2021; Watson et al., 2023). We seek experiments and learning algorithms that, together, extract as much information as possible under resource constraints.

The stock blend of experiments and algorithms is designed for supervised learning (Yang et al., 2019). First, collect (x, y) pairs recording the activity y of different amino acid sequences x. Then, fit a model, such as a neural network, to predict y from x. This gives an estimate of the sequence-activity map. The challenge is that the information we gain is bottlenecked by experimental resources: the only way to learn more is to collect more (x, y) pairs.

In this article, we pursue an orthogonal route to increasing information: we increase the number of x per y. That is, we test more than one x at the same time, obtaining a composite picture of their activity. As a result, we can no longer assign an activity to each sequence by eye. But we can still learn algorithmically, by deconvoluting the sequence-activity map during training. Indeed, it turns out we can learn much more than if we tested just one x at a time.

Overview and contributions Our approach builds on *compressed screens*, which test multiple different molecules x per measurement y, then reconstruct activity with a LASSO-style algorithm (Yao et al., 2023; Liu et al., 2024). In compressed screens, the goal is to efficiently learn the activity

[*]joint first author. [†] joint senior author. [1] JURA Bio, Boston, MA, USA. [2] Department of Chemistry, Technical University of Denmark, Kgs. Lyngby, Denmark. Correspondence to: Elizabeth B. Wood <ew@jurabio.com>, Eli N. Weinstein <enw@jurabio.com>

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

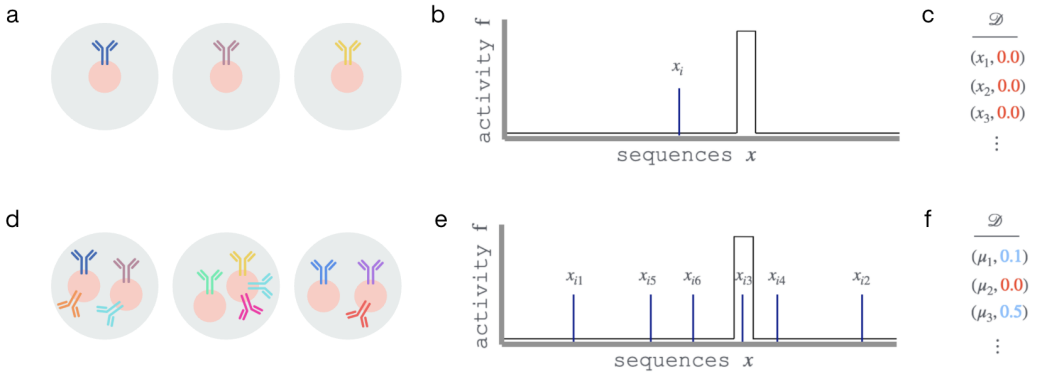


Figure 1: **Lifting data acquisition.** Top row: learning from (x, y) pairs. a. Test one sequence per cell. b. When the sequence-activity map f is sparse (black), most sequences (blue) show no activity. c. So, the resulting dataset contains little information. Bottom row: learning from mixtures $\mu_i(x) = \sum_{j=1}^{m_i} \delta_{x_{ij}}(x)$. d. Deliver multiple sequences x_{ij} to each cell. e. With more sequences tested, we are more likely to see signal. f. The resulting dataset provides more information about f, which we deconvolute during training.

of each molecule within a fixed set. We instead aim to train a model that can generalize, and predict the activity of unobserved molecules.

To do so, we *lift* compressed screens from finite to infinite dimensions. Rather than reconstruct a latent vector, we reconstruct a latent function parameterized by a neural network: the molecule-activity map. To accomplish this, we study a functional, neural extension of compressed sensing (Section 2). We realize this extension experimentally and algorithmically (Section 4). Theoretically, we prove the experimental process lets us reconstruct the molecule-activity map efficiently, under weak assumptions on the underlying biophysics (Section 5). Empirically, we demonstrate the method lets us train more predictive models, compared to testing sequences individually (Section 6). We validate with wetlab experiments on generative model-designed antibodies (scFv CAR-T cell therapy constructs), collecting protein-protein interaction data via droplet-based single cell sequencing, and predicting binding strength against challenging cancer targets (HLA-presented intracellular tumor antigens).

2 Lifting Data Acquisition

Our goal is to learn the relationship $\mathrm{f}(x)$ between biomolecules x and their activity.

A standard way to achieve this is to first experimentally test different sequences, collecting data $(x_1, y_1), \ldots, (x_n, y_n)$, then fit a model $y_i \sim f_\theta(x_i) + \epsilon_i$, where ϵ_i denotes the experimental noise. This returns an estimate $f_{\hat{\theta}} \approx \mathrm{f}$. The challenge is that experimental costs limit the number of datapoints n, and this limits how accurately we can learn f.

Rather than test one sequence at a time, test a mixture of different sequences, $x_{i1}, \ldots, x_{im_i}$. Now, we observe

$$y_i \sim \sum_{j=1}^{m_i} \mathrm{f}(x_{ij}) + \epsilon_i. \tag{1}$$

We assume here the activity of each sequence contributes additively to the overall measured outcome (Yao et al., 2023; Liu et al., 2024).

How can we learn f efficiently? As usual, we parameterize f with a neural network f_θ and fit it to the data. But it is unclear how to choose the mixtures. It is also unclear how to deconvolve each sequence's individual contribution.

Functional, Neural Compressed Sensing We have lifted the learning problem from regression on points to regression on mixtures. Although $\mathrm{f}(x)$ may be arbitrarily complex and nonlinear as a

function of x, observations are linear in the mixture. To see this, rewrite the mixture as a positive measure, $\mu_i(x) \triangleq \sum_{j=1}^{m_i} \delta_{x_{ij}}(x)$. Equation (1) becomes,

$$y_i = \mathrm{f} \cdot \mu_i + \epsilon_i, \tag{2}$$

where $\mathrm{f} \cdot \mu_i \triangleq \int \mathrm{f}(x)\mu_i(x)\mathrm{d}x$ denotes the L^2 inner product on functions. So in function space, learning f is a linear estimation problem.

To take advantage of the linearity, we apply ideas from compressed sensing. Compressed sensing designs limited linear measurements to reconstruct high-dimensional signals (Candes, 2006). It leads us to the following recipe.

Experimentally, make random incoherent measurements. If we test one sequence at a time, then $\mu_i(x) = \delta_{x_i}(x)$, and we only learn about the value of f at a single x_i. This is a maximally *coherent* measurement, meaning $\mu_i(x)$ is concentrated at one point (Figure 1b). It is a very inefficient way to learn a linear model (Candes & Plan, 2011; Russo & Van Roy, 2018). Instead, we should choose μ_i to be a random mixture of many different sequences, giving information about f at many locations. Such $\mu_i(x)$ are *incoherent*, meaning more spread out (Figure 1e). Using incoherent measurements reduces the number of datapoints n needed to learn f (Candes & Plan, 2011).

Algorithmically, exploit prior knowledge for regularization. With limited measurements, the unknown, high-dimensional f is underdetermined. However, we have prior knowledge. In biology, sequence-to-activity maps are often sparse: only x within a small region of sequence space show any activity, $\mathrm{f}(x) > 0$ (Figure 1b). This is particularly true for the hardest and most important design problems, where sequences with a desired activity can be exceedingly rare (Skora et al., 2015; Weinstein et al., 2026b). By using an L^1 regularizer during training, we can penalize non-sparse f and resolve the indeterminacy. The training objective becomes,

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} (f_\theta \cdot \mu_i - y_i)^2 + \lambda \|f_\theta\|_{1,\mathrm{p}} \tag{3}$$

where λ is the regularization strength and $\|f\|_{1,\mathrm{p}} = \int_{\mathcal{X}} |f(x)|\mathrm{dp}$ is the L^1 norm in function space, with base measure $\mathrm{p}(x)$ (Benjamin et al., 2019). Equation (3) is the LASSO but with functions in place of vectors. The unknown function is parameterized with a neural network.

In summary, we have lifted the learning problem from nonlinear regression on sequences to high-dimensional linear regression on measures. From this perspective, we see that testing sequences one at a time is inefficient. We instead test many sequences together, and exploit f's underlying sparsity to reconstruct the sequence-activity map.

3 Related Work

Our method builds on the *lifting trick*, which reformulates a non-convex optimization problem, $\max_x \mathrm{f}(x)$, into a convex but infinite-dimensional problem, $\max_\mu \mathrm{f} \cdot \mu$ (Bach, 2021; Yang et al., 2020). We similarly lift an estimation problem.

Standard compressed sensing reconstructs vectors. It corresponds to the case where $f_\theta(x) = \theta_x$ for $\theta \in \mathbb{R}^{|\mathcal{X}|}$ and $|\mathcal{X}| < \infty$ (Donoho, 2006; Candes & Tao, 2006). For example, x may be the row and column of a pixel, and θ_x its intensity. Our approach instead applies to structured, high dimensional x, such as molecules. It parametrizes $f_\theta(x)$ with a neural network, that can learn features which generalize to unseen x. Previous work has combined neural networks with compressed sensing. They replace the LASSO's ℓ^1 regularizer with a generative model, such as an image generator, and reconstruct a latent representation of the image (Bora et al., 2017; Grover & Ermon, 2019; Wu et al., 2019; Naderi & Plan, 2022). We instead replace the unknown vector with a neural regression model. Previous work has extended compressed sensing from latent vectors to latent functions, to reconstruct images (Adcock & Hansen, 2016; Adcock et al., 2017). They work with low-dimensional x, e.g. a position in $\mathbb{R}^2$ or $\mathbb{R}^3$. The unknowns are wavelet coefficients, or the coefficients on another fixed basis, so there is no feature learning.

Our model is an instance of a hierarchical causal model, with data from many molecules per cell (Weinstein & Blei, 2026). We assume cells are exchangeable and molecules are exchangeable within each cell, and study the causal impact of a molecule-level variable, sequence, on a cell-level variable,

activity. We assume no hidden confounding, since the experimental process of generating and delivering sequences is chemically and biophysically randomized (Section 4).

We build on compressed screens, which apply compressed sensing to biological screening (Yao et al., 2023; Liu et al., 2024; Cleary & Regev, 2020; Kainkaryam & Woolf, 2009). These screens test mixtures of biological sequences or small molecules, then reconstruct their activity via the LASSO or a related linear model. The success of compressed screens across different molecules and assays helps validate our assumption that outcomes depend additively on the activity of each molecule in a mixture. However, compressed screening relies on standard vector compressed sensing, describing each molecule's activity with a separate coefficient. This prevents generalization and limits scalability: to learn the activity of x, one must make multiple measurements of the same x.

We employ function-space regularizers and priors developed for neural networks (Fortuin, 2022; Tran et al., 2022). Our learning objective in Equation (3) corresponds to maximum *a posteriori* (MAP) estimation under a functional neural network prior (Benjamin et al., 2019). As these previous works point out, it is difficult to design and interpret priors on the parameters of a neural network. Instead, we use scientific domain knowledge to place a prior in function space. Unlike many previous neural functional priors, ours is designed to encourage sparsity, rather than smoothness.

4 Lifting Cell-Based Screens

We describe LIFT, a combination of experimental protocols and algorithms for lifting high throughput cell-based screens. In Section 5 we prove LIFT can substantially accelerate learning.

Experimental Design Our goal is to learn the sequence-activity map f, such that we can make accurate activity predictions for any $x \sim \mathrm{p}(x)$. For example, we may want to accurately predict binding for any human antibody. We consider the following laboratory protocol:

Synthesize sequences. We synthesize samples from a defined distribution $\mathrm{p}(x)$ at large scale, using variational synthesis (Weinstein et al., 2026a). In variational synthesis, the distribution $\mathrm{p}(x)$ can be very complex, as it is specified by a generative protein model. The result is a pool of quadrillions of DNA molecules, each of which is a separate sample $x \sim \mathrm{p}(x)$.

Deliver into cells. Sequences are delivered into cells virally. Biophysical models of viral infection describe the number of sequences m_i that enter an individual cell as Poisson, following rare event statistics (Ellis & Delbrück, 1939). The mean of the Poisson is the *multiplicity of infection (MOI)*, $\bar{m}$. We set $\bar{m} > 1$ to add multiple sequences to each cell, each an independent sample from $\mathrm{p}(x)$.

Measure activity. We use droplet-based single-cell sequencing to recover from each cell i both (a) sequences $\{x_{i1}, \ldots, x_{im_i}\}$, and (b) an overall measurement of activity y_i. For example, y_i can be the number of DNA-barcoded targets that are bound to the cell surface.

The result of this experimental protocol is datapoints $(\{x_{11}, \ldots, x_{1m_1}\}, y_1), \ldots, (\{x_{n1}, \ldots, x_{nm_n}\}, y_n)$, rather than datapoints $(x_1, y_1), \ldots, (x_n, y_n)$.

Another complementary strategy is to *overload* droplets during single-cell sequencing, adding multiple cells to each (Datlinger et al., 2021; Yao et al., 2023; Wu et al., 2024). Since activity is measured at the droplet level, this also provides many x_{ij} per y_i. Droplet overloading can be used separately or together with increased viral MOI.

Model and Training From these experiments, we learn a model of f. We posit the data generating process:

$$\theta \sim \pi_\lambda(\theta) = \frac{1}{Z_\lambda} \exp\left(-\lambda \|f_\theta\|_{1,\mathrm{p}}\right) \tag{4}$$

$$m_i \sim \mathrm{Poisson}(\bar{m}) \qquad x_{ij} \sim \mathrm{p}(x) \tag{5}$$

$$y_i \sim \mathrm{Noise}(\sum_{j=1}^{m_i} f_\theta(x_{ij}), \sigma) \tag{6}$$

Equation (4) is a functional prior over the neural network f_θ, with normalizing constant Z_λ. Equation (5) describes how the measurements μ_i are generated: $\mathrm{p}(x)$ is the variational synthesis distribution and $\mathrm{Poisson}(\bar{m})$ is from viral delivery at MOI $\bar{m}$. So, $\mu_i = \sum_{j=1}^{m_i} \delta_{x_{ij}}$ is a sample from an

inhomogeneous Poisson process over sequence space, with rate $\bar{m}\mathrm{p}(x)$. Equation (6) describes the noisy activity measurement. If we assume Gaussian noise, the model's negative joint log likelihood is directly proportional to the functional neural LASSO in Equation (3). For single-cell data we assume negative binomial noise, parameterized by its mean and dispersion (Wang et al., 2025).

We learn θ by MAP estimation, optimizing the log likelihood of the data. To scale to large datasets, we draw minibatches of cells, and optimize θ with Adam with decoupled weight decay (AdamW) (Kingma & Ba, 2015), (Loshchilov & Hutter, 2019). The term $\|f_\theta\|_{1,\mathrm{p}}$ involves an intractable integral. It can be approximated by Monte Carlo (Benjamin et al., 2019),

$$\|f_\theta\|_{1,\mathrm{p}} = \int |f_\theta(x)|\mathrm{p}(x)dx \approx \frac{1}{K}\sum_{k=1}^{K} |f_\theta(x^k)| \tag{7}$$

where $x^k \sim \mathrm{p}(x)$ are drawn computationally from the variational synthesis model.

5 Theory

In this section we study LIFT theoretically. We consider a simplified setting in which LIFT reduces to standard compressed sensing, so that we can analytically compute the measurements' incoherence. We find that when f is sparse, collecting n measurements with LIFT can provide similar estimates as collecting $\bar{m}\,n$ conventional (x, y) datapoints. That is, the effective dataset size scales roughly linearly with MOI.

5.1 Model Reduction

To tractably analyze LIFT, we study a simplified model where the neural network is replaced by a decision tree (Lee & Jaakkola, 2020).

Assumption 5.1 (Decision tree approximation)**.** *Assume f_θ takes the form of a decision tree, with $\theta = w \in \mathbb{R}^D$ and*

$$f_w(x) = \sum_{d=1}^{D} w_d \mathbb{I}(x \in V_d). \tag{8}$$

Here the V_d form a finite partition of the input space: $\cup_{d=1}^{D} V_d = \mathcal{X}$ and $V_d \cap V_{d'} = \emptyset$ for $d \neq d'$, for $D < \infty$.

For discrete x such as molecules, any f_θ can be decomposed as Equation (8). The simplification is that the features V_d are fixed, rather than learned, D is finite, and the weights w_d are unconstrained. Intuitively, we can interpret D as the *effective* size of sequence space, or the number of different types of sequences. We discuss the limitations of this simplification further in Section 7.

Under this simplified f_θ, LIFT reduces to a vector linear model. The proof, in Section A, relies on the observation that μ_i is a Poisson process.

Proposition 5.2 (Reduction to vector compressed sensing)**.** *Assume full support, $\mathrm{p}(x) > 0$ for all $x \in \mathcal{X}$. Under Assumption 5.1, the generative process for y (Equations* (4) *to* (6)*) is equivalent to*

$$w_d \sim \mathrm{Laplace}(0, (\lambda\beta_d)^{-1}) \tag{9}$$

$$a_{id} \sim \mathrm{Poisson}(\bar{m}\beta_d) \tag{10}$$

$$y_i \sim \mathrm{Noise}(w \cdot a_i, \sigma) \tag{11}$$

where $\beta_d \triangleq \mathrm{p}(x \in V_d)$. With Gaussian noise, the MAP estimate coincides with the LASSO estimate,

$$\underset{\tilde{w}\in\mathbb{R}^D}{\mathrm{argmin}} \frac{1}{2}\sum_{i=1}^{n}(y_i - \tilde{w}\cdot\tilde{a}_i)^2 + \tilde{\lambda}\|\tilde{w}\|_1 \tag{12}$$

where $\tilde{w}_d \triangleq \beta_d w_d$, $\tilde{a}_{id} \triangleq a_{id}/\beta_d$, $\tilde{\lambda} \triangleq \lambda\sigma^2$, and $\|\tilde{w}\|_1 = \sum_d |\tilde{w}_d|$ is the ℓ_1 norm.

5.2 Efficiency Gain

Our goal is now to understand the quality of our experimental design. Does random viral delivery of DNA from a variational synthesis library result in efficient reconstruction of f? What MOI should we use? To address this, we apply the theory of compressed sensing. Candes & Plan (2011) show that the quality of an experimental design depends critically on the measurements' *coherence*.

Definition 5.3 (Coherence (Candes & Plan, 2011))**.** *Let $a \sim q(a)$ denote a random vector in $\mathbb{R}^D$ with $\mathbb{E}[aa^\top] = I_D$, the identity. The coherence ν_q is the smallest number such that*

$$\max_d |a_d|^2 \leq \nu_q \tag{13}$$

holds deterministically, or stochastically in the sense that

$$\mathbb{E}[D^{-1}\|a\|_2^2 \mathbb{I}(\max_d |a_d|^2 > \nu_q)] \leq \frac{1}{20} D^{-3/2} \tag{14}$$

$$\mathbb{P}(\max_d |a_d|^2 > \nu_q) \leq D^{-2}, \tag{15}$$

where the expectation and probability are with respect to q.

The coherence determines how much data we need to accurately estimate the true model.

Proposition 5.4 (Informal summary of Theorem 1.1 and 1.2, Candes & Plan 2011)**.** *Assume a well-specified linear model $y_i = \tilde{w} \cdot a_i + \epsilon_i$ where $a_i \overset{iid}{\sim} q(a)$ and $\epsilon_i \sim \mathrm{Normal}(0, \sigma)$. Assume s entries of $\tilde{w} \in \mathbb{R}^D$ are non-zero. Then if we collect*

$$n \geq c\, \nu_q s \log D \tag{16}$$

datapoints, where c is a constant, we can reconstruct $\tilde{w}$ accurately with high probability using the LASSO. In particular, we obtain an estimate $\hat{w}$ with $\|\hat{w} - \tilde{w}\|_2^2 \leq \mathrm{polylog}(D) \frac{s}{n} \sigma^2$.

We can interpret n/ν_q as the *effective* dataset size. With less coherence ν_q, we need fewer datapoints n to achieve high accuracy. Candes & Plan (2011) show the result is quite tight, i.e. we cannot substantially reduce n and still obtain an accurate reconstruction. They also show the result is robust if $\tilde{w}$ is only approximately s sparse, or if the noise is non-Gaussian.

We now examine the coherence of our experimental design. For an interpretable result, we assume there is an even probability of a sequence falling in each region V_d, so $a_{id} \sim \mathrm{Poisson}(\bar{m}/D)$. To compute the dimensionless coherence ν_q, we must standardize this Poisson measurement distribution to be isotropic: we replace a_{id} with $\bar{a}_{id} = (a_{id} - \mathbb{E}[a_d])/\mathrm{Std}[a_d]$. (Technically, this corresponds to modeling mean-subtracted data, $y_i - \mathbb{E}[Y] = \int f_w(x)\mu_i(x)dx - \int f_w(x)\bar{m}\mathrm{p}(x)dx + \epsilon_i$, rather than directly modeling y_i.)

Proposition 5.5. *Assume $\mathrm{p}(x \in V_d) = 1/D$ and Assumption 5.1. If we test sequences $x \sim \mathrm{p}(x)$ individually, obtaining data $(x_1, y_1), \ldots$, the coherence is $\nu_q = D$. If we use standardized Poisson measurements (Equation* (5)*) then,*

$$\nu_q \leq 16 \left(1 + 2\sqrt{\frac{D}{\bar{m}}} + \frac{D}{\bar{m}}\right) \log^2 \delta \tag{17}$$

where $\delta = \max(\left(20\sqrt{2}\, D^2 \sqrt{1 + \frac{2}{D} + \frac{1}{\bar{m}}}\right)^4, 4D^6)$.

Proof in Section B. To interpret this result, first note the effective size of sequence space D can be very large. When the MOI $\bar{m}$ is small compared to D, ν_q is of order $\mathcal{O}(\frac{D}{\bar{m}} \log^2 D)$. Then, increasing the MOI by an order of magnitude, e.g. from 1 to 10, increases our effective data size, n/ν_q, by roughly an order of magnitude. That is, testing mixtures of 10 sequences is roughly as informative as collecting 10 times more (x, y) datapoints. Once $\bar{m} \approx D$, further increases in $\bar{m}$ provide little advantage. At this point, ν_q is of order $\mathcal{O}(\log^2 D)$, versus $\mathcal{O}(\frac{D}{\bar{m}} \log^2 D)$ for low MOI. So in the high MOI regime, we can effectively obtain orders of magnitude more data.

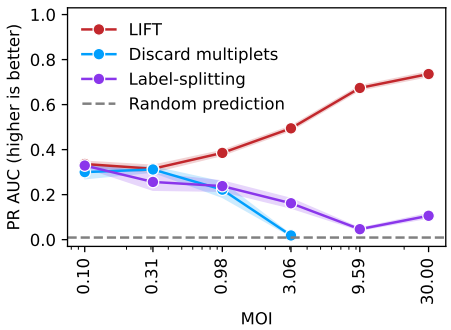


Figure 2: **Synthetic data performance.** The predictive performance of LIFT training versus label-splitting and discard-multiplets training across different experimental designs, on synthetic data. Error bars are s.e. across 10 independent simulations. Discard-multiplets has no points at high MOI, where almost no cells contain a single sequence.

6 Empirical Results

We examine LIFT in simulation and in wet lab experiments, comparing to conventional methods for collecting and training on (x, y) pairs. Our key finding is that the combination of modified experimental design (high MOI) and modified training algorithm (functional neural LASSO) substantially boosts predictive performance.

6.1 Synthetic data

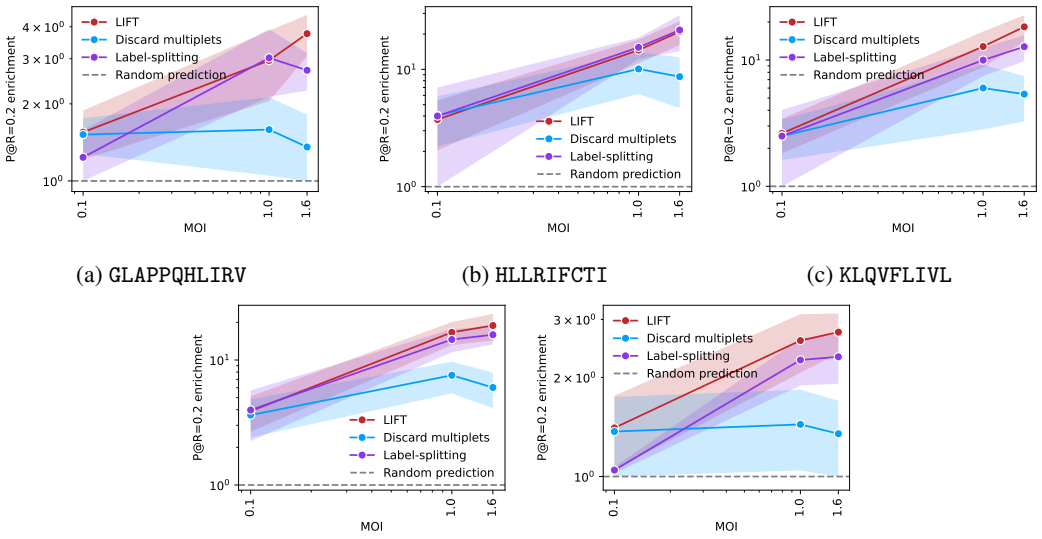
We first examined a synthetic data setting, where the true f is known. We set $\mathrm{p}(x)$ to be a variational synthesis model of human antibody CDRH3 sequences (Weinstein et al., 2026a). Following previous studies of antibody binding, we set $\mathrm{f}(x)$ to increase from zero in the presence of rare amino acid motifs in x (Section C) (Akbar et al., 2021; Pavlović et al., 2021; Weinstein et al., 2026b). The simulation is designed to closely follow laboratory screening workflows, including sorting steps and noise processes.

We generated datasets under different MOI $\bar{m}$. We then compared the LIFT model and training algorithm to a standard model and training algorithm that just uses (x, y) pairs. The architecture (a convolutional neural network) and noise model (negative binomial) were kept the same between the two approaches. We compare to two conventional baselines: *label-splitting*, which splits datapoints $(\{x_{i1}, x_{i2}, \ldots\}, y_i)$ into $(x_{i1}, y_i), (x_{i2}, y_i), \ldots$, and *discard-multiplets*, which trains only on cells containing a single sequence. We evaluated each model's predictive performance on held out $(x, \mathrm{f}(x))$ pairs. To understand the model's ability to predict rare desired properties, we binarize $\mathrm{f}(x)$ at a threshold and report the area under the precision recall curve.

The results demonstrate that using LIFT with high MOI produces substantial performance gains (Figure 2). At low MOI, most cells have at most one sequence, so the data are essentially just (x, y) pairs. Here, LIFT training offers minor benefit. But as the MOI increases, LIFT training improves the PR AUC from 0.34 to 0.74, even as conventional training degrades (to 0.11 for label-splitting at MOI 30, and 0.02 for discard-multiplets at MOI 3). These results demonstrate that co-design of experiments and algorithms is essential: just switching the algorithm, or just increasing MOI, does not lead to performance gains on its own. Instead, it is the combination that provides a major boost in performance.

6.2 Laboratory experiments: TCRm antibodies

We next deployed LIFT at large scale in the wetlab, with the goal of learning challenging and therapeutically relevant sequence-activity relationships. We focus on binding between human antibodies and peptides displayed in complex with human leukocyte antigens (HLA). Developing TCR mimicking (TCRm) antibodies that bind specific peptide-HLA (pHLA) complexes could advance oncology by enabling therapeutic targeting of intracellular tumor antigens. However, discovering such antibod-



(a) GLAPPQHLIRV (b) HLLRIFCTI (c) KLQVFLIVL

(d) QLCPICRAPV (e) RVLHSPPAV

Figure 3: **Laboratory data performance: predicting antibody-pHLA binding.** Factor improvement in the precision (at recall 0.2) over a random predictor (dashed), for LIFT versus the label-splitting and discard-multiplets baselines. Each model was trained separately at different experimental designs (MOI $\bar{m}$). Each plot shows results for one of five separate pHLA targets. Lines are means over 10 cross-validation folds; bands are $\pm$ s.d. across folds, clipped at the random baseline.

ies has been exceptionally difficult (Klebanoff et al., 2023; Skora et al., 2015). This implies the sequence-activity relationship is sparse, making this learning problem a good candidate for LIFT.

We used variational synthesis to create a library of quadrillions of samples from a generative model trained on human post-selection antibody CDRH3s (Weinstein et al., 2026a). The library was assembled into scFv CAR cell therapy constructs and delivered into human cells at MOI above one. Then this library was screened against a panel of fluorescent, DNA-barcoded pHLA targets. We sorted by fluorescence and used single-cell sequencing to recover datapoints $(\{x_{i1}, x_{i2}, \ldots\}, y_i)$ (Section E) (Weinstein et al., 2026b).

To evaluate LIFT, we designed a controlled internal comparison. Our experiment used an MOI just above one, so many cells have just one x. This lets us split the data to mimic different MOI experiments, subsampling cells according to different m distributions. This internal comparison avoids any potential batch variation across experiments. We use splits that set the MOI to $\bar{m} \in \{0.1, 1.0, 1.6\}$.

We trained separate models to predict binding to five oncology pHLA targets, with peptide sequences GLAPPQHLIRV, HLLRIFCTI, KLQVFLIVL, QLCPICRAPV, and RVLHSPPAV, presented on HLA-A*02:01. (The corresponding source antigens are p53, Transmembrane protein 178A, PrePro-IAPP, Livin and MACF1, respectively.) We evaluated the models on cells with $m_i = 1$ sequence, taken from a held-out screen (a biological replicate). We use 10-fold cross-validation over the training screen and exclude from training any x that appears in the test set, to evaluate generalization across sequences. We measure performance as the increase in precision at fixed recall versus a random baseline, which quantifies the expected increase in hit rate provided by the model.

We find the laboratory results follow the simulations (Figure 3, Table 1). At low MOI all methods perform alike. As MOI increases, LIFT steadily improves on every target, while discard-multiplets plateaus or degrades at high MOI. At MOI 1.6, LIFT enriches precision at recall 0.2 by 18–21$\times$ over random on HLLRIFCTI, KLQVFLIVL, QLCPICRAPV, versus 5–9$\times$ for discard-multiplets (e.g. 20.9$\times$ vs. 8.7$\times$ for HLLRIFCTI). On the two harder targets (GLAPPQHLIRV and RVLHSPPAV) all enrichments are small ($\leq 3.8\times$), but LIFT's is still highest. Label-splitting remains competitive at high MOI: it matches LIFT on HLLRIFCTI (21.6$\times$ vs. 20.9$\times$) and although it performs worse on average than LIFT on the other four targets at MOI 1.6 (e.g. 12.8$\times$ vs. 18.3$\times$ for KLQVFLIVL), it is within the variability

across folds. Note a limitation of this laboratory study is that the maximum MOI is still near 1, capping the possible gains we can observe from LIFT (compare the x-axis in Figure 2). Nonetheless, overall, we find that LIFT can offer a substantial performance boost over conventional low MOI methods by modifying both the experimental design and the training algorithm.

7 Discussion

We have proposed LIFT, a system for packing more information into high throughput biomolecular experiments. Theoretical analysis shows LIFT can increase the effective dataset size in proportion to the number of molecules per measurement. When deployed in the wet lab on challenging therapeutic problems, LIFT produced models with an estimated 2-7$\times$ higher hit rate compared to conventional low MOI experiments and training algorithms.

The core idea of LIFT is general. We replace supervised learning from (x, y) datapoints with functional estimation from (μ, y) datapoints. Any experimental setup where multiple inputs contribute additively to the outcome may be lifted in this way. Such situations are abundant in the molecular sciences, where we can physically mix molecules.

Limitations and assumptions LIFT does not assume the molecule-activity map f is linear, but it does assume measurements are linear: the overall activity depends on the sum of the activity of each molecule. This assumption is violated when one molecule modulates the effect of another, e.g. drug-drug interactions (Liu et al., 2024). LIFT also assumes sparsity: molecules with non-zero activity are rare. This assumption is violated when there is more gradation in the outcome. Even when linearity and sparsity are satisfied, LIFT may be limited. For example, using too high an MOI can lead to toxicity and cell death, capping improvements. High droplet overloading also poses challenges (Yao et al., 2023).

Our theoretical analysis required the simplifying assumption that there is a finite number of fixed features that drive activity (Equation (8)). However, in practice, we learned features with a neural network (Section 2). Advances in compressed sensing lend some support to this neural approach (Bora et al., 2017; Naderi & Plan, 2022). Consider the set of vectors $\mathcal{F} \triangleq \{f_\theta \in \mathbb{R}^{|\mathcal{X}|} : \theta \in \Theta\}$, where $f_\theta = (f_\theta(x) : x \in \mathcal{X})$. Note the dimension of f_θ will typically be much larger than the dimension of θ, even for large neural networks, since e.g. protein sequences of length 15 give a dimension $|\mathcal{X}| = 20^{15}$. Furthermore, if the last layer of the neural network $f_\theta(x)$ is linear, $\mathcal{F}$ is a cone. Then, Thm. 2 of Naderi & Plan (2022) can guarantee efficient reconstruction of f from observations $y_i = \mu_i \cdot f_\theta + \epsilon_i$, even when θ is imperfectly optimized. However, the result requires μ be sub-Gaussian, whereas in our application it is Poisson. The result also requires $|\mathcal{X}|$ be finite, rather than infinite (Adcock & Hansen, 2016; Amin et al., 2025). Future work might extend this theory, enabling deeper understanding, and improved experimental designs and algorithms.

Future directions Going forward, LIFT can be extended, its assumptions relaxed, and its robustness improved. There are many methods to robustify linear models or incorporate interactions (Buja et al., 2019; Filzmoser & Nordhausen, 2021; Prasad et al., 2020). They might be extended from the conventional setting of vector linear models, to LIFT's setting of functional neural linear models. For high dimensional y, such as vectors of gene expression, linearity may be more plausible with carefully learned representations (Lotfollahi et al., 2019; Wang et al., 2023). Meanwhile, advances in compressed sensing have shown that the sparsity assumption can be replaced with alternative prior assumptions. One approach is to assume the hidden vector is drawn from a generative model (Bora et al., 2017; Grover & Ermon, 2019; Wu et al., 2019; Naderi & Plan, 2022). Going forward, this approach might also be extended from vectors to neural functions (Garnelo et al., 2018).

Broadly, our work illustrates a path to scaling up biomolecular ML that does not depend just on added experimental cost. We can instead pack and extract more information from wet lab experiments, by designing for algorithmic interpretation.

Funding and Competing Interests

All authors are current or previous employees and/or shareholders of JURA Bio.

References

Adcock, B. and Hansen, A. C. Generalized sampling and infinite-dimensional compressed sensing. *Found. Comput. Math.*, 16(5):1263–1323, October 2016.

Adcock, B., Hansen, A. C., Poon, C., and Roman, B. Breaking the coherence barrier: A new theory for compressed sensing. *For. Math. Sigma*, 5(e4), 2017.

Akbar, R., Robert, P. A., Pavlović, M., Jeliazkov, J. R., Snapkov, I., Slabodkin, A., Weber, C. R., Scheffer, L., Miho, E., Haff, I. H., Haug, D. T. T., Lund-Johansen, F., Safonova, Y., Sandve, G. K., and Greiff, V. A compact vocabulary of paratope-epitope interactions enables predictability of antibody-antigen binding. *Cell Rep.*, 34(11):108856, March 2021.

Amin, A. N., Marks, D. S., and Weinstein, E. N. Biological sequence kernels with guaranteed flexibility. *J. Mach. Learn. Res.*, 26(216):1–63, 2025.

Bach, F. Finding global minima with kernel approximations, 2021.

Benjamin, A. S., Rolnick, D., and Kording, K. Measuring and regularizing networks in function space. In *International Conference on Learning Representations (ICLR)*, 2019.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. Compressed sensing using generative models. In *International Conference on Machine Learning (ICML)*, March 2017.

Buja, A., Brown, L., Berk, R., George, E., Pitkin, E., Traskin, M., Zhang, K., and Zhao, L. Models as approximations I: Consequences illustrated with linear regression. *Statistical Science*, 34(4): 523–544, 2019.

Candes, E. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*, 2006.

Candes, E. J. and Plan, Y. A probabilistic and RIPless theory of compressed sensing. *IEEE Trans. Inf. Theory*, 2011.

Candes, E. J. and Tao, T. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inf. Theory*, 52(12):5406–5425, December 2006.

Canonne, C. A short note on poisson tail bounds, 2019.

Chaloner, K. and Verdinelli, I. Bayesian experimental design: A review. *Stat. Sci.*, 10(3):273–304, 1995.

Cleary, B. and Regev, A. The necessity and power of random, under-sampled experiments in biology. *arXiv [q-bio.QM]*, December 2020.

Datlinger, P., Rendeiro, A. F., Boenke, T., Senekowitsch, M., Krausgruber, T., Barreca, D., and Bock, C. Ultra-high-throughput single-cell RNA sequencing and perturbation screening with combinatorial fluidic indexing. *Nat. Methods*, 18(6):635–642, June 2021.

Donoho, D. L. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, April 2006.

Ellis, E. L. and Delbrück, M. The growth of bacteriophage. *J. Gen. Physiol.*, 22(3):365–384, January 1939.

Filzmoser, P. and Nordhausen, K. Robust linear regression for high-dimensional data: An overview. *Wiley Interdiscip. Rev. Comput. Stat.*, 13(4), July 2021.

Fortuin, V. Priors in bayesian deep learning: A review. *Int. Stat. Rev.*, 90(3):563–591, December 2022.

Frazer, J., Notin, P., Dias, M., Gomez, A., Min, J. K., Brock, K., Gal, Y., and Marks, D. S. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Ali Eslami, S. M., and Teh, Y. W. Neural processes. *arXiv [cs.LG]*, July 2018.

Grover, A. and Ermon, S. Uncertainty autoencoders: Learning compressed representations via variational information maximization. *AISTATS*, 89:2514–2524, 2019.

Kainkaryam, R. M. and Woolf, P. J. Pooling in high-throughput drug screening. *Curr. Opin. Drug Discov. Devel.*, 12(3):339–350, May 2009.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Klebanoff, C. A., Chandran, S. S., Baker, B. M., Quezada, S. A., and Ribas, A. T cell receptor therapeutics: immunological targeting of the intracellular cancer proteome. *Nat. Rev. Drug Discov.*, pp. 1–22, October 2023.

Lee, G.-H. and Jaakkola, T. S. Oblique decision trees from derivatives of ReLU networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Linderman, S. STATS305B: Applied statistics II, 2025.

Lindley, D. V. On a measure of the information provided by an experiment. *Ann. Math. Stat.*, 27(4): 986–1005, December 1956.

Liu, N., Kattan, W. E., Mead, B. E., Kummerlowe, C., Cheng, T., Ingabire, S., Cheah, J. H., Soule, C. K., Vrcic, A., McIninch, J. K., Triana, S., Guzman, M., Dao, T. T., Peters, J. M., Lowder, K. E., Crawford, L., Amini, A. P., Blainey, P. C., Hahn, W. C., Cleary, B., Bryson, B., Winter, P. S., Raghavan, S., and Shalek, A. K. Scalable, compressed phenotypic screening using pooled perturbations. *Nat. Biotechnol.*, pp. 1–13, October 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. scGen predicts single-cell perturbation responses. *Nat. Methods*, 16(8):715–721, August 2019.

Naderi, A. and Plan, Y. Sparsity-free compressed sensing with applications to generative priors. *IEEE J. Sel. Areas Inf. Theory*, 3(3):493–501, September 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library. *arXiv [cs.LG]*, December 2019.

Pavlović, M., Scheffer, L., Motwani, K., Kanduri, C., Kompova, R., Vazov, N., Waagan, K., Bernal, F. L. M., Costa, A. A., Corrie, B., Akbar, R., Al Hajj, G. S., Balaban, G., Brusko, T. M., Chernigovskaya, M., Christley, S., Cowell, L. G., Frank, R., Grytten, I., Gundersen, S., Haff, I. H., Hovig, E., Hsieh, P.-H., Klambauer, G., Kuijjer, M. L., Lund-Andersen, C., Martini, A., Minotto, T., Pensar, J., Rand, K., Riccardi, E., Robert, P. A., Rocha, A., Slabodkin, A., Snapkov, I., Sollid, L. M., Titov, D., Weber, C. R., Widrich, M., Yaari, G., Greiff, V., and Sandve, G. K. The immuneML ecosystem for machine learning analysis of adaptive immune receptor repertoires. *Nature Machine Intelligence*, 3(11):936–944, November 2021.

Prasad, A., Suggala, A. S., Balakrishnan, S., and Ravikumar, P. Robust estimation via robust gradient estimation. *J. R. Stat. Soc. Series B Stat. Methodol.*, 82(3):601–627, July 2020.

Rainforth, T., Foster, A., Ivanova, D. R., and Smith, F. B. Modern bayesian experimental design. *Stat. Sci.*, 2024.

Russo, D. and Van Roy, B. Learning to optimize via information-directed sampling. *Oper. Res.*, 66 (1):230–252, February 2018.

Skora, A. D., Douglass, J., Hwang, M. S., Tam, A. J., Blosser, R. L., Gabelli, S. B., Cao, J., Diaz, Jr, L. A., Papadopoulos, N., Kinzler, K. W., Vogelstein, B., and Zhou, S. Generation of MANAbodies specific to HLA-restricted epitopes encoded by somatically mutated genes. *Proc. Natl. Acad. Sci. U. S. A.*, 112(32):9967–9972, August 2015.

Tran, B.-H., Rossi, S., Milios, D., and Filippone, M. All you need is a good functional prior for bayesian deep learning. *J. Mach. Learn. Res.*, January 2022.

Wang, Y., Shu, Z., Cao, Z., and Grima, R. From noise to models to numbers: Evaluating negative binomial models and parameter estimations in single-cell RNA-seq. *bioRxiv*, pp. 2025.05.05.652189, May 2025.

Wang, Z., Gui, L., Negrea, J., and Veitch, V. Concept algebra for (score-based) text-controlled generative models. *arXiv [cs.CL]*, February 2023.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, August 2023.

Weinstein, E. N. and Blei, D. M. Hierarchical causal models. *J. Mach. Learn. Res.*, 27(37):1–73, 2026.

Weinstein, E. N., Gollub, M. G., Slabodkin, A., Dobbs, K., Cui, X.-B., Gardner, C. L., Grant, R. J., Gurung, K., Bailey, A., Amin, A. N., Church, G. M., and Wood, E. B. Manufacturing-aware generative models enable petascale synthesis of designed DNA. *Nat. Biotechnol.*, pp. 1–9, March 2026a.

Weinstein, E. N., Slabodkin, A., Gollub, M. G., and Wood, E. B. Accelerated learning on large scale screens using generative library models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2026b.

Wu, B., Bennett, H. M., Ye, X., Sridhar, A., Eidenschenk, C., Everett, C., Nazarova, E. V., Chen, H.-H., Kim, I. K., Deangelis, M., Owen, L. A., Chen, C., Lau, J., Shi, M., Lund, J. M., Xavier-Magalhães, A., Patel, N., Liang, Y., Modrusan, Z., and Darmanis, S. Overloading and unpacKing (OAK) - droplet-based combinatorial indexing for ultra-high throughput single-cell multiomic profiling. *Nat. Commun.*, 15(1):9146, October 2024.

Wu, Y., Rosca, M., and Lillicrap, T. Deep compressed sensing. In *International Conference on Machine Learning (ICML)*, May 2019.

Yang, K. K., Wu, Z., and Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods*, 16(8):687–694, August 2019.

Yang, Z., Zhang, Y., Chen, Y., and Wang, Z. Variational transport: A convergent particle-based algorithm for distributional optimization. *arXiv [cs.LG]*, December 2020.

Yao, D., Binan, L., Bezney, J., Simonton, B., Freedman, J., Frangieh, C. J., Dey, K., Geiger-Schuller, K., Eraslan, B., Gusev, A., Regev, A., and Cleary, B. Scalable genetic screening for regulatory circuits using compressed perturb-seq. *Nat. Biotechnol.*, October 2023.

A Proof of Proposition 5.2

We first rewrite the prior (Equation (4)). We have

$$\|f_w\|_{1,\mathrm{p}} = \int |f_w(x)|\mathrm{p}(x)dx = \int \sum_{d=1}^{D} |w_d|\mathbb{I}(x \in V_d)\mathrm{p}(x)dx = \sum_{d=1}^{D} |w_d|\beta_d. \tag{18}$$

So the prior becomes

$$\pi_\lambda(w) = \frac{1}{Z_\lambda} \prod_{d=1}^{D} \exp(-\lambda\beta_d|w_d|) \tag{19}$$

and so $w_d \sim \mathrm{Laplace}(0, (\lambda\beta_d)^{-1})$.

Next we rewrite the measurement process (Equation (5)). Recall $\mu_i = \sum_{j=1}^{m_i} \delta_{x_{ij}}(x)$. Then $\mu_i(x)$ is a sample from an inhomogeneous Poisson process over sequence space $\mathcal{X}$ with intensity $\bar{m}\mathrm{p}(x)$ (e.g. Linderman, 2025). Define,

$$a_{id} \triangleq \int_{x \in V_d} \mu_i(x)dx. \tag{20}$$

Since the V_d are disjoint subsets of $\mathcal{X}$, we find from the properties of Poisson processes that $a_{id} \sim \mathrm{Poisson}(\bar{m}\mathrm{p}(x \in V_d))$ independently.

Finally, we rewrite the mean of the observation process (Equation (6)) as

$$\sum_{j=1}^{m_i} f_w(x_{ij}) = f_w \cdot \mu_i = \sum_{d=1}^{D} w_d \int_{x \in V_d} \mu_i(x)dx = \sum_{d=1}^{D} w_d a_{id}. \tag{21}$$

We have derived the first part of Proposition 5.2.

With Gaussian noise, the MAP estimate of w is,

$$\hat{w} \triangleq \operatorname*{argmin}_{w \in \mathbb{R}^D} \sum_{i=1}^{n} \frac{1}{2\sigma^2}(y_i - w \cdot a_i)^2 + \sum_{d=1}^{D} \lambda\beta_d|w_d| = \operatorname*{argmin}_{w \in \mathbb{R}^D} \sum_{i=1}^{n} \frac{1}{2}(y_i - w \cdot a_i)^2 + \lambda\sigma^2 \sum_{d=1}^{D} \beta_d|w_d| \tag{22}$$

We can reparameterize this optimization problem to work with $\tilde{w} \triangleq w \odot \beta$ where $\odot$ denotes elementwise multiplication. This gives Equation (12). Then, from the solution $\hat{\tilde{w}}$ to Equation (12), we obtain $\hat{w} = \hat{\tilde{w}} \odot \beta^{-1}$, where $\beta_d^{-1} = 1/\beta_d$. To ensure this is well defined, we need $\beta_d > 0$ for all d. This is guaranteed by the assumption that $\mathrm{p}(x) > 0$.

B Proof of Proposition 5.5

First assume we collect data about individual sequences $x \sim \mathrm{p}(x)$ instead of mixtures. Then we have, from the same argument as in Section A, that $a_i \sim \mathrm{Categorical}(1/D, \ldots, 1/D)$. Candes & Plan (2011) show that this has coherence $\nu_q = D$.

Next we consider the coherence of the Poisson measurements (Proposition 5.2). To compute the coherence, we must first write the measurements in the standardized form. Define $r = \bar{m}/D$ and define $q(\bar{a})$ as,

$$a_{id} \overset{iid}{\sim} \mathrm{Poisson}(r) \qquad \bar{a}_{id} \triangleq \frac{a_{id} - r}{\sqrt{r}}. \tag{23}$$

We see that $\bar{a}$ has mean zero and covariance the identity. If the outcome y_i is linear a_i, it is still linear in $\bar{a}_i$.

Now we compute the coherence of $\bar{a}$. First, via a union bound,

$$\mathbb{P}(\max_d |\bar{A}_d|^2 > \nu) \leq D\mathbb{P}(|\bar{A}_d| > \sqrt{\nu}) \tag{24}$$

$$= D\mathbb{P}(|A_d - r| > \sqrt{r\nu}) \tag{25}$$

Using the subexponential bound on the Poisson from Canonne (2019, Theorem 1),

$$\leq 2D \exp(-\frac{r\nu}{2(r+\sqrt{r\nu})}) \tag{26}$$

$$\leq 2D \exp(-\frac{\sqrt{r\nu}}{2(\sqrt{\frac{r}{\nu}}+1)}) \tag{27}$$

$$\leq 2D \exp(-\frac{\sqrt{r\nu}}{2(\sqrt{r}+1)}) \tag{28}$$

where in the last line we use $\nu \geq 1$ (Candes & Plan, 2011).

We require $\mathbb{P}(\max_d |\bar{A}_d|^2 > \nu) \leq D^{-2}$, which we can now see holds so long as

$$\nu_q \geq \left(1+\frac{1}{\sqrt{r}}\right)^2 \log^2\left(4D^6\right) \tag{29}$$

Next we consider the second bound determining the coherence. From Cauchy-Schwarz,

$$\mathbb{E}[D^{-1}\|\bar{A}\|_2^2 \mathbb{I}(\max_d |\bar{A}_d|^2 > \nu)] \tag{30}$$

$$\leq \sqrt{\mathbb{E}[D^{-2}(\|\bar{A}\|_2^2)^2]\mathbb{E}[\mathbb{I}(\max_d |\bar{A}_d|^2 > \nu)]} \tag{31}$$

$$= \sqrt{\frac{1}{D^2}(D\mathbb{E}[\bar{A}_d^4] + (D^2-D)\mathbb{E}[\bar{A}_d^2]^2)\mathbb{P}(\max_d |\bar{A}_d|^2 > \nu)} \tag{32}$$

$$\leq \sqrt{2(D+2+\frac{1}{r})} \exp(-\frac{\sqrt{r\nu}}{4(\sqrt{r}+1)}) \tag{33}$$

where in the last line we plugged in Equation (28) and the kurtosis of a Poisson distribution, $1/r+3$.

We require $\mathbb{E}[D^{-1}\|\bar{A}\|_2^2 \mathbb{I}(\max_d |\bar{A}_d|^2 > \nu)] \leq \frac{1}{20} D^{-3/2}$. This implies

$$\nu \geq 16\left(1+\frac{1}{\sqrt{r}}\right)^2 (\log 20 + \frac{1}{2}\log 2 + \frac{3}{2}\log D + \frac{1}{2}\log(D+2+\frac{1}{r}))^2 \tag{34}$$

Simplifying,

$$\nu \geq 16\left(1+\frac{1}{\sqrt{r}}\right)^2 \log^2(20\sqrt{2}D^2(1+\frac{2}{D}+\frac{1}{rD})^{1/2}) \tag{35}$$

The result follows, after recalling that $rD = \bar{m}$.

C Synthetic Data

In brief, our synthetic data simulations follow the same setup as in Weinstein et al. (2026b), except that instead of providing each cell just one x, we sample m_i values of x_{ij}, with m_i drawn from a Poisson (Equation (5)).

As in Weinstein et al. (2026b), $\mathrm{f}(x)$ is a deterministic function (based on the presence of specific amino acid sequences at specific positions) and we add noise to generate y. In particular, y follows a Negative Binomial distribution parametrized by deterministic 'sequence strength' $\mathrm{f}(x)$. By default, $\mathrm{f}(x) = 1$, and the presence of any of the following patterns increases $\mathrm{f}(x)$ by a multiplier of 10: (i) `"P"` or `"C"` at position 3 (zero-indexed), (ii) `"N"` or `"C"` at position 5, and (iii) `"PC"` or `"SS"` starting at position 6. The final observable value of y is sampled from the Negative Binomial distribution with a mean of $\mathrm{f}(x)$ and variance $\mathrm{f}(x) + \frac{\mathrm{f}(x)^2}{\phi}$ (where ϕ = 2.28, set based on experimental data).

Then, for each MOI value in $0.1, 0.31, 0.98, 3.06, 9.59, 30.0$, we sample N ($N = 50{,}000$) cells, drawing m_i from a Poisson with λ =MOI. For each cell, we sample the corresponding number of sequences from $\mathrm{p}(x)$, along with their observable strengths, as described above. As in Weinstein et al. (2026b), $\mathrm{p}(x)$ is a variational synthesis model of human antibody CDRH3 sequences, described in Weinstein et al. (2026a)). Again, as in Weinstein et al. (2026b), we assume only hits are sequenced, so we select only the cells with total observable strength ≥ 10 and employ a LeaVS correction in

all models, both LIFT and baselines (Section D). The LeaVS correction automatically provides L^1 regularization so we do not add any additional term $\lambda\|f_\theta\|_{1,\mathrm{p}}$ in the loss (Section D). Note that N is fixed, which yields different numbers of non-zero counts and, hence, hits, for different MOIs. We repeated the experiment 10 times, each time generating the data independently. We evaluate precision recall in terms of the model's ability to predict whether $\mathrm{f}(x)$ is greater than 10.

We used a CNN implemented in PyTorch (Paszke et al., 2019), with 16-dimensional amino acid embeddings and a convolutional block with 32 channels, kernel width 5, residual connections and max pooling. All three training methods used the same architecture and hyperparameters: AdamW with a constant learning rate of 10^{-3}, weight decay 5×10^{-3}, and minibatches of 16 observed cells. The unobserved-cell likelihood correction used 512 Monte Carlo droplets per minibatch. We used 10% of the screen droplets for the validation set, and early stopping on validation loss with patience 50 and at most 500 epochs, retaining the checkpoint with the lowest validation loss. Model computations used float32 and likelihood computations used float64. In total, we used $\sim$20 hours on NVIDIA A100 GPUs, across all simulations, for model training.

D LeaVS

We derive a LeaVS correction term following Weinstein et al. (2026b). After sorting, we do not sequence cells with low y values, e.g. $y = 0$, but do know the number of such cells. From Equation (5) and Equation (6) we can compute the corresponding likelihood term as

$$\mathrm{p}(y = 0) = \sum_{m=0}^{\infty} \left[\int \mathrm{p}(y = 0 \mid f(x))\mathrm{p}(x)dx \right]^m \mathrm{Poisson}(m \mid \bar{m}) \tag{36}$$

where $\mathrm{p}(y = 0 \mid f(x))$ is the likelihood under the noise model. We can estimate this term by Monte Carlo.

During training, we add the term $n_0 \log \mathrm{p}(y = 0)$ to our log likelihood, where n_0 is the number of cells with $y = 0$.

Connection to L^1 functional regularization Although the motivation is different, the LeaVS log likelihood correction implicitly applies functional L^1 regularization. To see this, note that with Poisson noise, $y \sim \mathrm{Poisson}(f_\theta(x))$, and $m = 1$, the LeaVS correction becomes

$$\log \int p_\theta(y = 0 \mid x)\mathrm{p}(x)dx = \log \int \exp(-|f_\theta(x)|)\mathrm{p}(x)dx \tag{37}$$

By Jensen's inequality, this upper bounds $-\int |f_\theta(x)|\mathrm{p}(x)dx$, the L^1 regularizer. In practice, we must approximate the integral in Equation (37) with finite samples, which makes the bound even tighter. Indeed, since $f_\theta(x)$ is typically close to zero in practice, Taylor approximation shows

$$\log \int \exp(-|f_\theta(x)|)\mathrm{p}(x)dx \approx \log(1 - \int |f_\theta(x)|\mathrm{p}(x)dx) \tag{38}$$

$$\approx -\int |f_\theta(x)|\mathrm{p}(x)dx \tag{39}$$

In short, the LeaVS correction approximates the L^1 regularizer.

In practice, we found that when using LeaVS, added L^1 regularization was unnecessary for strong performance, and so we do not include any extra L^1 regularization in our experiments.

E Laboratory Experiments

Laboratory experiments on TCRms follow the protocol in Weinstein et al. (2026a), with increased viral titer. The screen sequences only positive hits, so we include a LeaVS correction in all models, both LIFT and baselines (Section D) (Weinstein et al., 2026b). The LeaVS correction automatically provides L^1 regularization so we do not add any additional term $\lambda\|f_\theta\|_{1,\mathrm{p}}$ in the loss (Section D). We use precision-recall estimators described in Weinstein et al. (2026a).

Table 1: **Results on laboratory data.** Precision at recall 0.2, relative to a random predictor (mean ± s.d. over 10 folds). L: LIFT, S: label-splitting, D: discard-multiplets.

target	method	MOI 0.1	MOI 1.0	MOI 1.6
GLAPPQHLIRV	L	1.5±0.3	3.0±0.9	3.8±0.7
	S	1.2±0.3	3.0±0.9	2.7±0.4
	D	1.5±0.2	1.6±0.5	1.4±0.5
HLLRIFCTI	L	3.7±1.7	14.6±3.3	20.9±4.8
	S	4.0±3.0	15.4±3.2	21.6±7.3
	D	4.0±1.8	10.1±3.9	8.7±4.0
KLQVFLIVL	L	2.6±0.8	12.8±4.0	18.3±4.3
	S	2.5±1.5	10.0±2.5	12.8±3.0
	D	2.5±0.9	6.0±3.2	5.4±2.1
QLCPICRAPV	L	3.9±1.2	16.6±3.5	18.9±4.5
	S	4.0±1.7	14.6±3.1	15.9±2.6
	D	3.6±1.2	7.5±2.1	6.0±1.9
RVLHSPPAV	L	1.4±0.4	2.6±0.5	2.7±0.4
	S	1.0±0.0	2.3±0.4	2.3±0.4
	D	1.4±0.4	1.4±0.4	1.3±0.3

To make sure there is no shared information between the training and the test set, we split the data such that no cell and no sequence is present in both sets. To achieve this, we first constructed a bipartite graph where each cell and each sequence is represented by a vertex, and a cell-vertex and a sequence-vertex are connected if this sequence is present in this cell, and then we split the connected components of this graph. We split the training screen (28,951 sequenced cells) into 10 parts by connected component, balanced by total cells and by positive cells for each target, and used each part in turn for validation (10-fold cross-validation). The test set is a separate screen (a biological replicate) that sequenced 30,646 cells. We evaluate performance on the test set in terms of the enrichment of precision at recall 0.2 over a random predictor (Figure 3).

To perform the comparison for varying MOI, we subsampled the full training set to create different MOI subsets. We kept the test set constant for all MOI for comparability. The MOI of the full dataset was estimated as 1.18. We drew subsets of cells to match different MOI, following the simplifying assumption that, conditional on the number of sequences that actually occur in the cell, the MOI $\bar{m}$ does not affect the distribution of x and y.

In detail, since only the hits were sequenced, we first use a truncated Poisson to estimate the MOI $\bar{m}$. Then we assume that if we had used low MOI, our dataset would consist of $\tilde{n}$ samples from the above distribution, where $\tilde{n} = N\mathrm{Poisson}(m > 0 \mid \bar{m})$HIT-RATE with N the total number of cells and HIT-RATE the fraction of all cells that are hits ($y > 0$). To create a dataset with a lower MOI $\bar{m}'$, we subsampled cells following the truncated Poisson distribution with $\lambda = \bar{m}'$: the fraction of cells with k sequences was set to $\mathrm{Poisson}(m = k \mid \bar{m}', m > 0)$, and the total number of hits was $\tilde{n}' = N\mathrm{Poisson}(m > 0 \mid \bar{m}')$HIT-RATE. We did this for each of the 10 folds and each MOI $\in \{0.1, 1.0, 1.6\}$, resampling against a fixed reference number of cells (so MOIs above the screen's own can be reached by resampling).

We used a transformer implemented in PyTorch (Paszke et al., 2019), with four decoder layers, attention dimension 64, two attention heads of dimension 32, positional encodings, and attention dropout 0.05. All three methods used AdamW with peak learning rate 5×10^{-6}, 50 epochs of linear warmup followed by 300 epochs of cosine decay to 10^{-6}, weight decay 0.05, and minibatches of 512 observed droplets. The unobserved-droplet likelihood correction used 4096 Monte Carlo droplets per minibatch. We used exponential moving averages of model weights with decay 0.998. Training ran for at most 2500 epochs with early-stopping patience 1000; checkpoint selection minimized validation loss smoothed by a trailing 20-epoch median. Model computations used bfloat16 and likelihood computations used float64. In total, we trained 450 models (3 MOIs × 3 methods × 5 targets × 10 folds), using ∼180 hours on NVIDIA A100 GPUs.

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 claims are supported by theory, simulations, and laboratory experiments.

   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: Explicit discussion of limitations throughout, and summarized in the Discussion.

   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: [Yes]

Justification: Assumptions are provided in the main text and proofs in the appendix.

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: Full details in the appendices.

   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: [No]

Justification: We are unable to release proprietary wetlab datasets. However we **do** release full code implementing the method and reproducing the core experimental results on synthetic data, see supplementary material. The wetlab protocol used to generate the data is also publicly available.

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: See main text and appendices C and E.

   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: In figure captions and appendices.

   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: Appendix C and E.

   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: No ethics issues raised.

   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: [Yes]

    Justification: The work has direct impact on drug discovery, as discussed in sections on wetlab experiments and the discussion.

    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 pose such risks.

    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: We use and cite PyTorch for models.

    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: [Yes]

    Justification: We release documented code implementing the method, in the supplement.

    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: 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: 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.
    - 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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

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.