In silico design of epigenetic reprogramming payloads

Lucas Seninge [1] **Conner Kummerlowe** [1] **David L. Reynolds** [1] **Nicholas J. Bernstein** [1] **Jacob C. Kimmel** [1]

Abstract

Cell types and states can be reprogrammed by activating combinations of transcription factors (TFs). However, the TF sets that reprogram cells from one state to another are unknown in the general case. There are $> 10^{16}$ plausible TF sets in the human genome, making experimental search intractable and motivating *in silico* approaches to search this hypothesis space. Here, we describe a probabilistic model to design reprogramming interventions trained on a large corpus of single cell reprogramming data. Our model achieves strong performance on cell state and function prediction tasks and performance exhibits a data scaling law. Using our model in a simulated lab-in-the-loop, we were able to design successful reprogramming interventions significantly faster than pure experimental approaches.

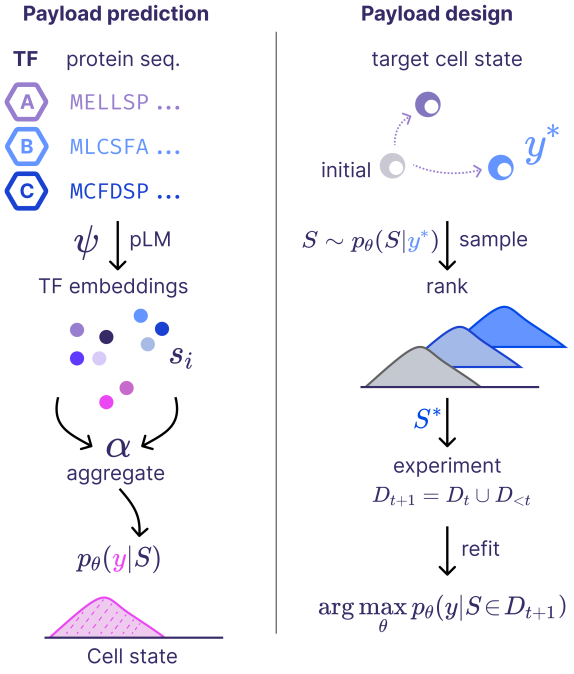

Figure 1. Ambrosia learns a probabilistic model to *predict* the effect or reprogramming payloads on cell state by transfer learning from molecular foundational models. Given a target cell state, Ambrosia then *designs* new payloads through a generative procedure that navigates a massive design space.

1. Introduction

All cells in the human body contain the same DNA code, yet they perform a wide variety of functions. Through diverse epigenetic codes, human cells execute distinct programs from a common genome, analogous to the control flow that guides the execution of specific functions in a large software codebase. Reprogramming the epigenetic code to elicit desired cell functions is both an important therapeutic challenge and a fundamental problem in molecular biology. Activating "payloads" of only 1-6 TFs is one approach to dramatically reprogram cell type and state, sufficient to convert adult cells to embryonic stem cells, skin cells to neurons, or restore youthful features in old cells (Takahashi & Yamanaka, 2006; Vierbuchen et al., 2010; Roux et al., 2022). However, the payloads that evoke a mapping between two arbitrary cell states A and B are unknown in most cases.

Even if payloads are limited to ≤ 6 TFs, there are $\approx 10^{16}$ combinations within the 2,000 TFs encoded by the human genome. The highest throughput experimental methods use single cell genomics to evaluate the effect of $10^2 - 10^3$ payloads in parallel, yielding a gene expression measurement for each (Norman et al., 2019; Joung et al., 2023). Here, our goal is to develop *in silico* reprogramming models that can design payloads S that achieve a desired cell state y given data from these single cell screens.

In particular, we focus on the case of combinatorial predictions where the individual TFs s_i in a payload $S = (s^{(1)}, \ldots, s^{(K)})$ may have been observed experimentally. Our task can be viewed as a special case of the more general problem of predicting the effects of genetic perturbations on cell state (Ji et al., 2021). Prior work has addressed the general perturbation problem with a variety of discrimina-

[*] Equal contribution [1] NewLimit Inc., South San Francisco, CA, USA. Correspondence to: Jacob C. Kimmel <jacob@jck.bio>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

tive and generative methods (Appendix A) trained on small data sets where few combinations are observed ($\approx 10^2$). In practice, trivial baseline approaches demonstrate the best performance in this setting, suggesting that an effective framework has yet to be invented and existing data may be too small for highly parameterized models (Ahlmann-Eltze & Huber, 2024; Li et al., 2025).

Here, we develop a probabilistic modeling approach to design TF payloads to achieve desired cell states and functions. Our approach takes advantage of transfer learning from protein foundation models (Lin et al., 2023) and learns to generate payloads given only a sparse sampling of the combinatorial TF space. We take advantage of a unique combinatorial reprogramming dataset that is an order of magnitude larger in scale than those previously available. We experimentally demonstrate that our method is superior to existing baselines for multiple cell state prediction tasks, performance scales with the size of our training set, and an active learning campaign can accelerate payload design in a retrospective campaign.

2. Approach

2.1. Task construction

We consider the scenario where we are given a dataset $\mathcal{D} = \{S_i, y_i\}_{i=1}^N$ where $S_i = (s^{(1)}, \ldots, s^{(K)})$ are TF payloads composed of $1, \ldots, K$ TFs $s \in \Omega$, where Ω is the set of observed individual TFs, and y_i is a scalar or vector representation of cell state. In particular, we focus on the *combinatorial* prediction scenario where all of the effects for individual TFs are observed, such that all sets containing a single TF are in the training set ($\{\{s\}, \mid s \in \Omega\} \in \mathcal{D}$). We wish to perform two distinct tasks: (1) estimate the distribution $p(y|S)$ to predict the effect of TF payloads on cell state and (2) design TF payloads to achieve a desired cell state by sampling from the posterior $S \sim p(S|y)$.

For the first **payload prediction** task, we perform classic 5-fold cross-validation across TF payloads in our dataset and evaluate the performance of models on an unseen test set. Cross validation folds are constructed so that all payloads are included in exactly one test set. We measure performance using both absolute prediction error (control scaled error, CSE; Pearson correlation coefficient, PCC) and rank-based measures (AUPRC). For a lab-in-the-loop setting, it's common to interrogate the top M payloads where M is set by experimental bandwidth, so the ability to rank payloads and assign binary "hit" labels is the most realistic task.

For the second **payload design** task, we construct an active learning benchmark where we perform successive experiments, each testing a group of payloads S attempting to achieve a target cell state y^*. We use models at each stage to recommend the payloads S to test in the next round and measure the fraction of "hits" that achieve a desired cell state y^* discovered with the goal of discovering more hits in fewer rounds with a strong model and sampling method for $S \sim p(S|y)$.

2.2. Ambrosia *in silico* reprogramming models

Here, we introduce a probabilistic *in silico* reprogramming model we call **Ambrosia**. Ambrosia leverages transfer learning from a pre-trained protein language model $\psi : l \rightarrow s$ to generate TF representations s from protein sequences l (Lin et al., 2023). To represent multi-TF payloads, Ambrosia aggregates TFs within a set S using an aggregation operation $\xi = \alpha(s^{(1)}, \ldots, s^{(K)})$, where ξ is a latent vector representation of the set S and α is an aggregation operation. This approach is inspired by work on deep set learning (Zaheer et al., 2017). In practice, we implemented α as a sum operation over TF representations, though in principle, any permutation invariant operator could be used, including attention mechanisms. This allows Ambrosia models to initialize from rich representations of TF biology, reducing subsequent tasks to learning a distribution of cell states conditioned on payload representations.

We learn the conditional distribution

$$p_\theta(y|S) = f_\theta(S)$$

where f_θ is implemented as a neural network with Monte Carlo dropout for uncertainty estimation (Gal & Ghahramani, 2016). Practically, we implemented f_θ using a 3-layer neural network with hidden layers of sizes $\{512, 128\}$. Each layer is paired with a ReLU activation and a dropout layer to allow for regularization and uncertainty estimation. We optimize models with Adam to minimize a mean-squared error (MSE) loss, approximating the optimization of a Gaussian log-likelihood for $p_\theta(y|S)$ in this setting (Fig. 1). For uncertainty estimation, we perform 100 forward passes through the model with dropout active and parametrize $p_\theta(y|S)$ as an empirical Gaussian distribution from these samples.

2.3. Payload design

We employ the trained model $p_\theta(y|S)$ to *design* new payloads S using two schemes. In the following, we assume y is a scalar value. In the first **constrained setting**, we model the scenario where a researcher has a small and finite number of TF payloads S that they can evaluate in the next round of experiments. Here, we nominate payloads S^* to achieve a desired cell state y^* through exhaustive evaluation $S^* = \arg\max_S p(y^*|S)$. In practice, we define y^* as a region of the cell state variable $y > \tau$, or $y^*_{y>\tau}$.

In the second **unconstrained setting**, we model the scenario where a researcher has an unconstrained set of TF payloads to test, or an intractably large finite set. Here, we nominate payloads S^* by sampling $S \sim p(S|y)$ through a Markov Chain Monte Carlo (MCMC) approach with a Metropolis-Hasting optimization procedure. We assume a simple uniform prior $p(S)$ across the discrete set of TF payloads where $|S| \leq 3$ (up to 3-TFs combinations). This is a practical expedient as our datasets do not include any payloads with more than 3 TFs, but relaxing this assumption is trivial. Algorithmically, we propose TF payloads by sampling $p(S)$ and draw samples $S \sim p(S|y^*)$ using the following Metropolis-Hasting acceptance rule:

$$A(S \rightarrow S') = \frac{p(S'|y^*)}{p(S|y^*)}$$

where S is the last sampled payload and S' is a payload sampled at the current iteration. We estimate $p(S|y^*) \propto p(y^*|S) \cdot p(S)$ by Bayes rule given that we sample with a uniform prior $p(S) \propto 1$. We use Monte Carlo dropout to sample $y \sim p(y|S)$, allowing us to estimate $p(y^*_{y>\tau}|S)$ empirically (**Algorithm 1**). This procedure is readily extensible to the case of arbitrarily large TF payloads, or the incorporation of synthetic TF sequences.

Algorithm 1 Ambrosia payload design algorithm

```
**Input:** Dataset $\mathcal{D} = \{(S_i, y_i)\}_{i=1}^n$, threshold $\tau$, number of iterations $N$, set of observable TFs $\Omega$, maximum payload size $K$, number of payloads to generate $M$

Fit model $p_\theta(y \mid S)$ on $\mathcal{D}$
Define payload space $\mathcal{S} = \bigcup_{i=1}^K \binom{\Omega}{i}$
Define proposal distribution $\phi(S' \mid S) = \mathcal{U}(\mathcal{S})$
Define scoring function $\pi(S, \tau) := p_\theta(y > \tau \mid S)$
Initialize state $S \sim \phi(S)$
for $t = 1$ to $N$ do
    Sample $S' \sim \phi(S' \mid S)$
    $\alpha \leftarrow \min\left(1, \frac{\pi(S',\tau)}{\pi(S,\tau)}\right)$
    Sample $u \sim \mathcal{U}(0, 1)$
    if $u < \alpha$ then
        $S \leftarrow S'$
    end if
    Record $S_t \leftarrow S$, and $p_t \leftarrow \pi(S, \tau)$
end for
Sort $\{S_t\}$ by $p_t$ in descending order

Construct pool $\Phi \subset \Omega$ from the top-ranked samples
$i = 0$
while $|\Phi| < M$ do
    $\Phi \leftarrow \Phi \cup \{s^{(j)} \in S_t\}_i$
    $i \leftarrow i + 1$
end while
Output: $\Phi$
```

2.4. Baselines

We selected baseline methods based on recent benchmarking studies for cell perturbation prediction (Ahlmann-Eltze & Huber, 2024; Li et al., 2025). The Additive and Mean methods below were reported as the state-of-the-art across both studies.

Additive model: The current best performing baseline for combinatorial prediction is an additive model, simply denoted as $f(S) = \sum_i^K y_{s_i}$ where y_{s_i} is the observed value of y for TF s_i. In our probabilistic notation, this model can be expressed as $p(y|S) = \mathbf{1}[\sum_i^K y_{s_i}]$ where $\mathbf{1}[\cdot]$ is a Dirac delta distribution.

Mean model: A constant model that predicts the mean of the training data values for y is likewise reported as a strong baseline for predicting unseen perturbation effects, $f_{\text{Mean}}(S) = \frac{1}{|\mathcal{D}|} \sum_i^{|\mathcal{D}|} y_i$. In our probabilistic notation, this can be expressed as $p(y|S) = \mathbf{1}[\mathbb{E}_{y \sim \mathcal{D}_{\text{train}}}[y]]$.

2.5. Ablations

Ambrosia-Linear: We train a linear model to predict reprogramming effects from protein embeddings of TFs: $f(S) = W\xi_S$ where W is a matrix of weights and ξ_S is a matrix of ESM2 embeddings. Again, in our probabilistic notation such a model can be framed as $p(y|S) = \mathbf{1}[W\xi_S]$. This is an ablation of our **Ambrosia** model eliminating the non-linear logic and uncertainty estimation components.

2.6. Datasets

We trained *in silico* reprogramming models on two datasets.

K562: We first used a public combinatorial CRISPR inhibition screening dataset ("**K562**") in K562 cells covering 236 genetic perturbations including 105 unique gene targets to offer an accessible comparison (Norman et al., 2019). To our knowledge, this is the most commonly used dataset for benchmarking perturbation prediction models (Roohani et al., 2024; Lotfollahi et al., 2023).

NLMT-cx0001: We also used **NLMT-cx0001**, a proprietary dataset activating 6503 TF sets containing 580 unique TFs across 3.6M primary human T cells with single cell RNA-seq read-outs. Reprogramming payloads in NLMT-cx0001were transiently activated, mimicking the "dose" of reprogramming that is typically achieved with an mRNA medicine. Data were generated in primary cells because they a more reliable model of human biology than the immortalized cell lines that are common in public domain datasets.

NLMT-cx0001 is more than an order-of-magnitude larger than any other existing single cell perturbation datasets in primary cells, providing us an opportunity to measure the performance of perturbation prediction models in a much larger data regime (Peidli et al., 2024). All TF sets in NLMT-cx0001 are tested across $\geq$ 5 unique human donors and represented in $\geq$ 50 cells. NLMT-cx0001 contains not only gene expression profiles induced by each payload, but also a functional measure of each payload's impact on T cell growth in culture. NLMT-cx0001 was collected across several screening rounds. As a result, we detail our strategy to mitigate experimental batch effects during modeling in section B.8.

3. Experiments

3.1. Ambrosia predicts reprogramming of cell state & function

We first measured the performance of Ambrosia and baseline methods on the **payload prediction** task using multiple distinct representations of cell state y. In both datasets, we predicted a compressed 50-dimensional PCA representation of gene expression y_E and scalar "gene set scores" y_G that represent target cell states of interest. For the K562 dataset, we predicted a gene set related to cell growth ("mTOR activity") and for the NLMT-cx0001dataset, we predicted a stem central memory T cell score (Tscm) that has been associated with stronger T cell responses in cancer and infectious disease settings (Gattinoni et al., 2017). Maximizing the Tscm score represents a task highly relevant to therapeutics development.

For the NLMT-cx0001 dataset, we also predicted a fitness score y_F that measures the ability of T cells to respond to stimulation and grow in culture. We constructed a rank based metric for the gene set and function tasks by designating the top 25% of all scores as "hits" and measuring the area under the precision recall curve (AUPRC) for each model on each score.

We found that Ambrosia and the ablated variants were the best performing methods in both datasets across all of the tasks reported here (Table 1, 2, 3). As previously reported, the Additive baseline also demonstrated meaningful performance across tasks. Ambrosia models performed well across datasets and across gene expression and cell *function* tasks. Given that the function measurements were collected with an orthogonal measurement system, these results argue that the Ambrosia method is generally applicable. We found that Ambrosia models excelled at predicting large, non-additive effects in combinatorial payloads (Fig. 2A). For example, Ambrosia models provided significantly more accurate predictions for the effect of payloads containing three "Yamanaka Factors" (**O**CT4, **S**OX2, **K**LF4; OSK) than baseline methods (Fig. 2B).

Model	**K562**		**NLMT-cx0001**	
	CSE [↓]	Cosine [↑]	CSE [↓]	Cosine [↑]
Mean	0.86	0.60	1.23	0.43
Additive	0.42	0.86	0.93	0.68
Ambrosia-Linear	0.22	**0.95**	0.47	**0.80**
Ambrosia	**0.21**	0.92	**0.37**	0.79

Table 1. Performance comparison on the *cell state* task across datasets.

Model	**K562**			**NLMT-cx0001**		
	CSE [↓]	PCC [↑]	AUPRC [↑]	CSE [↓]	PCC [↑]	AUPRC [↑]
Mean	0.44	0.00	0.27	0.84	0.00	0.27
Additive	0.31	0.80	0.87	0.86	0.78	0.82
Ambrosia-Linear	0.12	**0.86**	0.87	0.33	0.83	0.84
Ambrosia	**0.11**	0.85	**0.88**	**0.20**	**0.90**	**0.89**

Table 2. Performance comparison on the *gene set* task across datasets.

3.2. Ablation experiments

Ambrosia consists of three key components relative to baselines: (1) transfer learning from a protein language model (pLM), (2) a learned non-linear mapping from pLM representations to cell state effects, and (3) an uncertainty estimation procedure implemented through dropout. We performed ablation experiments to disentangle the benefits of the core components. Our Ambrosia-Linear model contains only the first component, replacing the latter two with a linear and deterministic mapping. We found that the Ambrosia-Linear model was superior to external baseline methods, indicating that transfer learning from pLMs provides value even with a low capacity model. The full Ambrosia model was superior to the ablated variant, suggesting that a high capacity model and uncertainty estimation improve *in silico* reprogramming performance.

We also trained models using ProtT5 pLM representations (Elnaggar et al., 2020) rather than ESM2 embeddings and found that performance was strong in both cases (Fig. 8). This result suggests that Ambrosia is not overly dependent on the properties of one particular foundation model's TF representation.

Model	**CSE [↓]**	**PCC [↑]**	**AUPRC [↑]**
Mean	0.95	0.00	0.27
Additive	2.26	0.62	0.70
Ambrosia-Linear	0.56	0.71	0.73
Ambrosia	**0.23**	**0.80**	**0.79**

Table 3. Performance comparison on the *function* prediction task in the NLMT-cx0001dataset. The full Ambrosia model demonstrates superior performance to baselines and ablated models.

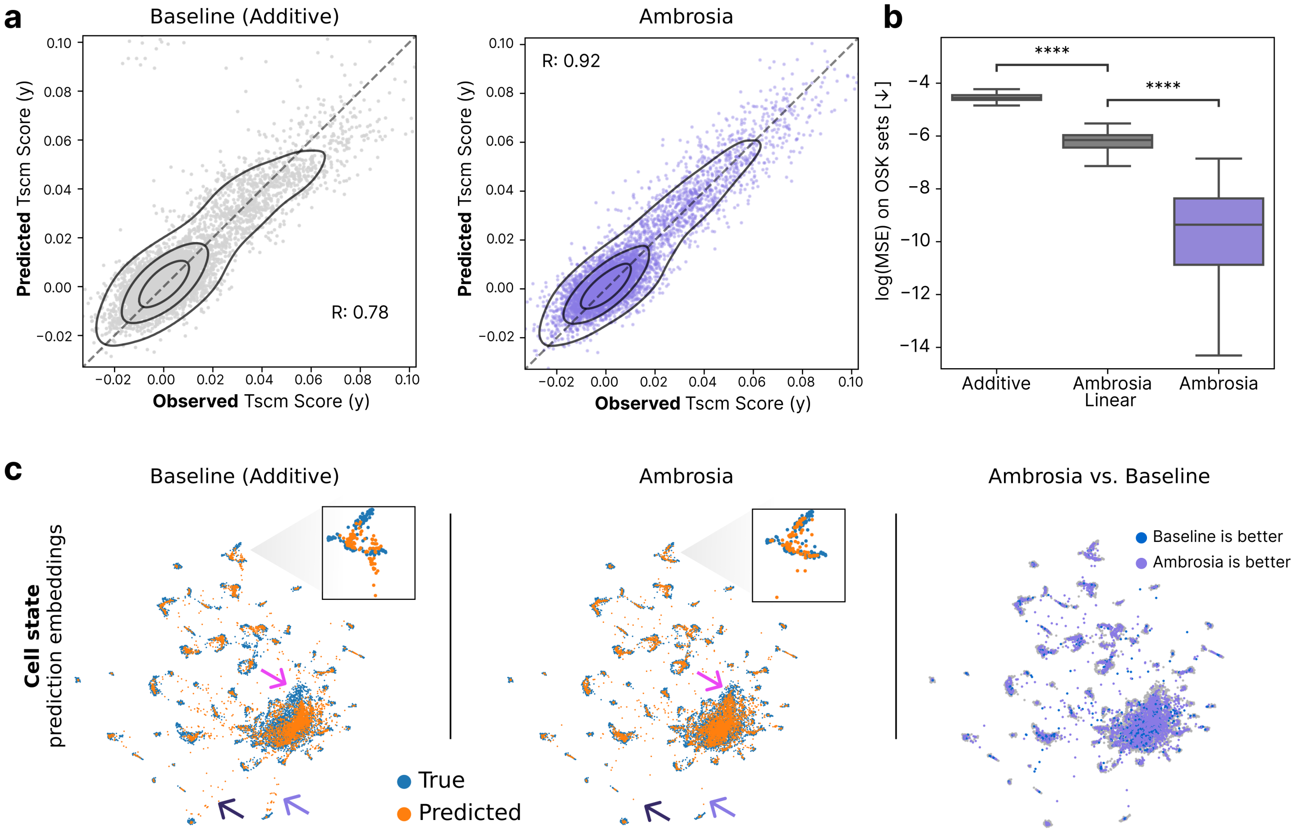

Figure 2. **Payload prediction performance: (A)** Performance of the additive baseline and Ambrosia *in silico* reprogramming models on the Tscm task. Each point is one payload. Data shown are for the NLMT-cx0001dataset. **(B)** Ambrosia models provide superior predictions of combinatorial payload effects relative to ablated approaches (Ambrosia-Linear) and the top baseline method (Additive). Payloads containing the Yamanaka Factor combination OSK alongside other factors are shown as an example (****: $p < 10^{-4}$; Mann Whitney U-test). **(C)** Ambrosia models provide higher fidelity predictions on the **cell state** task relative to the best baseline (Additive). Predictions are shown in a UMAP embedding where each point represents the predicted effect of one payload. Inset panels and color coded arrows highlight regions of perturbation space where Ambrosia models offer superior predictions.

3.3. *In silico* reprogramming exhibits a data scaling law

Generative models have been shown to exhibit scaling laws in other data domains, including natural language and computer vision. As the amount of training data available grows, model performance tends to increase (Kaplan et al., 2020). The scale of NLMT-cx0001 provides us one of the first opportunities to test if these laws are present in the single cell genomics perturbation prediction domain. To investigate, we trained Ambrosia models on data subsets $D_p \subset \mathcal{D}$ where $p \in [0, 1]$ is a proportion of the data used and measured performance on the payload prediction task. We constructed an initialization dataset $D_I \subset \mathcal{D}$ containing all single TF payloads and joined it with each data subset D_p.

We discovered that Ambrosia model performance improves as a function of data scale across multiple metrics (Fig. 3). Performance follows a log-linear trend with high correlation ($r > 0.8$), mirroring behavior in other domains. We imagine that this behavior may have been overlooked in earlier studies due to the small scale of public datasets. We hypothesize that these trends will extrapolate to larger data scales for TF payloads, and likewise emerge for other types of genetic perturbations in single cell genomics data.

3.4. Designing reprogramming interventions

Given a trained *in silico* reprogramming model $p_\theta(y|S)$, we wish to *design* payloads S^* that optimize for a target cell state y^*. To evaluate Ambrosia on this task, we trained models on the NLMT-cx0001 dataset to design payloads that maximized a therapeutically relevant Tscm score. We then designed payloads in the **constrained setting** through exhaustive computation of $p_\theta(y|S) \forall S \in \mathcal{D}_{\text{test}}$ and subsequent ranking. Ambrosia models nominated "hit" payloads that

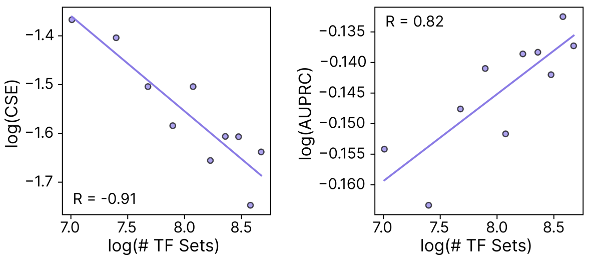

Figure 3. **Scaling law:** *In silico* reprogramming prediction performance exhibits a data scaling law across multiple metrics (control-scaled error, left; AUPRC for hit detection, right). Each point represents a single training and prediction run for an Ambrosia model at a particular data scale D_p.

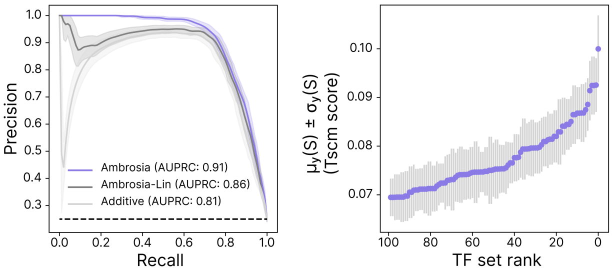

Figure 4. **Designing payloads: (Left)** *In silico* reprogramming models were evaluated on a payload design task to maximize a Tscm score in the NLMT-cx0001dataset. Ambrosia models outperformed ablated versions (Ambrosia-Linear) and the top baseline (Additive). **(Right)** Payload rankings from an Ambrosia model with 95% confidence intervals. Many payloads have similar rank when uncertainty is considered, empowering researchers to build more rationale experimental designs.

maximized the Tscm score more effectively than baselines (Fig. 4A).

Unlike baseline approaches, Ambrosia models provide uncertainty estimates for predicted effects. Qualitatively, many payload designs within the top 100 for the Tscm task are predicted to have effectively equal performance when accounting for uncertainty (Fig. 4B). These uncertainty estimates allow researchers to make more effective experimental design decisions. For example, researchers hoping to maximize activity may be more interested in ranking perturbations by the upper confidence bound, rather than the maximum posterior estimate. In other scenarios, researchers may use uncertainty estimates to weight the amount of effort expended to test each hypothesis in the ranked list, with more certain hypotheses receiving more resourcing. We explore the first of these scenarios in an active learning campaign below.

3.5. Ambrosia accelerates reprogramming discoveries with a lab-in-the-loop

In silico reprogramming methods have the potential to accelerate payload discoveries through a lab-in-the-loop workflow. In this setting, a model is trained on a set of data D_t, then used to prioritize the payloads to test in the next experimental round D_{t+1}. At each iteration t, the number of "hits" or desirable payloads discovered is used as a metric of success. The model $p_\theta(y|S)$ is retrained after each round of new data is collected. This scenario is analogous to active learning or Bayesian Optimization. If successful, a lab-in-the-loop workflow will improve upon the discovery rate of a random baseline.

We deployed Ambrosia in an active learning campaign across the NLMT-cx0001 dataset to optimize a therapeutically relevant T stem central memory (Tscm) cell state $y^*_{y>\tau}$. We constructed this task to represent a realistic experimental setting where the researcher must design a pool of individual TFs Φ to test in each experimental iteration t. We assume that the experimental system allows the researcher to then test all k-TF combinations containing $s \in \Phi$. This reflects the most common experimental methods in the field where payloads are constructed using either pooled molecular cloning or pooled delivery (Norman et al., 2019; Roux et al., 2022).

We initialized models with a dataset $D_0 \subset \mathcal{D}$ containing all single TF perturbations and 10% of multi-TF payloads. At each iteration $t \in [1, 5]$, we constructed a pool for the next experimental round Φ_t by designing payloads with Ambrosia models. Ambrosia was used to estimate the top payloads that remain to be tested $S \in \mathcal{D} \setminus D_t$ to maximize the target state y^*. We then assembled the pool Φ_t by greedily adding unique TFs within the top ranked combinations until $|\Phi_t| = 70$. We constructed a set of payloads D_{Φ_t} composed of TFs in the pool, then built the training dataset for the next round as $D_{t+1} = D_t \cup D_{\Phi_t}$ Intuitively, we add all payloads that contain only TFs in the chosen pool to the dataset for the next round. We then measured the cumulative fraction of hits in the dataset recovered by iteration t. The sampling procedure for Ambrosia to estimate top payloads S was varied across two settings.

We first evaluated performance in the **constrained setting** where we designed payloads in each cycle through exhaustive likelihood estimation across a small, finite set of possible payloads where we have ground truth data. This best represents a scenario where researchers have a limited hypothesis space of payloads to test due to experimental constraints (Methods 2.3). We designed payloads using two different acquisition strategies $a(S)$ to rank payload candidates: (1) the maximum predicted effect (MPE; $a_{\text{MPE}}(S) = \mu_y(S)$ or (2) the upper confidence bound (UCB; $a_{\text{UCB}}(S) = \mu_y(S) + \sigma_y(S)$).

We found that Ambrosia models enabled active learning in the constrained setting with performance superior to a random baseline (Fig. 5). The UCB acquisition function performed modestly better than the MPE function. In future work, we hope to explore if learning the conditional distribution $p(y|S)$ may improve payload design performance over a simple point estimate $\mathbb{E}[y|S]$.

We next performed active learning in the **unconstrained setting** where we design payloads by sampling the posterior $S \sim p_\theta(S|y)$ with an MCMC approach. This represents the scenario where researchers have an infinite or intractably large hypothesis space, as is the case for synthetic TF design or searching payloads that contain many unique TFs. For this setting, we defined our target cell state as the top 10% of the Tscm score distribution up to that iteration ($\tau = Q_{90\%}(y_t)$). We restricted our MCMC procedure to only 10,000 samples to model the realistic scenario where exhaustively computing $p(y|S)$ estimates across the entire search space is intractable. There are $> 10^6$ payloads possible in our experimental setup (Methods 2.3), so this represents sampling $< 1\%$ of the possible payloads.

We used our method described in **Algorithm 1** and found that Ambrosia models were likewise sufficient to accelerate the discovery of hit payloads in this setting (Fig. 5). Performance was in fact comparable to MPE ranking in the constrained setting, suggesting that our MCMC procedure is quite efficient. It's difficult to assess the quality of all samples generated by our model, as we only have ground truth data for a small fraction of the payload space. These results nonetheless suggest that our generative procedure is sufficient to accelerate biological discoveries in a lab-in-the-loop setting and many generated designs are high quality. All of these results were recapitulated using a second T effector gene set score (Gattinoni et al., 2017) computed in NLMT-cx0001. The T effector score contains low gene set overlap with our primary Tscm score and the two are poorly correlated, suggesting that our active learning results are reproducible across multiple, orthogonal design tasks (Fig. 10).

4. Conclusions

Here, we introduce a modeling approach (Ambrosia) for *in silico* reprogramming, a special case of the more general perturbation prediction problem in single cell genomics. We demonstrate that Ambrosia models produce performant predictions of reprogramming effects on cell state and function by transfer learning from protein language models, with results superior to leading baseline methods. Leveraging a unique single cell reprogramming dataset (NLMT-cx0001) with a much larger scale than prior reports, we discovered that *in silico* reprogramming exhibits a *data scaling law*, similar to other emerging biological domains such as nucleic acid and protein sequence modeling. We believe this phenomenon is likely to emerge in other cases of the perturbation prediction problem in single cell genomics as well, but has likely been difficult to observe due to the small scale of public datasets. Our results suggest that larger scale single cell perturbation datasets and transfer learning from molecular foundation models will unlock meaningful performance in perturbation prediction ("virtual cell") models (Bunne et al., 2024).

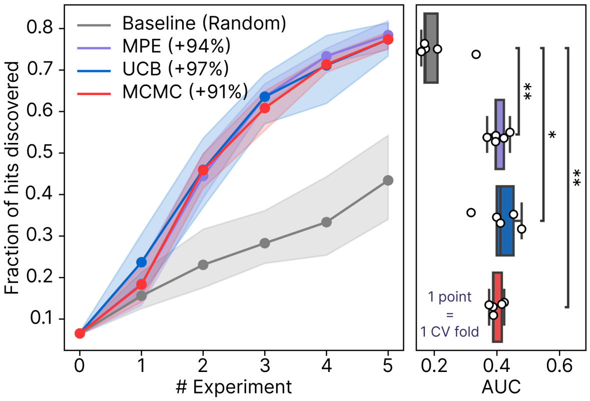

Figure 5. **Active learning:** Active learning campaigns using Ambrosia models relative to a random baseline. (Left) Ambrosia models accelerate the discovery of payloads S^* that achieve a target cell state y^* ("hits"). All Ambrosia design strategies are superior to a random baseline. Upper confidence bound (UCB) sampling is modestly superior to maximum posterior estimates (MPE) in the constrained setting. Markov Chain Monte Carlo (MCMC) demonstrated performance close to UCB/MPE even in the unconstrained setting. **(Right)** Ambrosia methods had significantly higher area under the curve (AUC) than the baseline (*: $p < 0.05$, **: $p < 0.01$; Mann Whitney U-test).

The ultimate goal of building *in silico* reprogramming models is to design payloads that induce target cell states. We found that Ambrosia models were able to improve the rate of designing hit payloads in multiple lab-in-the-loop settings. Reprogramming payload design space is too large for exhaustive *in silico* ranking procedures to be used absent some *a priori* constraint on the space (e.g. number of unique TFs, payload size). As experimental methods improve, these constraints cease to be a laboratory requirement, motivating *in silico* design approaches that can address the full extent of payload opportunities. Through a generative MCMC sampling procedure, Ambrosia models accelerated payload design in this emerging *unconstrained* setting as well. This generative approach opens the door to the design of reprogramming payloads within intractably large spaces, and even the design of entirely synthetic TFs.

In this work, we have demonstrated only a single possible implementation of a more general approach: transfer learning from molecular foundation models to design reprogramming payloads. In the future, we hope to explore models that incorporate a diversity of molecular representations learned in foundation models across the central dogma (DNA, RNA, protein). Likewise, we plan to extend the underlying Ambrosia architecture to employ inductive biases like attention operations to build more effective models. While we have constrained our work here to designing payloads composed of pre-defined, natural TFs, our modeling approach generalizes in principle to future synthetic TF design tasks. Our results to date support the conclusion that deploying *in silico* reprogramming models has the potential to accelerate payload discovery, unlocking the ability to rationally engineer cell states.

Impact Statement

This work employs machine learning tools to expedite the design of genetic interventions to engineer cell state and function. We imagine that these tools can accelerate the design of therapeutics to treat diseases. Today, epigenetic reprogramming is recognized as a powerful technology, but the design of therapeutics has been limited by the largely trial-and-error process. These applications could provide meaningful health benefits for society in the long-term. We believe there are few negative impacts of accelerating the design of reprogramming payloads.

References

Ahlmann-Eltze, C. and Huber, W. Regression on Latent Spaces for the Analysis of Multi-Condition Single-Cell RNA-Seq Data. 2024.

Bertin, P., Rector-Brooks, J., Sharma, D., Gaudelet, T., Anighoro, A., Gross, T., Martínez-Peña, F., Tang, E. L., Suraj, M. S., Regep, C., Hayter, J. B. R., Korablyov, M., Valiante, N., Sloot, A. v. d., Tyers, M., Roberts, C. E. S., Bronstein, M. M., Lairson, L. L., Taylor-King, J. P., and Bengio, Y. RECOVER identifies synergistic drug combinations in vitro through sequential model optimization. *Cell Reports Methods*, 3(10), October 2023. ISSN 2667-2375. doi: 10.1016/j.crmeth.2023.100599. URL https://www.cell.com/cell-reports-methods/abstract/S2667-2375(23)00251-5. Publisher: Elsevier.

Bunne, C., Roohani, Y., Rosen, Y., Gupta, A., Zhang, X., Roed, M., Alexandrov, T., AlQuraishi, M., Brennan, P., Burkhardt, D. B., Califano, A., Cool, J., Dernburg, A. F., Ewing, K., Fox, E. B., Haury, M., Herr, A. E., Horvitz, E., Hsu, P. D., Jain, V., Johnson, G. R., Kalil, T., Kelley, D. R., Kelley, S. O., Kreshuk, A., Mitchison, T., Otte, S., Shendure, J., Sofroniew, N. J., Theis, F., Theodoris, C. V., Upadhyayula, S., Valer, M., Wang, B., Xing, E., Yeung-Levy, S., Zitnik, M., Karaletsos, T., Regev, A., Lundberg, E., Leskovec, J., and Quake, S. R. How to Build the Virtual Cell with Artificial Intelligence: Priorities and Opportunities, October 2024. URL http://arxiv.org/abs/2409.11654. arXiv:2409.11654 [q-bio].

Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., and Wang, B. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, August 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02201-0. URL https://doi.org/10.1038/s41592-024-02201-0.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., BHOWMIK, D., and Rost, B. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv*, 2020. doi: 10.1101/2020.07.12.199554. URL https://www.biorxiv.org/content/early/2020/07/21/2020.07.12.199554.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, October 2016. URL http://arxiv.org/abs/1506.02142. arXiv:1506.02142 [stat].

Gattinoni, L., Speiser, D. E., Lichterfeld, M., and Bonini, C. T memory stem cells in health and disease. *Nature Medicine*, 23(1):18–27, January 2017. ISSN 1546-170X. doi: 10.1038/nm.4241. URL https://www.nature.com/articles/nm.4241. Publisher: Nature Publishing Group.

Hao, M., Gong, J., Zeng, X., Liu, C., Guo, Y., Cheng, X., Wang, T., Ma, J., Zhang, X., and Song, L. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21(8):1481–1491, August 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02305-7. URL https://doi.org/10.1038/s41592-024-02305-7.

Huang, K., Lopez, R., Hütter, J.-C., Kudo, T., Rios, A., and Regev, A. Sequential Optimal Experimental Design of Perturbation Screens Guided by Multi-modal Priors. *bioRxiv*, 2023. doi: 10.1101/2023.12.12.571389. URL https://www.biorxiv.org/content/early/2023/12/13/2023.12.12.571389.

Ji, Y., Lotfollahi, M., Wolf, F. A., and Theis, F. J. Machine learning for perturbational single-cell omics. *Cell Systems*, 12(6):522–537, June 2021. ISSN 2405-4720. doi: 10.1016/j.cels.2021.05.016.

Joung, J., Ma, S., Tay, T., Geiger-Schuller, K. R., Kirchgatterer, P. C., Verdine, V. K., Guo, B., Arias-Garcia, M. A., Allen, W. E., Singh, A., Kuksenko, O., Abudayyeh, O. O., Gootenberg, J. S., Fu, Z., Macrae, R. K., Buenrostro, J. D., Regev, A., and Zhang, F. A transcription factor atlas of directed differentiation. *Cell*, 186(1):209–229.e26, January 2023. ISSN 1097-4172. doi: 10.1016/j.cell.2022.11.026.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling Laws for Neural Language Models, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv:2001.08361 [cs, stat].

Li, C., Gao, H., She, Y., Bian, H., Chen, Q., Liu, K., Wei, L., and Zhang, X. Benchmarking AI Models for In Silico Gene Perturbation of Cells, January 2025. URL https://www.biorxiv.org/content/10.1101/2024.12.20.629581v2. Pages: 2024.12.20.629581 Section: New Results.

Liberzon, A., Birger, C., Thorvaldsdóttir, H., Ghandi, M., Mesirov, J. P., and Tamayo, P. The Molecular Signatures Database Hallmark Gene Set Collection. *Cell Systems*, 1(6):417–425, December 2015. ISSN 2405-4712. doi: 10.1016/j.cels.2015.12.004. URL https://doi.org/10.1016/j.cels.2015.12.004. Publisher: Elsevier.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/10.1126/science.ade2574. Publisher: American Association for the Advancement of Science.

Lopez, R., Tagasovska, N., Ra, S., Cho, K., Pritchard, J. K., and Regev, A. Learning causal representations of single cells via sparse mechanism shift modeling. *Conference on Causal Learning and Reasoning*, 2023.

Lotfollahi, M., Klimovskaia Susmelj, A., De Donno, C., Hetzel, L., Ji, Y., Ibarra, I. L., Srivatsan, S. R., Naghipourfar, M., Daza, R. M., Martin, B., Shendure, J., McFaline-Figueroa, J. L., Boyeau, P., Wolf, F. A., Yakubova, N., Günnemann, S., Trapnell, C., Lopez-Paz, D., and Theis, F. J. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, June 2023. ISSN 1744-4292. doi: 10.15252/msb.202211517. URL https://www.embopress.org/doi/full/10.15252/msb.202211517. Publisher: John Wiley & Sons, Ltd.

Norman, T. M., Horlbeck, M. A., Replogle, J. M., Ge, A. Y., Xu, A., Jost, M., Gilbert, L. A., and Weissman, J. S. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, August 2019. doi: 10.1126/science.aax4438. URL https://www.science.org/doi/10.1126/science.aax4438. Publisher: American Association for the Advancement of Science.

Peidli, S., Green, T. D., Shen, C., Gross, T., Min, J., Garda, S., Yuan, B., Schumacher, L. J., Taylor-King, J. P., Marks, D. S., Luna, A., Blüthgen, N., and Sander, C. scPerturb: harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, March 2024. ISSN 1548-7105. doi: 10.1038/s41592-023-02144-y. URL https://www.nature.com/articles/s41592-023-02144-y. Publisher: Nature Publishing Group.

Replogle, J. M., Norman, T. M., Xu, A., Hussmann, J. A., Chen, J., Cogan, J. Z., Meer, E. J., Terry, J. M., Riordan, D. P., Srinivas, N., Fiddes, I. T., Arthur, J. G., Alvarado, L. J., Pfeiffer, K. A., Mikkelsen, T. S., Weissman, J. S., and Adamson, B. Combinatorial single-cell CRISPR screens by direct guide RNA capture and targeted sequencing. *Nature Biotechnology*, 38(8):954–961, August 2020. ISSN 1546-1696. doi: 10.1038/s41587-020-0470-y. URL https://doi.org/10.1038/s41587-020-0470-y.

Roohani, Y., Huang, K., and Leskovec, J. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, June 2024. ISSN 1546-1696. doi: 10.1038/s41587-023-01905-6.

Roux, A. E., Zhang, C., Paw, J., Zavala-Solorio, J., Malahias, E., Vijay, T., Kolumam, G., Kenyon, C., and Kimmel, J. C. Diverse partial reprogramming strategies restore youthful gene expression and transiently suppress cell identity. *Cell Systems*, pp. S240547122200223X, June 2022. ISSN 24054712. doi: 10.1016/j.cels.2022.05.002. URL https://linkinghub.elsevier.com/retrieve/pii/S240547122200223X.

Szałata, A., Hrovatin, K., Becker, S., Tejada-Lapuerta, A., Cui, H., Wang, B., and Theis, F. J. Transformers in single-cell omics: a review and new perspectives. *Nature Methods*, 21(8):1430–1443, August 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02353-z. URL https://doi.org/10.1038/s41592-024-02353-z.

Takahashi, K. and Yamanaka, S. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell*, 126(4):663–676, August 2006. ISSN 0092-8674. doi: 10.1016/j.cell.2006.07.024.

Tirosh, I., Izar, B., Prakadan, S. M., Wadsworth, M. H., Treacy, D., Trombetta, J. J., Rotem, A., Rodman, C., Lian, C., Murphy, G., Fallahi-Sichani, M., Dutton-Regester, K., Lin, J.-R., Cohen, O., Shah, P., Lu, D., Genshaft, A. S., Hughes, T. K., Ziegler, C. G. K., Kazer, S. W., Gaillard, A., Kolb, K. E., Villani, A.-C., Johannessen, C. M., Andreev, A. Y., Van Allen, E. M., Bertagnolli, M., Sorger, P. K., Sullivan, R. J., Flaherty, K. T., Frederick, D. T., Jané-Valbuena, J., Yoon, C. H., Rozenblatt-Rosen, O., Shalek, A. K., Regev, A., and Garraway, L. A. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science (New York, N.Y.)*, 352 (6282):189–196, April 2016. ISSN 1095-9203. doi: 10.1126/science.aad0501.

Vierbuchen, T., Ostermeier, A., Pang, Z. P., Kokubu, Y., Südhof, T. C., and Wernig, M. Direct conversion of fibroblasts to functional neurons by defined factors. *Nature*, 463(7284):1035–1041, February 2010. ISSN 1476-4687. doi: 10.1038/nature08797. URL https://www.nature.com/articles/nature08797. Number: 7284 Publisher: Nature Publishing Group.

Wolf, F. A., Angerer, P., and Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, February 2018. ISSN 1474-760X. doi: 10.1186/s13059-017-1382-0. URL https://doi.org/10.1186/s13059-017-1382-0.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

A. Related Work

Predicting the effect of combinatorial perturbations in single-cell screens has been the focus of recent work. Variational autoencoder (VAEs) (Lotfollahi et al., 2023; Lopez et al., 2023) and graph neural network (GNNs) (Roohani et al., 2024) architectures have been employed to predict the outcome of unseen chemical and genetic perturbations at the single-cell level. Recently, single-cell foundation models (scFMs) have been used to leverage the large corpus of public single-cell data and learn powerful gene representations that allow fine-tuning of models for perturbation prediction (Cui et al., 2024; Hao et al., 2024). However, recent benchmarking efforts have highlighted that most of these approaches fail to outperform simple baselines and linear models (Ahlmann-Eltze & Huber, 2024; Li et al., 2025). These results taken together highlight the need for thorough benchmarking of proposed models versus simple approaches, across datasets of various sizes and diverse prediction tasks (Szałata et al., 2024).

The problem of iterative experimental design for perturbation screens has also been tackled in the past, often through the lens of active learning (Bertin et al., 2023; Huang et al., 2023). However, these approaches have been typically limited to either studying model loss improvements or prioritizing a list of combinatorial perturbations for testing in lower-throughput, arrayed-format screens. The problem of pooled screen design – particularly relevant for combinatorial genetic screens (Replogle et al., 2020)– has received limited attention, in part because of data acquisition constraints.

B. Methods

B.1. Generating gene set scores

We computed gene set scores from the gene expression data by using the procedure described in (Tirosh et al., 2016) and implemented in (Wolf et al., 2018). We obtained gene sets from the Molecular Signature Database (MSigDB) (Liberzon et al., 2015). We used an mTOR gene set (M5924) for the **K562** dataset, and constructed a Tscm score for the **NLMT-cx0001** dataset as the difference between gene set scores of up- and down-regulated genes in T stem cell memory vs T effector memory (M8429, M8441). As an additional validation, we used a T effector score (M8428), ensuring that it is orthogonal to the Tscm score (Fig. 10).

B.2. Generating cell state representations

To generate cell state representations, we followed standard single cell RNA-seq best practices to produce PCA embeddings (Wolf et al., 2018). We first performed library size normalization and log-transformed the normalized data using the standard $\log(x + 1)$ transform. We selected the top 5000 genes using a coarse grained normalized variance estimation procedure (Wolf et al., 2018) and generated PCA representations from these genes alone. We compressed our representation to the top $k = 50$ PCs.

B.3. Generating fitness scores for the NLMT-cx0001dataset

We derived fitness effects for the payloads in NLMT-cx0001by computing the average $\log_2$ fold-change (log2fc) of read or cell counts between pre- and post-reprogramming conditions in our reprogramming experiments. For some experiments in NLMT-cx0001, the fitness score is derived from read counts in a bulk DNA-seq assay. In others, the fitness score is derived from single cell counts in single cell RNA-seq. We have found the two measurements are strongly correlated and treat them as interchangable for the purposes of modeling in this work.

B.4. Aggregation of cells at the perturbation level

To derive a perturbation-level representation of our expression-based prediction targets, we performed pseudo-bulking by averaging cells for a given perturbation. The average control value from each experimental batch was then subtracted from the resulting averages, to derive control-centered perturbation effects.

B.5. Defining target cell states

For rank based evaluation metrics (AUPRC), we defined target cell states as the top 25% of our gene set score and fitness distributions. We then evaluated the model predictions as a binary classification task. This reflects the most realistic laboratory scenario where researchers are interested in detecting "hit" payloads to interrogate in subsequent experiments.

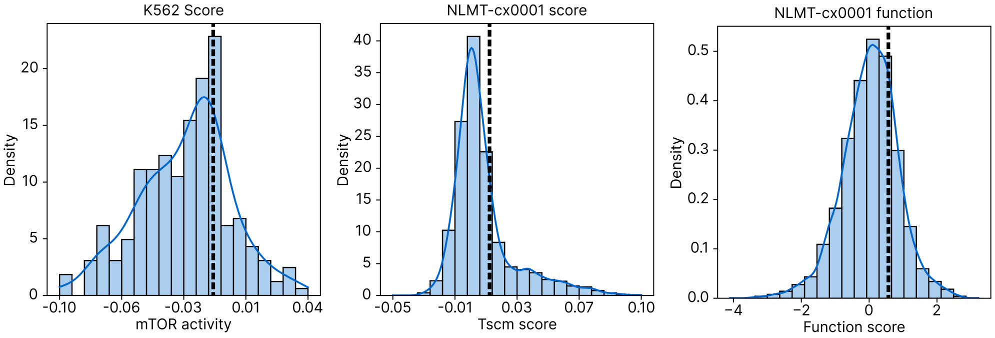

Figure 6. Binary target cell states y^* were defined by thresholding the top 25% of scores for each of the gene set and fitness prediction tasks. Black dotted lines indicate the threshold used for each score.

B.6. UMAP projection of cell state predictions

We performed qualitative inspection of payload predictions using UMAP projections. To construct embeddings, we used the mean cell state embeddings in our PCA representation and generated a UMAP projection from the ground truth data. We then fit a distance weighted nearest neighbors regression model to the ground truth data. Each prediction point was embedded using the distance weighted nearest neighbors regression model to allow qualitative comparison of the predicted and true data points.

B.7. Evaluation metrics

We used the following metrics to evaluate model performance.

CSE: We introduce the Control-Scaled mean squared Error, defined as the MSE of the model divided by the effect size of the perturbation (MSE of systematically predicting the control value). This metric is motivated by the observation that many of perturbations have small effects, and that a simple model predicting the control value could in practice achieve relatively low MSE in a large fraction of perturbations (Ahlmann-Eltze & Huber, 2024).

$$\mathrm{CSE}(y, \hat{y}, y_{\mathrm{ctrl}}) = \frac{1}{n} \sum_{i}^{n} \frac{(y_i - \hat{y}_i)^2}{(y_i - y_{\mathrm{ctrl}})^2}$$

PCC: Pearson correlation coefficient, defined as:

$$\mathrm{PCC}(y, \hat{y}) = \frac{\sum_i^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i^n (y_i - \bar{y})^2} \sqrt{\sum_i^n (\hat{y}_i - \bar{\hat{y}})^2}}$$

AUPRC: Area under the precision-recall curve, where p_i, r_i are the precision and recall at a given threshold i respectively, computed as:

$$\mathrm{AUPRC} = \sum_{i=1}^{n} (r_i - r_{i-1}) \cdot p_i$$

Cosine: The cosine similarity between true and predicted cell state vectors:

$$\text{Cosine}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\mathbf{y} \cdot \hat{\mathbf{y}}}{\|\mathbf{y}\|\|\hat{\mathbf{y}}\|}$$

Fraction of hits discovered: For a number of hits discovered at the active learning round i, N_i and a total number of hits to discover N_{tot}, the fraction of hits discovered F_i is defined as $\frac{N_i}{N_{\text{tot}}}$

B.8. Learning representation of batch effects in NLMT-cx0001

The NLMT-cx0001dataset consists of multiple tranches collected across several screening rounds, leading to the presence of batch effects. For the benchmarking experiment summarized in Table 1, 2, 3, we condition all models (including baselines) on experimental batches.

Mean model: We learn the conditional mean of each experimental batch b, $f_{\text{Mean}}(S, b) = \frac{1}{|\mathcal{D}_b|} \sum_i^{|\mathcal{D}_b|} y_i$, where $\mathcal{D}_b$ denotes the set of samples in batch b.

Additive model, Ambrosia-Linear: Experimental batches are encoded as one-hot vectors. In the case of **Ambrosia-Linear**, those one-hot vectors are concatenated to the payload representation.

Ambrosia: Experimental batches are represented as learnable embeddings of size $d = 8$, which are concatenated to the payload representation.

In the active learning experiments, we did not condition the Ambrosia models on experimental batch, mimicking the realistic scenario of unknown (future) experimental batch effects.

C. Experiments

C.1. Qualitative inspection of payload prediction performance

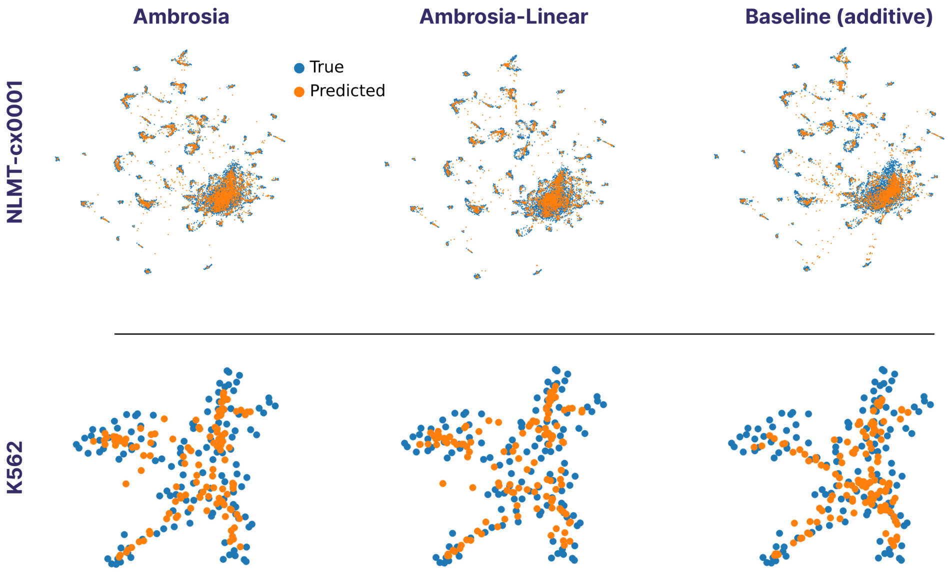

Figure 7. Qualitative inspection of payload prediction performance across models and combinatorial screening datasets on the *cell state* task. Cell state effects were predicted in the PCA representation for each payload, then projected for each dataset using UMAP (B) Ambrosia models show a higher fidelity of predicted effects to the ground truth data.

C.2. Evaluating generality across protein language model embeddings

We asked whether the sequence representations derived from larger pLMs provide benefits in the context of the perturbation prediction tasks presented in this work. We compared the performance of our Ambrosia model trained on embeddings from ProtT5 (Elnaggar et al., 2020) (3B parameters, $d = 1024$) with models trained on embeddings from the more recent ESM2 model (Lin et al., 2023) (15B parameters, $d = 5120$). We observed that performance was stable across pLM representations, with a slight gain in performance when using ESM2 representations in the cell state prediction task (Fig.8).

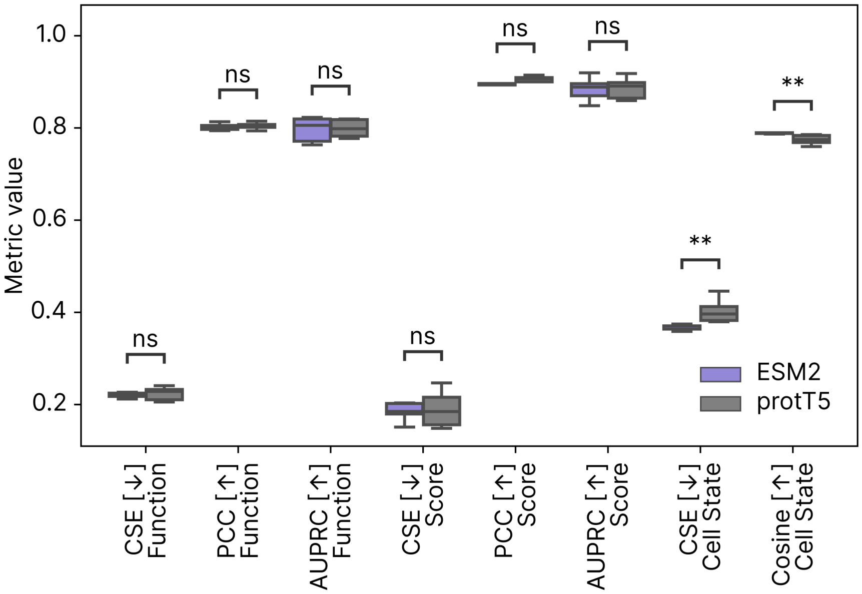

Figure 8. Ambrosia models trained using protein embeddings from multiple protein foundation models are capable of achieving strong performance. Transfer learning from protein foundation models appears to be a general principle, rather than a special case of emergent properties in the ESM2 embeddings we ultimately employed.

C.3. Generation of reprogramming payloads by Markov Chain Monte Carlo

During generative design experiments, we performed burn-in for 1000 iterations prior to collecting samples from our Markov chains. We subsequently logged the conditional likelihood $p_\theta(y > \tau|S)$ of all proposed payloads in the chain. Our chain demonstrated stationary behavior with a useful diversity of payload qualities.

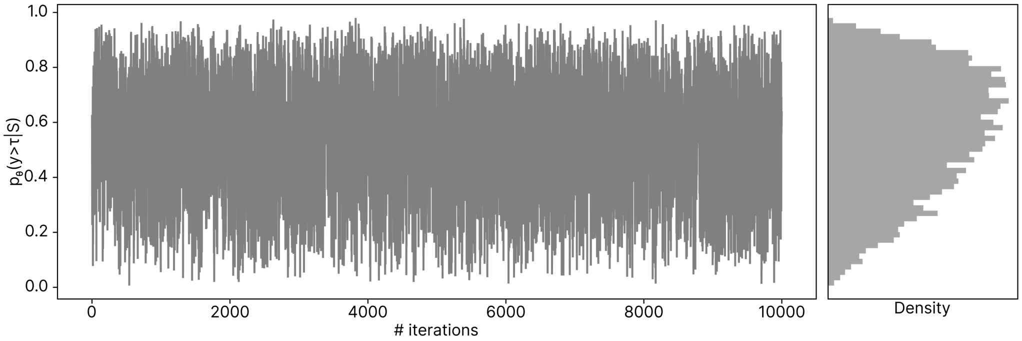

Figure 9. **Generative design with Markov Chain Monte Carlo sampling:** The likelihood of proposed payloads demonstrated useful variation across the sampling chain.

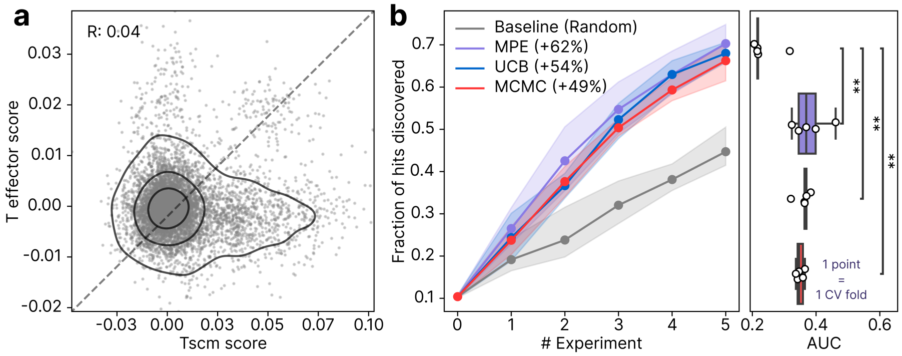

Figure 10. **Active learning campaigns on the T effector score**: **(Left)** The T effector score shows no correlation with the Tscm score ($r = 0.04$), serving an orthogonal prediction target for our active learning campaigns. Each point represents a single pseudobulk sample in our dataset. **(Right)** Ambrosia methods have significantly higher area under the curve (AUC) than the baseline when performing active learning campaigns on the T effector score (*: $p < 0.05$, **: $p < 0.01$; Mann Whitney U-test).

C.4. Active learning campaigns on an orthogonal T effector score

We repeated our active learning experiment on a different *gene set* score (T effector score) computed on NLMT-cx0001. We ensured this new gene set has little overlap with the Tscm gene set (Jaccard index: 0.02), and the score has low correlation ($r = 0.04$) with the Tscm score, providing orthogonal validation of Ambrosia's performance in the active learning setting. We found similarly that the Ambrosia methods were sufficient to accelerate hit discovery.