Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

In silico perturbation as an adversarial attack: combinatorial searchover cell states in single-cell foundation models

Yukio Oishi
Kyushu Institute of Technology
Momoko Hamano
Kyushu Institute of Technology
Kazuhiro Takemoto
takemoto.kazuhiro035@m.kyutech.ac.jp

Kyushu Institute of Technology

Research Article

Keywords: in silico perturbation, single-cell foundation model, Geneformer, adversarial attack, Greedy Coordinate Gradient, cellular reprogramming

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11449639/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

In silico perturbation as an adversarial attack: combinatorial search over cell states in single-cell foundation models

Yukio Oishi[1], Momoko Hamano[1], Kazuhiro Takemoto[1*]

[1*]Department of Bioscience and Bioinformatics, Kyushu Institute of Technology, Iizuka, Fukuoka, Japan.

*Corresponding author(s). E-mail(s): takemoto.kazuhiro035@m.kyutech.ac.jp;

Abstract

Background: Identifying which genes to perturb in order to drive a cell towards a desired state is a common problem in biology, and the answer is often a combination of genes. Single-cell foundation models can predict the effect of forcing a gene on without an experiment, but *in silico* perturbation has been confined to single genes or to a few hundred enumerated combinations. In token-based models, choosing which genes to force on is formally the same optimization problem as a discrete adversarial attack on a language model. We therefore investigated whether Greedy Coordinate Gradient (GCG), a search developed for such attacks, can be used to find combinations of genes.

Results: Our results suggest that GCG is useful for finding combinations of genes. In Geneformer, the effect of a set was well below the sum of the individual effects of its members, which suggests that assembling combinations from single-gene effects is not a promising approach. GCG exceeded forward-selection greedy search at equal cost in two models (Geneformer and scGPT) and at every set size. Against published reprogramming protocols, GCG recovered the reported factors at a rate well above chance and higher than the random-pool and all-gene greedy searches at equal cost, although ranking genes by their individual effect gave a higher recovery rate. Recovery corresponded to whether a search used the embedding gradient, which suggests that the gradient is useful for narrowing the search, whereas stronger optimization of the objective did not raise recovery, which suggests that the similarity induced by the model does not fully reflect the regulatory logic of genes.

Conclusions: These results are obtained within the constraints of the foundation models; nevertheless, searching for combinations of factors with foundation

models extends *in silico* perturbation beyond single genes, reveals a property of the induced objective that single-gene evaluation does not, and points to a new direction for using and evaluating the models.

Keywords: in silico perturbation, single-cell foundation model, Geneformer, adversarial attack, Greedy Coordinate Gradient, cellular reprogramming

1 Background

Identifying which genes to perturb in order to drive a cell towards a desired state is a problem common to many areas of biology. In the induction of pluripotent stem cells from somatic cells [1], and in direct cell reprogramming, which converts one somatic cell type into another without passing through a pluripotent state [2, 3], the choice of the factors to introduce is critical. The search for disease targets [4] and the prioritization of candidates for perturbation screens [5] pose the same problem. In many cases, the answer is not a single gene but a combination of genes [1, 5].

Single-cell foundation models provide a new tool for addressing this problem. Geneformer [4], scGPT [6] and many others [7, 8] represent a cell as a sequence of gene tokens, whereas some models, such as scFoundation, take expression values as continuous input [9]. With token-based models, the input can be edited as if a gene were forced on, the edited input re-embedded, and the magnitude and direction of the resulting shift in cell state measured (*in silico* perturbation). Forcing a gene on is thus a discrete edit of the token sequence. Because no experiment is required, the cost is a single forward pass. Although the accuracy of perturbation-response prediction has been criticized [10–13], *in silico* perturbation is a promising means of narrowing down candidates prior to experiments [4, 6].

Searching over combinations, however, remains difficult. The number of ways of choosing k genes from a set of candidates grows rapidly with the number of candidates and with k, and exhaustive experimental testing of plausible sets of factors has been described as inefficient and unscalable [14]. Experimental combinatorial screening has recently been scaled up with barcoded libraries of transcription factors, as in REPROcode [15], but the sets of factors to be tested still have to be nominated in advance, and a computational narrowing of candidates can complement such screens. Existing computational methods either rank transcription factors individually, as CellNet [16, 17], the entropy-based method of D'Alessio et al. [18], Mogrify [14] and ANANSE [19] do, or predict the consequences of a given perturbation, as CellOracle [20] does; combinations are therefore assembled from individual evaluations or specified in advance. Mogrify goes further by adding factors in order of rank, omitting those that do not increase the influence of the set, until the set covers nearly all genes expressed in the target cell; however, its criterion is the coverage of influence on an explicit network rather than the effect of the combination itself [14]. In the published foundation-model workflows, the *in silico* perturbation module of Geneformer [4] handles a single gene or a small, user-specified set, and enumeration is the main option when the effect of a combination is to be evaluated by the model itself.

Cui et al. [6] took this approach, enumerating 210 one- and two-gene combinations drawn from 20 genes, but the range that can be covered by enumeration is small. Foundation models offer an advantage in this regard: because they are differentiable, gradients are available, which allows combinations to be searched more efficiently than by enumeration.

A token-based foundation model is a sequence model, and *in silico* perturbation is the problem of selecting an edit to its input that moves its output towards a target, here a gene-expression state. This problem has the same form as a discrete adversarial attack on a language model [21, 22], in which k tokens are inserted into a prompt (the adversarial suffix) to steer the output of the model. The two problems are difficult for the same reason: the search space is combinatorial and the input is discrete, so that gradient descent cannot be applied directly. The standard solution, Greedy Coordinate Gradient (GCG) [23], computes the gradient of the objective with respect to the one-hot encoding of the discrete input. This gradient does not indicate which token to use, but it ranks the candidates, and the ranking is sufficient to narrow n candidates down to a few hundred before any of them is evaluated. The search then substitutes one token at a time from this shortlist, retaining the best candidate in each batch. The correspondence between the two problems is exact: the k perturbed genes play the role of the k tokens chosen by the attacker, the similarity to the target cell state is the attack objective, and the gradient with respect to the input embedding provides the proposal distribution over substitutions. The genes are not located literally at the end of the sequence, however; they are placed at the front of the rank-ordered sequence in Geneformer and in place in scGPT. The problem does not become easy, as it remains combinatorial and the gradient is a linear approximation at a discrete point, but a first-order signal becomes available.

Here, we investigate whether the idea underlying GCG can be used to search efficiently for combinations of factors. We implement GCG (hereafter the *attack*) on Geneformer, with scGPT as a second model, and compare it with greedy forward selection and other search methods at equal computational cost. As a test bed, we use direct cell reprogramming, mentioned above, because it allows many different conversions to be examined at once and because factor combinations that have been reported experimentally to achieve reprogramming provide an external reference. Our approach is not specific to reprogramming and could in principle be applied to other factor-search problems. Across 18 conversions, we first found that the induced objective is strongly non-additive (measured in Geneformer), so that summing individual effects is not a viable shortcut. The attack then found combinations with higher similarity to the target, as measured by the model, than greedy search at the same cost, and the margin increased with set size. Decomposing the attack into its design choices showed that the two specific to GCG, namely proposing candidates from the gradient and adopting the best candidate in each batch unconditionally, were individually ineffective or harmful and beneficial only in combination. The returned sets also allowed us to ask what the objective itself captures, by comparing them with experimentally reported reprogramming combinations. This comparison becomes possible only once combinations can be searched, and it provides a means of evaluating and selecting models.

2 Methods

2.1 Single-cell foundation models and *in silico* perturbation

We used two token-based single-cell foundation models, Geneformer and scGPT. Geneformer is the primary model, because its pretrained model supports *in silico* perturbation without fine-tuning [4]. The perturbation functionality of scGPT, by contrast, is a fine-tuning target, and its published perturbation results, including the retrieval of Cui et al. [6], use a model fine-tuned on measurements from the system in question. From scGPT we therefore used only the pretrained embedding and the token-edit operator, which suffice to define the objective and to run the search, and we used it as a second model to test whether the comparison of the search methods and the evaluation of single-gene rankings depend on the model. Because the pretrained scGPT does not support zero-shot *in silico* perturbation, the measurement of non-additivity (Section 3.1) and the set-level recovery of reported factors (Section 3.5) were carried out in Geneformer only. Geneformer represents a cell as a sequence of gene tokens, ordering the expressed genes by expression rank, so that expression magnitude is discarded and position carries expression level [4]. scGPT pairs each gene token with a token for its expression value, binned within the cell [6]. In both models, a transformer computes an embedding of the cell. Neither model was fine-tuned: we used the publicly released pretrained models with their default settings. For Geneformer (V1), tokenization followed the reference implementation, the input length was 2048, and the final layer was used. For scGPT, we used the whole-human checkpoint (scGPT_human) with a maximum sequence length of 1200 and 51 bins; expression was normalized to 10^4 counts per cell, log1p-transformed and binned per cell following the official preprocessing, and the cell embedding was the cell embedding output by the model (512 dimensions).

Overexpression of a gene was implemented by an operator specific to each model. In Geneformer, following the reference implementation, the gene is removed from its current position, if present, and inserted at the front of the sequence. Position carries meaning, so that two sets containing the same k genes in different orders are different inputs. In scGPT, the expression value of the gene is set to the highest bin (index 50 of 51), and the gene is appended if it is absent from the cell. The positions of the other tokens are unchanged, so that the order of the k genes does not affect the input. When a sequence had reached the maximum length, k positions were reserved by truncating the base sequence to the maximum length minus k (Additional file 1).

2.2 Objective

The target cell type is represented by the L_2-normalized centroid c of the embeddings of up to 200 target cells. For a source cell x and a set S of k genes, the objective is the cosine similarity between the embedding of the perturbed cell and c,

$$f(S) \;=\; \cos\big(E(\mathrm{perturb}(x, S)),\, c\big), \tag{1}$$

where E is the cell embedding of the model and perturb is the overexpression operator described above. We report $\Delta(S) = f(S) - f(\emptyset)$, the improvement over the unperturbed cell, in units of 10^{-3}. This objective builds on the *in silico* perturbation analysis of Geneformer, which quantifies the effect of a perturbation by the cosine similarity between the cell embeddings before and after the perturbation [4]; here the similarity is measured to the centroid of the target cell type, so that the effect is expressed as movement towards the target. Whereas that analysis evaluates single genes or sets specified in advance, we treat the selection of a set of k genes from the candidate transcription factors (Section 2.5) as an optimization problem; k is given externally.

2.3 Greedy Coordinate Gradient and its extension to foundation models

GCG is a method for discrete adversarial attacks on language models [23] (Figure 1a). A suffix $s = (s_1, \ldots, s_k)$ of k tokens is appended to a prompt and chosen to minimize a loss $L(s)$, the negative log-likelihood of a target response. Because tokens are discrete, L cannot be differentiated with respect to s directly. GCG therefore takes, for each position i, the gradient of L with respect to the one-hot encoding e_{s_i} of the token at that position. This gradient is a first-order estimate of the change in L when the token at position i is replaced by each token v of the vocabulary, and the tokens with the most negative components form the candidates for position i. A number of single-token substitutions are then sampled from these candidates, the loss of each is evaluated exactly with a forward pass, and the best substitution is adopted. The procedure is repeated. The gradient does not determine which token is best, but it narrows the vocabulary to a short list before any candidate is evaluated.

We extended this procedure to token-based foundation models, with the correspondence of Table 1 (Figure 1b): the k perturbed genes play the role of the suffix, and the objective f of Section 2.2, which is maximized, that of $-L$. The gradient of f with respect to the one-hot encoding of the gene at position i is $g_i = W\,\partial f/\partial h_i$, where h_i is the input embedding at that position and W is the gene-embedding table; its v-th component estimates the change in f when that gene is replaced by gene v. Candidates were restricted to the n transcription factors (Section 2.5). At each step, the gradient was computed once, the 256 genes with the largest components of g_i were taken for each of the k positions, giving a pool of $k \times 256$ substitutions, and 128 substitutions were drawn from this pool, with the position chosen uniformly at random for each. Each was evaluated exactly with a forward pass, and the best was adopted whether or not it improved the objective (unconditional acceptance). The initial set was random, and three random restarts were run per (cell, method); we report the restart mean and the best of three. The only model-specific components are the evaluation of f and of the gradient. In Geneformer, S is an ordered tuple, which corresponds exactly to an ordered suffix; in scGPT, it is an unordered set (Additional file 1: Figure S1). We refer to this method as the *attack*.

The 256 candidates per position follow the default of GCG [23]. The number of candidates per step (128, smaller than the default of 512) and the number of restarts (three) were set with computational cost in mind (Section 2.6). Cost was measured in forward passes, with one gradient evaluation charged as two forward passes, so that a

Table 1 The correspondence, term by term.

adversarial attack on a language model	in silico perturbation of a single-cell model
prompt tokens	gene tokens of the cell
adversarial suffix of length k	k overexpressed genes
attack objective on target logits	cosine similarity to target cell centroid
gradient w.r.t. token one-hot	gradient w.r.t. gene one-hot
top rows of the embedding table	top rows of the gene embedding table
batch of substitutions, best adopted	batch of substitutions, best adopted

step costs at most 130 forward passes, whatever k is, and slightly fewer when duplicates leave fewer than 128 candidates. The search ran for $\lfloor \text{budget}/128 \rfloor$ steps. The budgets were set from the greedy side: greedy over a pool of p genes costs $\sum_{i=0}^{k-1}(p-i)$ forward passes, which is 1289 at $k = 2$ and 2574 at $k = 4$ for a pool of 645 genes, and 2548 at $k = 8$ for a pool of 322 genes. The attack was given 1290 (10 steps), 2580 (20 steps) and 2580 (20 steps), within 1.3% of these, and, in the comparison with greedy at its natural budget, $k \times n$ (164 steps at $k = 8$). At $k = 4$, the budget falls just under the 2632 passes that the ranking by single-gene effect requires, which is why Section 3.1 compares the two at that set size. The realized mean costs of the attack were within 1% of the specified budgets (Additional file 1).

2.4 Comparison methods

We compared the attack with the methods below. Those closest to previous approaches are the rankings and greedy forward selection; the others are controls for the components of the attack.

Ranking by single-gene effect orders the candidates by the effect $\Delta(j)$ of overexpressing each alone and takes the top k as the set. This corresponds to the standard use of *in silico* perturbation [4], in which genes are evaluated one at a time and ranked, and to computational methods that rank transcription factors individually [14, 16, 17, 19]. It is the simplest way to obtain a combination, and we used it as the reference for comparison. Ranking by differential expression between target and source cells approximates the procedure by which reprogramming factors have historically been nominated [1], namely genes expressed in the target and not in the source, and ranking by mean expression in the target cell type is its control for the contribution of the source. These rankings, which we call the ranking baselines, were used in the comparison with curated factors (Sections 3.4 and 3.5).

Greedy forward selection (greedy over all genes) starts from the empty set, evaluates every candidate together with the genes already chosen, commits to the best, and repeats this k times. It builds a set by evaluating combinations with the model itself, as an experimenter would, and extends the enumeration-based approach of Cui et al. [6]. Its cost is fixed at $\sum_{i=0}^{k-1}(n-i) = kn - k(k-1)/2$, which we write as $k \times n$ where the correction does not matter; at $k = 8$ it is 21028 for Geneformer and 21228 for scGPT. To compare at the cost of the attack, we also used greedy selection restricted to budget/k genes, drawn uniformly at random (random pool) or from the top of the

Table 2 The 2×2 decomposition of the attack.

	improve-only acceptance	unconditional acceptance
uniform candidates	(a) plain hill climbing	(c) unconditional, uniform
gradient candidates	(b) gradient hill climbing	(d) GCG (the attack)

Cell (d) is GCG as specified by Zou et al. [23]; cells (a) to (c) are the ablations.

input-embedding gradient (gradient pool). At $k = 2$, the space is small enough to be enumerated exhaustively (Additional file 1).

The remaining methods are not previous approaches but controls. Best-of-random samples complete sets independently and keeps the best; it performs no local search and is the floor. Hill climbing starts from a complete set and substitutes one member at a time, adopting a substitution only if it improves the objective. It differs from the attack in the two respects in which the attack follows GCG, namely the source of candidates (the gradient or uniform sampling of the vocabulary) and the acceptance rule (unconditional, or improve-only); crossing the two gives four searches (Table 2), of which cell (d) is the attack.

2.5 Data

Cells were taken from Tabula Sapiens v1 and v2 combined [24, 25], accessed through the CELLxGENE census [26] release of 2025-11-08 (1136218 cells across 28 tissues). The candidate vocabulary was the list of human transcription factors and cofactors compiled in AnimalTFDB4 [27] (2661 genes). Of these, 2632 are in the token dictionary of Geneformer and 2657 in that of scGPT, and n denotes the model-specific count throughout.

Each pathway is a conversion from a source to a target cell type, taken from published protocols, the references of which are listed in Additional file 1; a protocol is a factor set reported to achieve a conversion. The factor sets and the pathway definitions are provided with the code (Availability of data and materials). Source cell types follow the source publications of the protocols. We retained the 18 pathways for which both cell types are represented in Tabula Sapiens with sufficient cells and the reported factors are identifiable in the vocabulary (Additional file 1: Table S1); the selection process and the excluded pathways are described in Additional file 1. The curated factor list of a pathway is the union of the factor sets of its protocols. These lists serve only as an external reference against which to read the results; they are not used in any search, in the accounting of cost or in any comparison between search methods.

The non-additivity analysis, the matched-cost comparison and the evaluation against curated factors used all 18 pathways. The 2×2 decomposition, the anytime analysis and the comparison with greedy at its natural budget, which require a run at the full $k \times n$ budget for every condition, were restricted to five pathways (P01, P03, P07, P12 and P14) to limit computational cost; Additional file 1 shows that these five are representative of the full set.

2.6 Evaluation and statistics

Searches were compared at equal cost on 20 source cells per pathway, with the same cells and the same initial sets for all methods. Comparisons were paired at the level of the source cell ($18 \times 20 = 360$ pairs, and $5 \times 20 = 100$ pairs for the comparison with greedy at its natural budget, the 2×2 decomposition and the anytime analysis, all of which use the five-pathway subset) and tested with paired t-tests and Wilcoxon signed-rank tests, reported side by side (two-sided; Additional file 1: Table S2); aggregation to the pathway level used sign tests. The random seed was 42. The numbers of source cells, restarts (Section 2.3) and target cells (at most 200 per centroid) were chosen to balance computational cost, given that every method was run at the $k \times n$ budget on 18 pathways, against the precision of the estimates.

The ranking methods of Section 2.4 treat the effect of a set as the sum of the effects of its members. To test this assumption in Geneformer, we measured the individual effect $\Delta(j)$ of each candidate transcription factor in 50 source cells per pathway and, for each set size k, regressed the measured $\Delta(S)$ of 200 random sets per cell (10000 sets per pathway and set size) on the additive prediction $\sum_{j \in S} \Delta(j)$, by ordinary least squares with an intercept, pooled over the cells and random sets of each pathway and set size.

Finally, the results of the searches were compared with the curated factors. A ranking of the whole vocabulary was evaluated against the curated factors of a pathway in two ways, both based on the area under the receiver operating characteristic curve (AUC), for which 0.5 is chance. The conventional AUC takes the factors of the pathway as the positive class and all other genes as the negative class; the pathway-specific AUC takes as the negative class the curated factors of the other pathways, excluding any that also appear in the list of the pathway itself (19 of the 62 distinct curated genes appear in more than one pathway). A set returned by a search was scored by its largest overlap with any single protocol C_j of the pathway, $h = \max_j |S \cap C_j|$, and the chance rate was obtained by scoring 20000 random k-subsets of the vocabulary per pathway by the same procedure.

3 Results

3.1 Non-additivity of the induced objective

This analysis was restricted to Geneformer (Section 2.1). The measured $\Delta(S)$ of random sets of size k was regressed on the additive prediction $\sum_{j \in S} \Delta(j)$ (Section 2.6); an additive objective gives a slope of 1.00.

The mean slope was 0.491, 0.151 and 0.083 at $k = 2$, 4 and 8, and no pathway had a slope above 0.9 at any set size (Table 3; Figure 2a). The mean Pearson correlation fell from 0.69 at $k = 2$ to 0.34 at $k = 8$ (Figure 2b). At $k = 4$, the four factors with the largest individual effects gave an improvement of +4.215 (in units of 10^{-3}) at a cost of 2632 forward passes, and a set found by the search gave +6.417 at a budget of 2580 forward passes, which was higher in all 18 pathways.

Table 3 Regression of measured set effect on additive prediction.

set size k	mean slope	pathways with slope > 0.9
2	+0.491	0 of 18
4	+0.151	0 of 18
8	+0.083	0 of 18

Geneformer, 18 pathways $\times$ 50 cells. An additive objective would give a slope of 1.00.

Table 4 Matched-budget comparison of the attack with greedy search.

model	k	attack	greedy (gradient pool)	ratio	p
Geneformer	2	3.629	3.480	1.043	1.7×10^{-3}
Geneformer	4	6.417	5.434	1.181	5.2×10^{-25}
Geneformer	8	10.483	7.493	1.399	7.3×10^{-59}
scGPT	2	1.144	0.974	1.175	3.1×10^{-16}
scGPT	4	1.941	1.631	1.190	4.6×10^{-26}
scGPT	8	3.058	2.540	1.204	1.7×10^{-39}

Paired over $n = 360$ source cells (18 pathways $\times$ 20 cells). Ratio is the improvement of the attack divided by that of greedy over the gradient pool. p is from a paired t-test; a signed-rank test on the same pairs gives smaller values in five of the six conditions and a comparable value in the sixth, so the t-test is the conservative choice. Ratios and the margins quoted in the text are computed before rounding, so they need not equal the difference of the rounded columns. Improvements are in units of 10^{-3}.

3.2 Comparison with greedy forward selection at equal cost

At matched cost, greedy forward selection was restricted to a pool of budget/k genes (Section 2.4).

The attack gave higher improvements than the gradient-pool greedy in all six conditions (two models and three set sizes), under both a paired t-test and a signed-rank test (Table 4), and than the random-pool greedy in all six, with improvements 1.22 to 2.17 times those of the pool. At the level of pathways, the attack had the higher mean than the gradient-pool greedy in all 18 pathways in five of the six conditions (sign test $p = 7.6 \times 10^{-6}$) and in 15 of 18 in Geneformer at $k = 2$ ($p = 0.0075$), and than the random-pool greedy in all 18 pathways in every condition.

The difference from the gradient-pool greedy increased with set size in both models (Figure 3b): from +0.149 to +0.983 to +2.991 in Geneformer and from +0.170 to +0.310 to +0.518 in scGPT. The proportion of cells on which the attack was ahead rose from 54% to 83% to 95% in Geneformer and from 61% to 76% to 90% in scGPT.

With greedy at its natural budget of $k \times n$ and $k = 8$, the attack reached 12.855 against 9.397 for greedy in Geneformer, a ratio of 1.37 ($p = 1.3 \times 10^{-15}$), and 3.112 against 3.116 in scGPT ($p = 0.59$; 90% confidence interval of the difference,

[−0.50%, +0.25%] of the greedy value). Greedy did not exceed the attack at any budget (Figure 3c). At $k = 2$, where the space can be enumerated exhaustively, the attack reached 84.3% of the per-cell optimum at 1/5368 of the exhaustive cost (Additional file 1: Figure S2).

3.3 Components of the attack

Replacing the gradient-guided candidates of the attack with uniform sampling from the full vocabulary, with everything else unchanged, still outperformed greedy at its natural budget in Geneformer at $k = 8$ (paired difference +2.995, $p = 1.5 \times 10^{-13}$; ratio 1.319, against 1.368 for the full attack). All four variants in the following analysis outperformed greedy, with ratios between 1.307 and 1.368 (every p below 4×10^{-12}).

We crossed the source of the candidates (gradient or uniform) with the acceptance rule (improve-only or unconditional) at $k = 8$ and the $k \times n$ budget, in five pathways (Section 2.5; Figure 4, Additional file 1: Table S2). The interaction is the difference of differences: how much more the gradient helps under unconditional acceptance than under improve-only acceptance, or, equivalently, how much more unconditional acceptance helps with gradient candidates than with uniform ones. Under improve-only acceptance, gradient candidates changed nothing in Geneformer and gave no improvement in scGPT. Unconditional acceptance with uniform candidates lowered the objective in 88 of 100 cells in scGPT and 65 of 100 in Geneformer. The interaction was +0.588 in Geneformer and +0.064 in scGPT, significant in both models under both tests, positive in 5 of 5 pathways in each, and robust to best-of-restarts aggregation ($p = 0.0019$ and $p = 2.0 \times 10^{-4}$). When the two tests disagreed, it was on a single factor and never on the interaction (Additional file 1). Relative to plain hill climbing, the combination gave +0.463 in Geneformer (3.7% of the objective) and +0.008 in scGPT (0.25%).

The two acceptance rules gave identical trajectories through 512 forward passes in both models and for both candidate sources (Figure 5). With uniform candidates they first differed at 1024, and from 2048 on unconditional acceptance gave the lower value at every budget in both models. With gradient-guided candidates they first differed at 2048. In Geneformer the difference was positive at 2048, negative at 4096 and positive from 8192 on (+0.37 at 16384), so that it was not monotone in budget. After the divergence, the sign of the difference was thus largely determined by the candidate distribution rather than by the budget. Under improve-only acceptance, the ratio of gradient to uniform candidates in Geneformer peaked at 1.292 around a budget of 1024 and returned to 1.016 by 16384; in scGPT it fell below 1 at large budgets.

3.4 Single-gene rankings against reported reprogramming factors

The rankings were evaluated against the curated factors by the conventional and the pathway-specific AUC (Section 2.6). Across the 18 pathways, Geneformer and differential expression scored 0.875 and 0.875 by the conventional AUC and 0.701 and 0.899 by the pathway-specific AUC. Ranking by mean expression in the target cell type scored 0.764 and 0.869. Differential expression had the higher pathway-specific

Table 5 Recovery of published reprogramming factors by the returned sets.

	budget	sets	$h \geq 1$	vs. chance
Differential expression, top k	0	18	55.6%	38.4
Single-gene effect, top k	2632	18	44.4%	30.7
Attack	2580	1080	15.6%	10.8
Greedy, gradient pool	2580	360	8.3%	5.8
Greedy, random pool	2580	360	5.3%	3.6
Greedy, all genes	21028	360	3.6%	2.5
Target expression, top k	0	18	0.0%	0.0

At $k = 8$ on Geneformer over 18 pathways. "vs. chance" is the ratio to a random k-subset scored identically. Budget is the specified number of forward passes per cell (Section 2.3); the three ranking baselines return one set per pathway.

AUC in 14 of the 18 pathways and the two tied in one; the per-pathway difference ranged from -0.194 to $+0.705$ (Additional file 1).

In scGPT, the AUCs were 0.718 by the conventional and 0.583 by the pathway-specific measure. The conventional AUC exceeded the pathway-specific AUC in 17 of 18 pathways in each model, whereas the pathway-specific AUC exceeded the conventional AUC in 18 of 18 pathways for target expression and in 12 of 18 for differential expression.

3.5 Recovery of reported factors by the returned sets

The recovery of reported factors by the returned sets was evaluated in Geneformer only (Section 2.1). At $k = 8$, a set returned by a search was scored by its largest overlap h with any single protocol of its pathway (Section 2.6); a random k-subset of the vocabulary reached $h \geq 1$ in 1.45% of draws and $h \geq 2$ in 0.011%. The proportion of sets with $h \geq 1$ was 55.6% for the ranking by differential expression (38.4 times chance), 44.4% for the ranking by single-gene effect (30.7), 15.6% for the attack (10.8), 8.3%, 5.3% and 3.6% for the gradient-pool, random-pool and all-gene greedy (5.8, 3.6 and 2.5), and 0.0% for the ranking by target expression (Table 5).

At the level of pathways, the attack had the higher recovery than the random-pool greedy in 10 of the 11 pathways in which the two differed (sign test $p = 0.012$) and than the all-gene greedy in 10 of 12 ($p = 0.039$); it was not distinguishable from the gradient-pool greedy (7 of 11, $p = 0.55$), from the ranking by single-gene effect (4 of 12, $p = 0.39$) or from the ranking by differential expression (5 of 15, $p = 0.30$), and it exceeded the ranking by target expression (11 of 11, $p = 0.001$), which recovered nothing. Sets with $h \geq 2$ made up 1.0% of the attack's sets. One of them, in the fibroblast to hepatocyte pathway, contained FOXA1 and HNF4A and reproduced a

published two-factor protocol in full; it is the only complete recovery of a multi-factor cocktail by any method (the two pathways with a single-factor protocol being excluded).

Averaging the two hit rates in each group, the two searches that use the embedding gradient recovered in 11.9% of sets and the two that do not in 4.4%. The all-gene greedy had the second-highest mean improvement of the four searches and the lowest recovery. Within a single method, the rank correlation across pathways between the improvement achieved by a set and its overlap had a median of -0.106 and was positive in 5 of 11 pathways. This did not meet the criterion fixed before the comparison for claiming a relationship, namely a median exceeding 0.10 in absolute value with signs consistent across pathways at $p < 0.05$ (here $p = 1.0$).

4 Discussion

Overall, our results suggest that GCG, a search developed for adversarial attacks on language models, is useful for finding combinations of factors with token-based single-cell foundation models. We discuss below the non-additivity that makes such a search necessary, the sources of its advantage over greedy selection and what the returned sets reveal about the objective, followed by the implications and the limits of the correspondence.

4.1 Non-additivity and the necessity of a search

Our results suggest that assembling a combination of factors from a ranking of single-gene effects, which is the standard use of *in silico* perturbation [4], is not a promising approach. In Geneformer, the slope of the measured effect of a set on the additive prediction was well below 1.00 at every set size, and at $k = 4$ the four genes with the largest individual effects gave a smaller improvement than a set found by the search at a comparable cost (Section 3.1). This sub-additivity is consistent with the structure of the input of Geneformer, a fixed-length sequence of rank-ordered tokens, in which inserted genes displace one another and compete for the same context. Two other explanations are unlikely: a rescaling of an otherwise ordered prediction, because the correlation fell in parallel with the slope, and regression dilution, because the individual effects were measured deterministically.

4.2 Why the attack outperformed greedy selection

The attack achieved a higher value of the objective than greedy selection at the same cost, through a revision of the set that greedy cannot perform (Section 3.2). Greedy adds one factor at a time and never reconsiders: a factor that looked best when the set was empty remains in the set even when the other $k - 1$ members make it redundant. The attack, in contrast, starts from a complete set and repeats substitution and acceptance, so that it can undo an early commitment. That the difference grew with set size is consistent with an accumulation of commitments that greedy cannot revise.

Consistent with this interpretation, replacing the gradient with uniform sampling still outperformed greedy at its natural budget (Section 3.3; ratio 1.319, against 1.368

for the full attack), so that most of the advantage derives from the revision of the set rather than from the gradient. The gradient is nevertheless useful for narrowing the search. Under improve-only acceptance, the ratio of gradient to uniform candidates in Geneformer peaked at 1.292 around a budget of 1024 and returned to 1.016 by 16384 (Figure 5b). While the budget is limited, narrowing the candidates for each position from 2632 to 256 before any is evaluated (Section 2.3) therefore makes the search more efficient, whereas at large budgets uniform candidates catch up.

Proposing candidates from the gradient and adopting the best candidate unconditionally, the two components specific to GCG, were individually ineffective or harmful and beneficial only in combination (Section 3.3). We interpret this as follows. Unconditional acceptance is a mechanism for leaving a local optimum, and leaving is worth its cost only if the candidates from which the replacement is drawn keep the trajectory within a useful region; under uniform candidates, an accepted non-improving move is simply a loss. At large budgets, the gradient therefore acts as a proposal distribution rather than a descent direction.

The anytime curves are in line with this interpretation (Figure 5a). While the search is still ascending, adopting the best candidate and adopting it only if it improves select the same move, so that the two rules can differ only after a local optimum is reached. Accordingly, the trajectories of the two rules were identical through 512 forward passes and first differed at 1024 with uniform candidates and at 2048 with gradient candidates, the later divergence under gradient candidates being likely due to the later stall of their ascent. This explains why unconditional acceptance mattered only at large budgets.

4.3 What the induced objective captures

The attack found sets with higher values of the objective than the ranking of single-gene effects, which is the standard use of *in silico* perturbation [4], and than greedy selection at the same cost (Sections 3.1 and 3.2; the former at $k = 4$). It also recovered reported reprogramming factors more often than the random-pool and all-gene greedy searches, although not distinguishably from the gradient-pool greedy (Section 3.5). The search made it possible to examine the induced objective itself at the level of sets, and the objective carried information on the reported factors. The single-gene ranking by Geneformer scored the same conventional AUC as differential expression (0.875; Section 3.4), the sets returned by the attack recovered reported factors 10.8 times more often than chance, and every search was above chance (Section 3.5). The objective was also complementary to the expression-based rankings: of the 13 pairs of a factor and a pathway that the ranking by target expression placed below the midpoint of the vocabulary, the single-gene ranking by Geneformer placed all higher and lifted seven into the top 10%, including NEUROG3 and PAX4, which specify endocrine lineages (Additional file 1).

On the other hand, the discrimination of the factors of one conversion from those of the others was weaker than that of differential expression (pathway-specific AUC of 0.701 against 0.899), and the conventional AUC exceeded the pathway-specific AUC in 17 of the 18 pathways, which suggests that the signal shared by reprogramming factors in general was captured mainly. At the level of sets, recovery was lower than by

the ranking by differential expression (55.6%) or by single-gene effect (44.4%), against 15.6% for the attack. The recovery by the searches appeared to follow the first-order information that the search consulted on the way, rather than the objective value at which it arrived (11.9% for the two searches that use the gradient, against 4.4% for the two that do not), although the rank correlation across sets within a method did not meet the criterion fixed before the comparison. The ordering is consistent with the candidate distribution carrying the biological signal, although we have not isolated the acceptance rule at the set level. Overall, the sets optimized for the objective recovered reported factors less often than the ranking by single-gene effect, which is not optimized, and stronger optimization did not raise recovery (the all-gene greedy had the second-highest mean improvement and the lowest recovery). This suggests that the similarity induced by the model may reflect directions in the embedding space that raise the similarity more readily than the regulatory logic of genes. It is consistent with the property of adversarial attacks that the harder an objective is optimized, the more an input exploits the vulnerabilities of the model. The higher recovery of the single-gene ranking may be because the insertion of one gene is close to the input distribution of pretraining, whereas an optimized insertion of eight genes departs from it, which we did not test. This property, which single-gene evaluation does not reveal, was brought to light by the set-level search, and clarifying its origin and improving the model and the objective accordingly are subjects for future work.

The comparison needs to be read with the properties of the benchmark in mind. The ranking by target expression scored 0.764 by the conventional AUC and 0.869 by the pathway-specific AUC, close to the 0.899 of differential expression. In the pathway-specific AUC the negative class consists of the factors of other lineages, which are expected to be expressed less in the target, so that the measure rewards expression specific to the target. Reprogramming combinations have also been assembled, as the pluripotency factors were [1], by nominating genes expressed in the target and not in the source and testing them. A ranking by differential expression is close to that procedure, and agreement with it is partly a measure of agreement with the procedure. The benchmark can credit a method only for recovering combinations that have already been proposed, and a search over a different objective is penalized for every factor that it proposes that is correct but untried. The dissociation between the conventional and pathway-specific AUCs appeared in both token-based models and in the opposite direction for the two expression rankings; that two models that share no architecture, vocabulary or pretraining corpus fall on one side and two expression rankings on the other suggests that it lies in the benchmark rather than in either model. The summary means also conceal considerable heterogeneity across pathways, and a benchmark of this granularity cannot resolve differences of that size.

Set-level search provides a way of evaluating foundation models along an axis other than predictive accuracy. Existing criticisms concern accuracy measured one gene or one cell at a time [10–12]; the present analysis concerns the shape of the objective over sets, and a model can be accurate in the first sense and still have the second problem, which only a set-level search can reveal. A better model should have the property that the set that maximizes the induced similarity is the set that an experimenter would want to introduce. This property can be measured cheaply (Section 3.5) and before

any wet-lab work, although our set-level analysis was limited to Geneformer. A model could be selected against it, though not fine-tuned on it: the agreement is partly with differential expression, and optimizing it directly would train that in.

4.4 What the correspondence gives to single-cell modelling

The correspondence makes combinatorial *in silico* perturbation feasible at a cost below that of a single greedy pass over all genes. In Geneformer at $k = 8$, the attack, with 2580 forward passes, outperformed greedy, which requires 21028 (Section 3.2, Figure 3c; in scGPT the two were equivalent, Section 4.6). No new algorithm was required: the search algorithm existed in the adversarial-attack literature, and the barrier appears to have been not computational hardness in the abstract but the absence of a bridge between the two vocabularies. A token-based single-cell foundation model is a transformer over a discrete vocabulary with a fixed context and a learned embedding table. Algorithms developed to manipulate such models, including adversarial attacks on language models, can therefore be expected to be applicable to it through the correspondence. Extensions of GCG have already been proposed [28, 29] (Section 4.5), and using them may enable a more efficient search. This direction shows the practical value of recognizing the correspondence and is left for future work.

The correspondence also changes what the method depends on. Retrieval over enumerated candidates [6] requires a model that predicts the post-perturbation profile of each candidate accurately enough for the closest match to be the right one, and that model has to be fine-tuned on perturbation measurements from the system in question. The search reported here asks less of the model: it uses the pretrained embedding as given, without fine-tuning, and requires only that the embedding order candidate sets sensibly, not that it reconstruct expression. Whether the embedding does so is the question addressed in Section 4.3. Whether this is a lower bar in practice is a subject for future work, but it is a different question, and the search may not inherit the failure mode identified by recent benchmarks of perturbation-response prediction [10–12]. Fine-tuning the model on measured perturbation responses is also compatible with the search, which requires only a differentiable score, and may be one way to improve the objective.

4.5 What the correspondence gives back to adversarial robustness

The 2×2 decomposition is also an ablation of the design of GCG itself (Section 3.3). Zou et al. [23] adopt the best candidate of each batch unconditionally. This choice has been examined through the reuse of intermediate candidates [28] and through a simulated-annealing rule [29], but, to our knowledge, the generation of candidates and the acceptance rule have not been manipulated independently. The annealing variant applies its temperature at both stages at once, and an interaction of the kind reported here cannot be seen when both change together. That Tan et al. [29] recommend careful tuning of the temperature parameter is consistent with what would be expected from a bundled change whose two halves are individually ineffective or harmful.

Budget is also a variable missing from the annealing study, which reported that increasing exploration lowers attack success at its own operating point (2×10^4 candidate evaluations) and named the exploration-exploitation trade-off as future work [29]. The anytime curves (Section 3.3, Figure 5) show that the acceptance rule is irrelevant for short optimizations and consequential for long ones. At fixed proposal quality, this predicts that annealing schedules for text attacks should also be tuned against budget.

4.6 Dependence of the advantage on the model

The advantage of the attack held at matched cost in both models (Section 3.2), but at the natural budget of greedy it held in Geneformer only, and in scGPT the two were indistinguishable. The interaction structure of the 2×2 decomposition replicated in both models (Section 3.3). The size of the performance advantage therefore depends on the model. One possible contributor is that the pretrained scGPT does not support zero-shot *in silico* perturbation (Section 2.1), so that its objective was not designed or validated to capture the response to perturbation. With two models, this cannot be separated from other factors, such as differences in the perturbation operators [30]. Because the models that satisfy the applicability condition are currently limited (Section 4.7), examination of a third model is left for future work.

4.7 Applicability conditions

The correspondence requires a model that exposes a discrete input space whose embedding table is the gradient's projection target. This is a concrete criterion, checkable before any experiment, for deciding whether a new model admits the method as stated. It does not depend on the biological problem: any task that asks which genes to perturb, whether for reprogramming, for disease targets or for the prioritization of screening candidates, can be treated as the same optimization problem once an objective over the output of the model is defined. Geneformer and scGPT meet the criterion. A model that takes expression as a continuous value does not: scFoundation embeds the value itself [9], so that a perturbation is a real-valued displacement rather than a token edit, and the substitution step of the attack has no counterpart there. The pretrained models that satisfy the criterion are currently limited, and models fine-tuned from Geneformer or scGPT share their architecture and vocabulary, so that they would not constitute independent tests.

This bounds the discrete form of the correspondence, not the correspondence itself. For models that take expression as a continuous value, such as scFoundation, projected gradient descent (PGD) [31], the standard attack against continuous inputs, appears more suitable than GCG. It applies directly to a real-valued expression vector under a norm or sparsity constraint, and the objective, the gradient and the target are the same three objects used here. This extension is left for future work. A displacement in expression space and an edit to a token sequence cannot be given equal budgets, so the matched-cost comparison of Section 3.2 would need to be redesigned. In addition, the perturbation capability of scFoundation is not a property of the pretrained model: its published perturbation results pass the pretrained gene embeddings to GEARS [32], which is itself trained on measurements from the system being perturbed. Practically,

the method needs a differentiable score over an input that the experimenter controls, and the form of that input determines which attack algorithm can be borrowed, not whether one exists.

4.8 Limitations

The advantage over greedy at its natural budget was demonstrated in one of the two models; at matched cost it held in both (Section 4.6). The non-additivity measurement and the set-level analysis were run on Geneformer only, and the comparison at the natural budget of greedy and the 2×2 decomposition used five of the 18 pathways. The benchmark consists of 18 conversions for which both cell types are present in an adult atlas, and conversion to pluripotency (the Yamanaka factors) is therefore not included. In addition, the formulation is a search specific to an input, namely a given source cell, which differs from the search for factors common to arbitrary cells (a universal perturbation). The latter can be incorporated into the same framework by maximizing the objective averaged over many source cells, as universal adversarial suffixes are optimized over multiple prompts [21, 23], and its examination is left for future work. The objective is embedding-space similarity rather than a measured biological outcome, and experimental validation of the returned sets is a subject for future work; a set of eight genes is small enough to be tested, and whether the optimum works is the question that matters next. The curated benchmark is partly circular, eight of the 18 pathways have three or fewer curated factors, and the pathway-specific AUC takes the factors of other pathways as negatives, which favours rankings that reflect expression specific to the target, so that absolute rates should be read against chance and AUC differences between rankings as descriptions of the benchmark.

The values are specific to one insertion convention in Geneformer and to untuned batch and pool sizes, three restarts and a single seed, and cost was counted in forward passes (a gradient as two), so that wall-clock ratios may differ. Tabula Sapiens may partly overlap with the pretraining data of the models [4, 6]. Because the models were not fine-tuned and the cells serve only to define source and target states, any overlap is shared by all methods compared here and does not affect the comparisons among them; confirming the absolute values on data that were not part of pretraining is a natural next step.

5 Conclusions

Our results suggest that attack methods developed for language models, in particular GCG, are useful for finding combinations of genes with token-based single-cell foundation models. The search also showed that the optimum of the induced objective does not necessarily coincide with reported factors, a property of the models that single-gene evaluation does not reveal. These results are obtained within the constraints of the foundation models, and the validation of the returned combinations is a subject for future work. Nevertheless, the ability to search for combinations of factors with foundation models extends *in silico* perturbation, which has been applied to single

genes, to combinations, and points to a new direction for both the use and the evaluation of the models. We expect this direction to develop as more foundation models become available.

Supplementary information

Additional file 1. Format: PDF. Title: Supplementary text, tables and figures. Description: implementation details and reproducibility, order sensitivity and the size of the search space, protocol selection, representativeness of the five pathways, distance to the optimum, simple and main effects of the 2×2 decomposition, spread and sign structure of the single-gene ranking evaluation, complementarity of the search with the expression baselines (Supplementary Text); the eighteen reprogramming pathways (Table S1); paired t-tests and Wilcoxon signed-rank tests (Table S2); order sensitivity (Figure S1); distance to the optimum (Figure S2).

Declarations

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Availability of data and materials. The datasets analysed in this study are publicly available. Tabula Sapiens (v1 and v2) is available from the CZ CELLxGENE Discover data portal as the CELLxGENE collection https://cellxgene.cziscience.com/collections/e5f58829-1a66-40b5-a624-9046778e74f5 [33] and was accessed through the CELLxGENE Discover Census, release 2025-11-08 (https://chanzuckerberg.github.io/cellxgene-census/) [34]. The list of human transcription factors was obtained from AnimalTFDB, version 4.0 (http://bioinfo.life.hust.edu.cn/AnimalTFDB4/) [35]. The pretrained models Geneformer (V1; Hugging Face, https://huggingface.co/ctheodoris/Geneformer) [36] and scGPT (scGPT_human; GitHub, https://github.com/bowang-lab/scGPT) [37] are publicly available. The source publications of the reprogramming protocols used for the benchmark are listed in Additional file 1: Table S1. The code, the factor sets of the protocols and the definitions of the pathways are available on GitHub (https://github.com/kztakemoto/advCell) [38] and archived on Zenodo (https://doi.org/10.5281/zenodo.23521218) [39].

Competing interests. The authors declare that they have no competing interests.

Funding. This work was supported by JSPS KAKENHI (grant no. 26K03029).

Authors' contributions. KT conceived and designed the study. MH and KT curated the data. YO and KT implemented the methods and performed the analyses. KT wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements. Not applicable.

References

[1] Takahashi K, Yamanaka S. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. Cell. 2006;126(4):663–676. https://doi.org/10.1016/j.cell.2006.07.024.

[2] Vierbuchen T, Ostermeier A, Pang ZP, Kokubu Y, Südhof TC, Wernig M. Direct conversion of fibroblasts to functional neurons by defined factors. Nature. 2010;463:1035–1041. https://doi.org/10.1038/nature08797.

[3] Ieda M, Fu JD, Delgado-Olguin P, Vedantham V, Hayashi Y, Bruneau BG, et al. Direct reprogramming of fibroblasts into functional cardiomyocytes by defined factors. Cell. 2010;142(3):375–386. https://doi.org/10.1016/j.cell.2010.07.002.

[4] Theodoris CV, Xiao L, Chopra A, Chaffin MD, Al Sayed ZR, Hill MC, et al. Transfer learning enables predictions in network biology. Nature. 2023;618(7965):616–624. https://doi.org/10.1038/s41586-023-06139-9.

[5] Norman TM, Horlbeck MA, Replogle JM, Ge AY, Xu A, Jost M, et al. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. Science. 2019;365(6455):786–793. https://doi.org/10.1126/science.aax4438.

[6] Cui H, Wang C, Maan H, Pang K, Luo F, Duan N, et al. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. Nature Methods. 2024;21(8):1470–1480. https://doi.org/10.1038/s41592-024-02201-0.

[7] Yang F, Wang W, Wang F, Fang Y, Tang D, Huang J, et al. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. Nature Machine Intelligence. 2022;4(10):852–866. https://doi.org/10.1038/s42256-022-00534-z.

[8] Yang X, Liu G, Feng G, Bu D, Wang P, Jiang J, et al. GeneCompass: deciphering universal gene regulatory mechanisms with a knowledge-informed cross-species foundation model. Cell Research. 2024;34(12):830–845. https://doi.org/10.1038/s41422-024-01034-y.

[9] Hao M, Gong J, Zeng X, Liu C, Guo Y, Cheng X, et al. Large-scale foundation model on single-cell transcriptomics. Nature Methods. 2024;21(8):1481–1491. https://doi.org/10.1038/s41592-024-02305-7.

[10] Ahlmann-Eltze C, Huber W, Anders S. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. Nature Methods. 2025;22(8):1657–1661. https://doi.org/10.1038/s41592-025-02772-6.

[11] Wong DR, Hill AS, Moccia R. Simple controls exceed best deep learning algorithms and reveal foundation model effectiveness for predicting genetic perturbations. Bioinformatics. 2025;41(6):btaf317. https://doi.org/10.1093/

bioinformatics/btaf317.

[12] Kedzierska KZ, Crawford L, Amini AP, Lu AX. Zero-shot evaluation reveals limitations of single-cell foundation models. Genome Biology. 2025;26(1):101. https://doi.org/10.1186/s13059-025-03574-x.

[13] Liu X, Boylan J, Bouiller T, Solovyeva E, Ataman B, Hoersch S, et al. Evaluating foundation models for in-silico perturbation. bioRxiv. 2025;Preprint. https://doi.org/10.1101/2025.05.11.653338.

[14] Rackham OJL, Firas J, Fang H, Oates ME, Holmes ML, Knaupp AS, et al. A predictive computational framework for direct reprogramming between human cell types. Nature Genetics. 2016;48(3):331–335. https://doi.org/10.1038/ng.3487.

[15] Kurochkin I, Altman AR, Caiado I, Pértiga-Cabral D, Halitzki E, Minaeva M, et al. A combinatorial transcription factor screening platform for immune cell reprogramming. Cell Systems. 2026;17(1):101457. https://doi.org/10.1016/j.cels.2025.101457.

[16] Cahan P, Li H, Morris SA, Lummertz da Rocha E, Daley GQ, Collins JJ. CellNet: network biology applied to stem cell engineering. Cell. 2014;158(4):903–915. https://doi.org/10.1016/j.cell.2014.07.020.

[17] Morris SA, Cahan P, Li H, Zhao AM, San Roman AK, Shivdasani RA, et al. Dissecting engineered cell types and enhancing cell fate conversion via CellNet. Cell. 2014;158(4):889–902. https://doi.org/10.1016/j.cell.2014.07.021.

[18] D'Alessio AC, Fan ZP, Wert KJ, Baranov P, Cohen MA, Saini JS, et al. A systematic approach to identify candidate transcription factors that control cell identity. Stem Cell Reports. 2015;5(5):763–775. https://doi.org/10.1016/j.stemcr.2015.09.016.

[19] Xu Q, Georgiou G, Frölich S, van der Sande M, Veenstra GJC, Zhou H, et al. ANANSE: an enhancer network-based computational approach for predicting key transcription factors in cell fate determination. Nucleic Acids Research. 2021;49(14):7966–7985. https://doi.org/10.1093/nar/gkab598.

[20] Kamimoto K, Stringa B, Hoffmann CM, Jindal K, Solnica-Krezel L, Morris SA. Dissecting cell identity via network inference and in silico gene perturbation. Nature. 2023;614(7949):742–751. https://doi.org/10.1038/s41586-022-05688-9.

[21] Wallace E, Feng S, Kandpal N, Gardner M, Singh S. Universal adversarial triggers for attacking and analyzing NLP. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019. p. 2153–2162.

[22] Carlini N, Nasr M, Choquette-Choo CA, Jagielski M, Gao I, Koh PWW, et al. Are aligned neural networks adversarially aligned? In: Advances in Neural Information Processing Systems. vol. 36; 2023. .

[23] Zou A, Wang Z, Carlini N, Nasr M, Kolter JZ, Fredrikson M. Universal and transferable adversarial attacks on aligned language models. arXiv preprint arXiv:230715043. 2023;https://doi.org/10.48550/arXiv.2307.15043.

[24] The Tabula Sapiens Consortium. The Tabula Sapiens: a multiple-organ, single-cell transcriptomic atlas of humans. Science. 2022;376(6594):eabl4896. https://doi.org/10.1126/science.abl4896.

[25] The Tabula Sapiens Consortium, Quake SR. Tabula Sapiens reveals transcription factor expression, senescence effects, and sex-specific features in cell types from 28 human organs and tissues. bioRxiv. 2024;Preprint; version 2 posted 2025-08-27. https://doi.org/10.1101/2024.12.03.626516.

[26] CZI Cell Science Program, Abdulla S, Aevermann B, Assis P, Badajoz S, Bell SM, et al. CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. Nucleic Acids Research. 2025;53(D1):D886–D900. https://doi.org/10.1093/nar/gkae1142.

[27] Shen WK, Chen SY, Gan ZQ, Zhang YZ, Yue T, Chen MM, et al. AnimalTFDB 4.0: a comprehensive animal transcription factor database updated with variation and expression annotations. Nucleic Acids Research. 2023;51(D1):D39–D45. https://doi.org/10.1093/nar/gkac907.

[28] Liao Z, Sun H. AmpleGCG: learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed LLMs. In: Conference on Language Modeling (COLM); 2024. ArXiv:2404.07921.

[29] Tan Y, Li X, Li Z, Shu H, Hu P. The resurgence of GCG adversarial attacks on large language models. arXiv preprint arXiv:250900391. 2025;https://doi.org/10.48550/arXiv.2509.00391.

[30] Demir A, Anderson EW, Jenkins JL, Mukherjee S. CellxPert: inference-time MCMC steering of a multi-omics single-cell foundation model for in-silico perturbation. In: ICLR Machine Learning for Genomics Explorations Workshop; 2026. ArXiv:2605.00930.

[31] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (ICLR); 2018. ArXiv:1706.06083.

[32] Roohani Y, Huang K, Leskovec J. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. Nature Biotechnology. 2024;42(6):927–935. https://doi.org/10.1038/s41587-023-01905-6.

[33] The Tabula Sapiens Consortium. Tabula Sapiens collection. CZ CELLxGENE Discover; 2026. Accessed 10 Oct 2026. Available from: https://cellxgene.cziscience.com/collections/e5f58829-1a66-40b5-a624-9046778e74f5.

[34] CZ CELLxGENE Discover. Census documentation; 2026. Accessed 10 Oct 2026. Available from: https://chanzuckerberg.github.io/cellxgene-census/.

[35] AnimalTFDB. Animal transcription factor database, version 4.0; 2026. Accessed 10 Oct 2026. Available from: http://bioinfo.life.hust.edu.cn/AnimalTFDB4/.

[36] Theodoris CV. Geneformer. Hugging Face; 2026. Accessed 10 Oct 2026. Available from: https://huggingface.co/ctheodoris/Geneformer.

[37] Bo Wang Lab. scGPT. GitHub; 2026. Accessed 10 Oct 2026. Available from: https://github.com/bowang-lab/scGPT.

[38] Oishi Y, Hamano M, Takemoto K. advCell. GitHub; 2026. Accessed 21 Oct 2026. Available from: https://github.com/kztakemoto/advCell.

[39] Oishi Y, Hamano M, Takemoto K. advCell. Zenodo; 2026. Accessed 21 Oct 2026. Available from: https://doi.org/10.5281/zenodo.23521218.

Figures

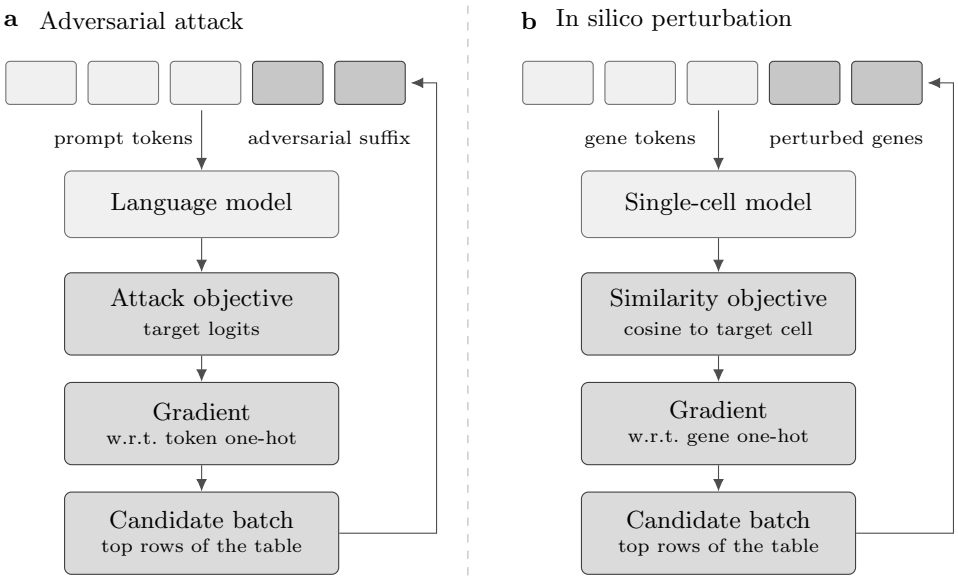


Fig. 1 In silico perturbation is a discrete adversarial attack. (a) GCG constructs an adversarial suffix for a language model: the gradient of the attack objective with respect to the one-hot encoding of the suffix tokens is projected onto the token embedding table, the highest-scoring rows form a batch of candidate substitutions, the best candidate is adopted, and the loop repeats. (b) The same procedure searches over perturbation sets for a single-cell foundation model, with gene tokens in place of prompt tokens, k overexpressed genes in place of the suffix, and cosine similarity to a target cell centroid in place of the attack objective. No component of the algorithm is changed between the two panels. The panels are drawn alike to make the correspondence visible, and the decision variable is drawn last in both for that reason; where it actually sits in the sequence differs by model, at the front of the rank-ordered sequence for Geneformer and in place for scGPT, which rewrites the expression value of a gene the cell already has (Section 2.1). The correspondence is exact for Geneformer, whose rank value encoding makes the perturbed genes ordered as suffix tokens are ordered; for scGPT the suffix collapses to an unordered set (Additional file 1: Figure S1).

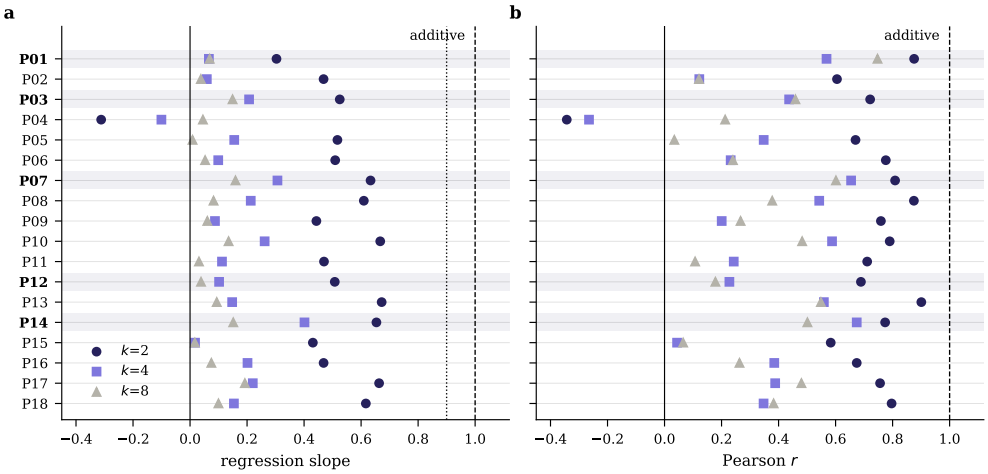


Fig. 2 Measured set effects against the additive prediction in Geneformer. (a) Slope of the regression of the measured improvement of random perturbation sets on the sum of their members' individual effects, for each of the 18 pathways and each set size. An additive objective would give 1.00 (dashed line); the dotted line marks 0.9. (b) The corresponding Pearson correlations. Shaded rows are the five pathways used for the 2×2 decomposition (Section 3.3). Each point summarizes 10000 random sets over 50 cells.

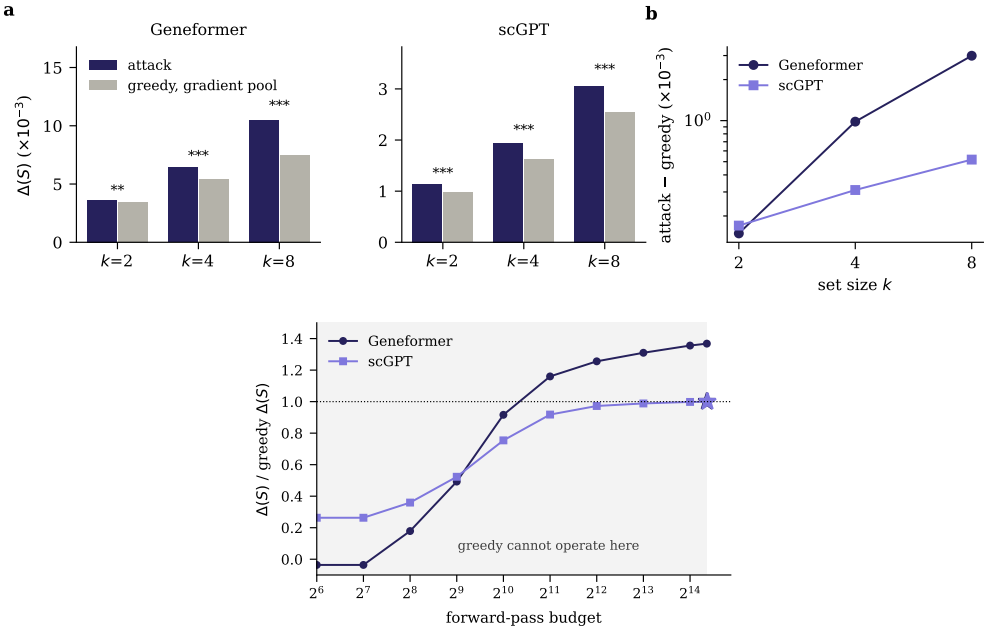


Fig. 3 Comparison of the attack with greedy search at matched forward-pass budget. (a) Mean improvement (in units of 10^{-3}) by model and set size for the attack and for greedy search restricted to an equal-cost candidate pool. Significance marks are paired t-tests against gradient-pool greedy ($n = 360$ source cells). (b) The difference between the attack and gradient-pool greedy (in units of 10^{-3}) against set size. (c) The improvement of the attack as a fraction of that of full greedy search, against budget. Greedy occupies a single point (star) at $k \times n$ and cannot be evaluated anywhere in the shaded region.

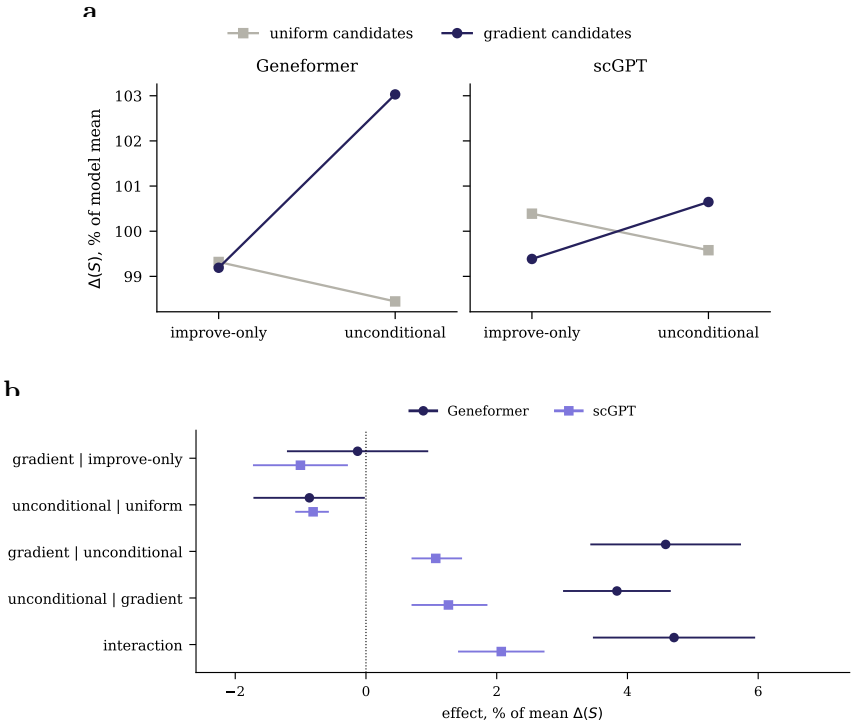


Fig. 4 Contrasts of the 2×2 decomposition of the attack. (a) Mean improvement for each cell of the 2×2, by model. (b) Simple effects and the interaction, normalized bqy each model's mean improvement so that the two models can be compared on one axis. Paired over $n = 100$ cells; bars are 90% confidence intervals.

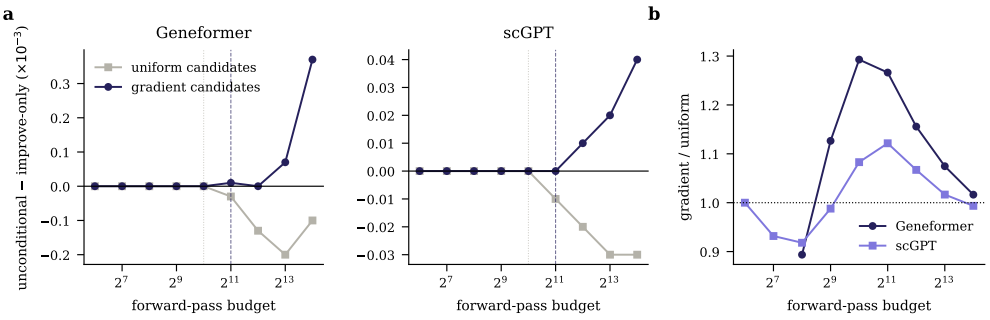


Fig. 5 Divergence of the two acceptance rules with budget. (a) Difference between the two acceptance rules, unconditional minus improve-only, at each budget, plotted separately for uniform and for gradient-guided candidates (in units of 10^{-3}). The difference is zero to floating-point precision through 512 forward passes. The vertical lines mark the first budget at which it is non-zero (dotted for uniform candidates, dashed for gradient-guided ones); at that budget it is small, and in some panels below the resolution of the plot. With uniform candidates it first departs from zero at 1024 and is negative from 2048 on in both models. With gradient-guided candidates it is positive at 2048 in Geneformer, negative at 4096 and positive thereafter, and negative at 2048 in scGPT and positive thereafter. (b) The ratio of gradient-guided to uniform candidates under improve-only acceptance, which peaks at intermediate budget and returns to, or falls below, unity.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- additionalfile1.pdf