SENTRY-BD: BILEVEL MEASUREMENT DESIGN FOR CROSS-CONTEXT PERTURBATION RECOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce SENTRY-BD, a framework that learns which genes to measure by optimizing cross-context perturbation recovery after refitting the predictor. Compact panels could extend perturbation screens across cell states and time points, but useful measurements must predict the endpoint and remain informative alongside the other selected genes. We formulate panel learning as a budget-constrained bilevel problem: a shared gene ordering determines context-grouped measurement weights, an analytic inner model fits the response map, and disjoint query perturbations train the design. An exact gene-utility derivative accounts for decoder refitting; a signed response loss complements magnitude reconstruction. One training run produces nested panels at three budgets. On an independent confirmation cohort of 1,699 primary CD4 perturbations, a 128-gene panel recovers a 10,282-gene endpoint with 1.711% lower MSE than a representation-matched PERSIST adaptation (95% CI: 1.492–1.945%). The reduction is 3.498% over 1,676 K562-to-RPE1 perturbations. Five training seeds and component controls show that learning through query prediction provides the main gain. The framework connects measurement allocation to the response prediction task through a differentiable, computationally compact design objective.

1 INTRODUCTION

Perturbation screens reveal how genetic interventions change cellular behavior. Perturb-seq and related methods combine pooled genetic interventions with single-cell transcriptional readouts, allowing many regulatory effects to be studied in parallel (Dixit et al., 2016; Adamson et al., 2016; Jaitin et al., 2016; Datlinger et al., 2017). These measurements have mapped genetic interactions, organized essential genes into functional programs, and exposed regulators whose effects depend on cellular context (Norman et al., 2019; Replogle et al., 2022; Zhu et al., 2026). The growing scientific opportunity is comparative: understanding how a response changes across cell types, stimulation conditions, and observation times.

The number of required measurements grows with these comparisons. A reference screen may provide complete profiles, whereas a larger follow-up experiment may only support a small, targeted gene panel. Targeted sequencing and direct guide capture already provide practical ways to focus perturbation readouts (Schraivogel et al., 2020; Replogle et al., 2020). Landmark expression profiling and compressed imaging offer further evidence that informative high-dimensional responses can be recovered from limited measurements (Subramanian et al., 2017; Cleary et al., 2021). This motivates a concrete design problem: use a paired reference atlas to choose a fixed set of source genes, then measure those genes for new perturbations to recover their responses in another context. For example, measurements at rest and eight hours after stimulation could support recovery of a later T-cell response.

Choosing that panel requires more than identifying variable or individually predictive genes. A gene that changes strongly in the source context may provide little information about the endpoint. Two useful genes can also become redundant once measured together. Feature-ranking and residual-selection methods address parts of this problem, but their scores need not track the quality of the predictor obtained after the panel changes. The relevant value of a measurement is therefore conditional: how does it affect prediction on another perturbation once the response map is refitted?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

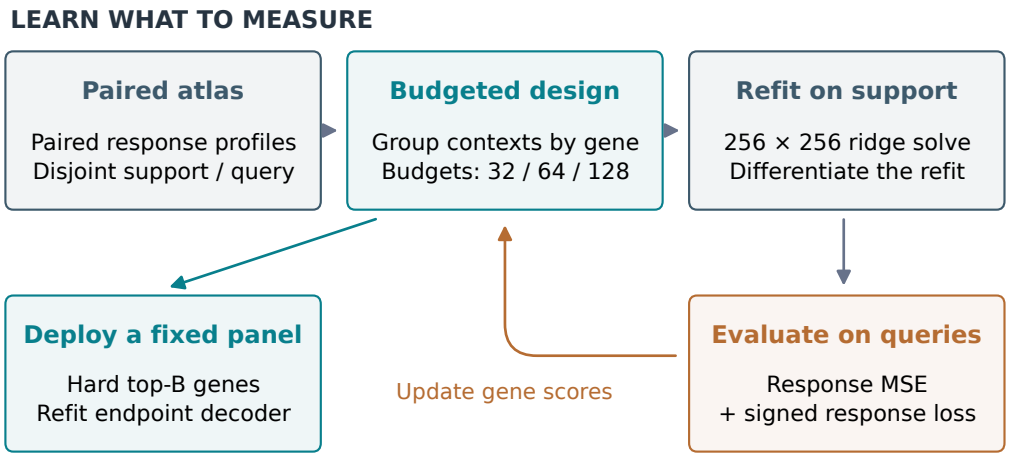

Figure 1: **Learn measurements through the predictor they support.** Each episode separates construction-atlas perturbations into support and query groups. Gene weights obey a budget and are shared across source contexts. The support model is solved analytically; query reconstruction and signed response errors train the weights through that solve. A shared ordering yields nested hard panels, each refitted with the same endpoint decoder used for the baselines. Deployment requires only the selected source measurements.

We answer this question with a bilevel measurement-design formulation. The outer problem learns gene weights under a cardinality budget. The inner problem fits a cross-context response predictor using the weighted measurements. Query perturbations then evaluate the fitted predictor, providing a training signal for the design itself. Because the inner model is ridge regression in a compact response space, the effect of refitting can be differentiated exactly. The resulting derivative has an interpretable structure: it couples the candidate gene's unexplained query measurements to the endpoint responses encoded by the support fit. This makes joint measurement utility an explicit part of learning.

SENTRY-BD integrates this formulation into a practical workflow. A single score vector is trained across several budgets; an entropy-constrained gate gives differentiable gene weights, and the corresponding hard panels share one ordering. A distribution over positive and negative response coordinates emphasizes response direction alongside MSE. Batched GPU kernel operations keep the inner system small even when every gene is a candidate. The same framework supports multiple source contexts and single-source transfer between cell lines.

Our contributions are threefold. First, we formulate cross-context gene-panel design through an analytically refitted predictor, with context-grouped weights and a shared ordering across budgets. Second, we derive the exact measurement-utility gradient and a finite-change identity connecting soft designs to hard panels, and implement the resulting objective with a signed response term. Third, we evaluate the complete workflow on an independent CD4 confirmation cohort and a five-fold K562-to-RPE1 task, using common decoders, representation-matched controls, five training seeds, and component ablations. Together, these experiments identify when learning the design improves endpoint recovery and which part of the formulation produces the gain.

2 RELATED WORK

Learning compact measurement panels. Gene-panel methods target cell-type discrimination, local expression structure, or reconstruction, as illustrated by scGeneFit, geneBasis, and PERSIST (Dumitrascu et al., 2021; Missarova et al., 2021; Covert et al., 2023). Concrete autoencoders and stochastic gates provide differentiable mechanisms for learning sparse inputs (Balın et al., 2019; Yamada et al., 2020). These objectives express familiar relevance–redundancy tradeoffs (Guyon &

Elisseeff, 2003; Peng et al., 2005). Here, selected source measurements must support recovery in another context.

Task-driven experimental design. Classical sensor placement uses uncertainty, reconstruction geometry, or information gain to allocate measurements (Joshi & Boyd, 2009; Krause et al., 2008; Manohar et al., 2018). Matrix subset selection gives a related perspective on retaining informative columns (Drineas et al., 2008). TADRED optimizes measurement channels jointly with an imaging task (Blumberg et al., 2024); distributionally robust feature selection studies how a limited feature set supports downstream models across populations (Swaroop et al., 2025). These recent approaches motivate evaluating a design through its predictive use. We develop this perspective for grouped cross-context response measurements, using disjoint perturbation queries and an exact inner solve. The entropy projection follows Amos et al. (2019), while recent bilevel optimization research provides a broader context for differentiating through fitted learners (Yang et al., 2025). The particular inner problem here gives a direct formula for the value of each gene.

Perturbation response modeling and evaluation. scGen, CPA, GEARS, and CellOT learn response transformations using latent representations, compositional structure, gene relationships, or transport (Lotfollahi et al., 2019; 2023; Roohani et al., 2024; Bunne et al., 2023). Foundation models such as scGPT provide another source of expression representations (Cui et al., 2024). Our task uses observed source responses for each query perturbation and asks which of those measurements are sufficient for endpoint recovery. Recent evaluations emphasize strong linear baselines and metrics that separate perturbation-specific effects from shared response patterns (Ahlmann-Eltze et al., 2025; Viñas Torné et al., 2026). We therefore hold the final decoder fixed across selectors and report both reconstruction error and response-specific readouts.

3 Problem Setup

A construction atlas contains N paired perturbation responses. For source context $c \in \{1, \dots, C\}$, let $X^{(c)} \in \mathbb{R}^{N \times G}$ contain the measured gene responses, and let $Y \in \mathbb{R}^{N \times G}$ denote the endpoint. A panel $S \subseteq \{1, \dots, G\}$ contains B genes. Measuring one panel gene provides its response in every available source context, giving CB input values per perturbation. The deployment map is

$$\left(x_S^{(1)}, \dots, x_S^{(C)}\right) \longmapsto \widehat{y} \in \mathbb{R}^G. \tag{1}$$

The panel is shared by all query perturbations. Learning uses complete construction profiles; deployment uses only the selected inputs.

Each source gene is centered and scaled using construction statistics. The endpoint is standardized similarly, with mean μ and diagonal scale matrix D. A rank-r decomposition of the standardized endpoint gives scores $Z \in \mathbb{R}^{N \times r}$ and a row-orthonormal basis $V \in \mathbb{R}^{r \times G}$ (Halko et al., 2011; Izenman, 1975). Predicted scores F map to response units as $\widehat{Y} = \mu + FVD$. We use $r = 64$ throughout the decoder comparison. The directly perturbed endpoint gene is mean-imputed before fitting and omitted from evaluation, so scores measure recovery of downstream responses. Standardizers and the basis use the construction atlas; within that atlas, support and query identities are separated for each predictor fit.

4 Bilevel Measurement Design

The framework learns a score vector $s \in \mathbb{R}^G$. For each training episode, a budget B determines a soft measurement design, support perturbations fit the response map, and query perturbations evaluate that map. Gradients update s through both the budget gate and the fitted predictor. Figure 1 summarizes the process.

4.1 One gene ordering across measurement budgets

For temperature $\tau > 0$, the design weights solve

$$w_B(s) = \underset{0 < w_g < 1,\ \sum_g w_g = B}{\arg\max} \left\{ s^\top w + \tau \sum_g h(w_g) \right\}. \tag{2}$$

Here $h(u) = -u \log u - (1-u)\log(1-u)$ is binary entropy. This limited multi-label projection (Amos et al., 2019) yields $w_g = \sigma((s_g - \nu_B)/\tau)$, where the scalar ν_B enforces the budget. Its implicit derivative is

$$J_B = \mathrm{diag}(t) - \frac{tt^\top}{\sum_g t_g}, \qquad t_g = \frac{w_g(1-w_g)}{\tau}. \tag{3}$$

The shared threshold preserves the ordering of s at every budget. Hard deployment therefore selects the top B scores, producing nested panels. Training cycles through $B \in \{32, 64, 128\}$ so that one ordering serves all three measurement budgets. Each weight applies to the full group of context measurements for its gene.

4.2 Differentiate the predictor after refitting

In one episode, let $X_g \in \mathbb{R}^{n\times C}$ contain the support measurements for gene g and $Q_g \in \mathbb{R}^{m\times C}$ its query measurements. Let Z_T be the support endpoint scores. Scaling every feature in group g by $\sqrt{w_g}$ defines a weighted ridge predictor (Hoerl & Kennard, 1970). Its dual form is

$$\begin{aligned} K(w) &= \sum_g w_g X_g X_g^\top, & C(w) &= \sum_g w_g Q_g X_g^\top, \\ A(w) &= K(w) + \alpha I, & F(w) &= C(w)A(w)^{-1}Z_T. \end{aligned} \tag{4}$$

We solve the linear system directly. Its dimension is the support size n, independent of the number of candidate genes. The weighted model is refitted at every design update.

The central design question is how changing one gene weight changes the refitted query prediction. Define $H = A^{-1}Z_T$ and $P = CA^{-1}$. The following identity gives the answer.

Proposition 1 (Conditional measurement utility)**.** *For $\alpha > 0$ and nonnegative w, the query prediction in Equation* (4) *is differentiable and satisfies*

$$\frac{\partial F}{\partial w_g} = \underbrace{(Q_g - PX_g)}_{\textit{query innovation}} \underbrace{X_g^\top H}_{\textit{endpoint coupling}}. \tag{5}$$

For an outer loss $\mathcal{L}(F)$ with derivative E, the gene gradient is $\partial\mathcal{L}/\partial w_g = \langle E, (Q_g - PX_g)X_g^\top H\rangle_F$.

The proof differentiates the kernel and its regularized solve (Appendix A). The first factor measures what the candidate adds at the query after accounting for the current support predictor. The second connects that information to endpoint variation. Redundancy and predictive relevance thus enter the same utility calculation, including the response of the fitted decoder to the measurement change. Combining Equation (5) with the budget Jacobian trains all gene scores jointly.

A finite-change version also links a relaxed design w to a hard panel v:

$$F(v) - F(w) = \big[\Delta C - P(w)\Delta K\big]A(v)^{-1}Z_T, \tag{6}$$

where $\Delta K = K(v) - K(w)$ and $\Delta C = C(v) - C(w)$. The difference depends on the same conditional innovation structure as the derivative. Appendix A proves the identity and a corresponding norm bound.

4.3 Train on response magnitude and direction

Query predictions are mapped back to centered response units, $\widehat{R} = FVD$. Let R contain the centered query endpoints and let v_Y be the mean squared centered response in the construction atlas. The primary outer loss is masked query MSE normalized by v_Y. To represent response direction as well as magnitude, we define a distribution over signed gene coordinates:

$$\begin{aligned} p_R &= \mathrm{softmax}([R, -R]/\sqrt{v_Y}), \\ \mathcal{L} &= \frac{\|\widehat{R} - R\|_\Omega^2}{mGv_Y} + \eta\,\frac{1}{m}\sum_i D_{\mathrm{KL}}\Big(p_{R_i} \| p_{\widehat{R}_i}\Big). \end{aligned} \tag{7}$$

Here Ω excludes each query's directly perturbed gene; both signed coordinates of that gene are excluded from the distribution. The signed loss places greater probability on strong responses while retaining whether each response increases or decreases. The MSE term anchors the prediction to the measured response scale.

We minimize the expected episode loss over support/query samples and budgets using Adam. Each episode draws 256 support and 128 distinct query perturbations. Training uses 360 updates, learning rate 0.04, $\alpha = 1.5n$, and a geometric temperature schedule from 0.5 to 0.05. The signed weight $\eta = 0.1$ is selected on an internal construction split. Scores start from a small standardized log-covariance ranking with seed-dependent noise. After learning, each hard panel is refitted on all construction perturbations using the common rank-64 ridge decoder. Appendix B gives the complete training and deployment procedure.

5 EXPERIMENTS

Our experiments address three questions: whether learning through the fitted predictor improves recovery, whether the improvement transfers across tasks and budgets, and which design choices account for it.

5.1 DATA, CONTROLS, AND EVALUATION

Primary CD4 confirmation. We use author-released differential-response profiles from a primary human CD4 Perturb-seq screen (Zhu et al., 2026). Source contexts are rest and eight-hour stimulation; the endpoint is the 48-hour response across 10,282 genes. The construction atlas contains 6,144 perturbations. A salted identity hash splits it into 4,904 fitting and 1,240 validation perturbations for method development. Five candidate loss settings are evaluated by their mean raw MSE across the three budgets, and the selected recipe is refitted on all construction data.

The primary confirmation cohort contains 1,699 additional perturbations in the previously unused upper 15% of the original identity-hash partition. These identities are disjoint from construction and from an earlier 1,636-perturbation evaluation cohort. The design recipe and primary comparisons were recorded before confirmation outcomes were computed. The earlier cohort provides a secondary comparison in Appendix C; it is not used for the primary claim. Eligibility requires at least 20 cells in every context and complete source–endpoint coverage.

Transfer between cell lines. The Replogle dataset (Replogle et al., 2022) supplies a second task, using K562 responses to recover RPE1 responses for 1,676 perturbations and 7,226 genes. We use five identity-hash folds, fitting the panel and decoder independently on the other four folds. The CD4 design settings transfer directly to this task. Metrics pool the out-of-fold predictions over all perturbations. Each retained target has at least 30 cells in both cell lines. Both tasks use aggregate response profiles; Appendix C details their construction.

Selection controls. Every method receives the same construction information, gene budget, and final rank-64 decoder. Variance selection ranks pooled source variation. PLS-VIP uses a supervised response representation (Wold et al., 2001). Residual selection scores source genes against unexplained endpoint coordinates and repeatedly refits a growing panel, following the pursuit principle (Mallat & Zhang, 1993). We compare its 8D implementation with a 64D control that also removes QR batch filtering. PLS-VIP is evaluated at both ranks. Adapted PERSIST uses its supervised MSE selector, a 1,000-gene variance prefilter, and five training seeds selected by inner validation error (Covert et al., 2023). A further PERSIST control uses the same 64D target representation as SENTRY-BD. These rank-matched controls distinguish the design objective from the amount of endpoint information supplied to the selector.

Metrics and uncertainty. The primary metric is raw endpoint MSE at 128 genes, averaged first over downstream genes and then over perturbations. Additional metrics include gene-standardized MSE, Pearson correlation, signed top-20 overlap, correlation after subtracting the construction endpoint mean (ρ_{Δ}), and centroid accuracy inspired by Viñas Torné et al. (2026). Centroid accuracy measures how often a prediction is closer to its own endpoint than to another perturbation's endpoint. Paired bootstrap intervals use 5,000 perturbation resamples (Efron, 1979). They describe

Table 1: **Endpoint recovery with 128 measured genes.** CD4 results use the independent 1,699-perturbation confirmation cohort; K562-to-RPE1 results pool 1,676 out-of-fold predictions. Lower MSE and higher ρ_Δ and signed top-20 overlap are better. All panels use the same final decoder. Best entries are bold. 8D/64D denotes the selector's endpoint representation.

	CD4: rest + 8 h → 48 h			K562 → RPE1		
Selector	MSE ↓	ρ_Δ ↑	Top-20 ↑	MSE ↓	ρ_Δ ↑	Top-20 ↑
Variance	0.08005	0.207	**0.225**	0.03926	0.443	**0.274**
PLS-VIP (8D)	0.07949	0.213	0.213	0.03904	0.433	0.253
PLS-VIP (64D)	0.07947	0.213	0.213	0.03904	0.434	0.254
PERSIST (8D)	0.07972	0.211	0.218	0.03917	0.436	0.259
PERSIST (64D)	0.07957	0.214	0.216	0.03925	0.435	0.260
Residual selection (8D)	0.07891	0.227	0.215	0.03846	0.442	0.261
Residual selection (64D)	0.07873	0.231	0.216	0.03794	**0.447**	0.267
SENTRY-BD	**0.07821**	**0.243**	0.219	**0.03788**	0.444	0.272

evaluation-cohort variation for fitted models; a separate five-seed experiment measures stochastic panel-learning variation. Metric definitions and all additional outputs are provided in Appendix D.

5.2 Learning the design improves endpoint recovery

At 128 genes, SENTRY-BD achieves raw MSE 0.07821 on the CD4 confirmation cohort (Table 1). This is a 1.891% reduction relative to the 8D PERSIST adaptation and a 0.884% reduction relative to 8D residual selection. The paired 95% intervals are [1.670, 2.121]% and [0.729, 1.039]%, respectively. The panel measures 1.245% of the available genes in each of two source contexts.

The improvement is accompanied by better perturbation-specific predictions. Relative to 8D residual selection, ρ_Δ increases from 0.227 to 0.243, Pearson correlation from 0.283 to 0.295, and centroid accuracy from 0.591 to 0.600. Signed top-20 overlap rises from 0.215 to 0.219. These readouts indicate that the MSE change is accompanied by a closer match to the identity and direction of downstream responses.

The 64D residual control, which also omits QR filtering, is stronger than the 8D implementation. SENTRY-BD still reduces CD4 MSE by 0.657% against this control, with a 95% interval of [0.513, 0.802]%. This stronger residual recipe narrows the gap, while the query-trained design supplies a further gain on the CD4 task.

The representation-matched PERSIST comparison leads to the same conclusion. Using 64 endpoint coordinates improves its CD4 MSE to 0.07957; SENTRY-BD retains a 1.711% advantage (95% CI: 1.492–1.945%). On Replogle, the reduction against the 64D adaptation is 3.498% (95% CI: 2.477–4.554%). Increasing the baseline's response rank therefore does not remove the advantage of the complete design workflow.

5.3 Budget transfer and the second response task

The shared ordering improves recovery across the tested budgets (Figure 3). Its value is operational as well as statistical: a user can choose among 32, 64, and 128 genes after training without running a separate selector. Residual-selection baselines also support prefixes; the distinction is that our ordering is trained against prediction losses at all three budgets.

On K562-to-RPE1 transfer, SENTRY-BD reaches MSE 0.03788, a 3.301% reduction over PERSIST (95% CI: 2.206–4.481%) and a 1.512% reduction over 8D residual selection (95% CI: 0.280–3.107%). The 64D residual control reaches 0.03794; its difference from SENTRY-BD is small and the paired interval includes zero. The second task therefore supports transfer of the learned recipe and improvement over the original selectors, while showing that a stronger residual representation closes much of the remaining gap.

The metrics also expose different panel priorities. Variance selection has the highest signed top-20 overlap in both tasks, whereas SENTRY-BD gives lower raw MSE. The 64D residual control has slightly lower standardized MSE than SENTRY-BD. These outcomes reflect the distinction between

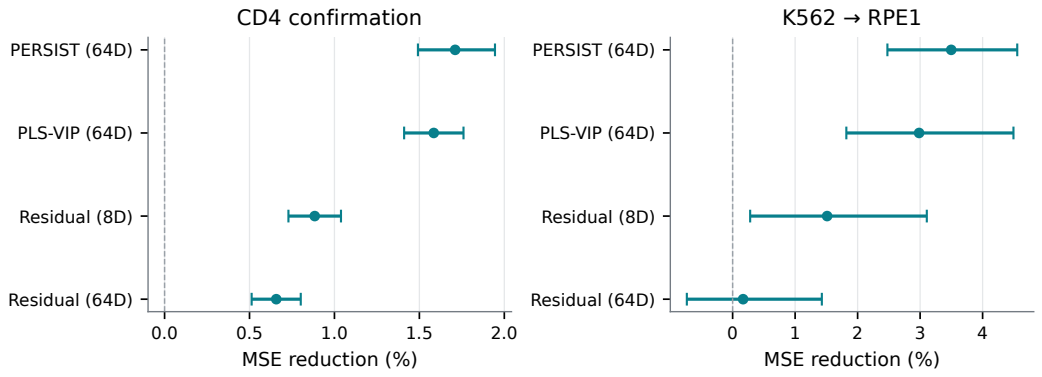

Figure 2: **Paired effect estimates at 128 genes.** Points show relative MSE reduction of SENTRY-BD against each comparator; bars show 95% percentile intervals from 5,000 paired perturbation resamples. The 64D residual control closes much of the gap on Replogle, while the CD4 difference remains positive.

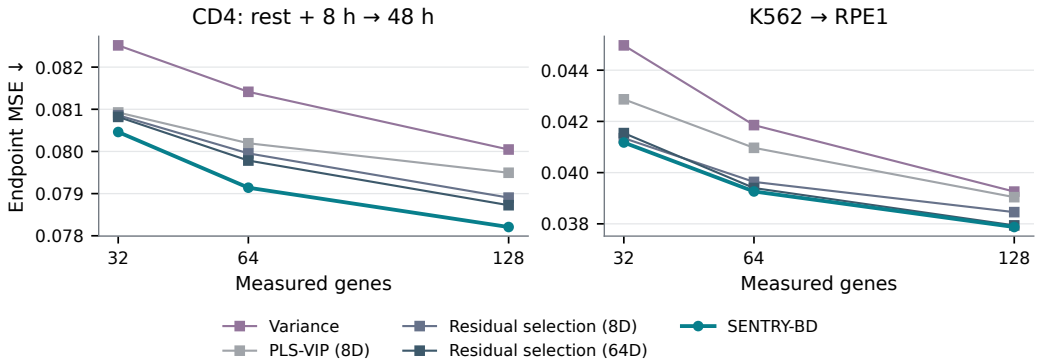

Figure 3: **One learned ordering supports several budgets.** Raw MSE is averaged over the CD4 confirmation cohort (left) and pooled Replogle out-of-fold predictions (right). Each SENTRY-BD curve uses prefixes of one ordering per training split. The comparison keeps the final decoder fixed, exposing how measurement allocation changes recovery as the panel grows.

minimizing total response error, recovering the largest changes, and weighting genes equally after standardization. Appendix D provides the complete metric comparison.

5.4 WHAT THE DESIGN LEARNS

The largest component effect comes from learning measurement utility (Table 2). Keeping the supervised initialization fixed gives MSE 0.08028, compared with 0.07821 after design training. Reusing support identities as query identities gives 0.07850. The improvement with disjoint queries supports the central premise: a measurement should be valued by the response predictor it supports on another perturbation.

Stopping the gradient through the inner fit gives MSE 0.07824, close to the full method. The gain is therefore most sensitive to learning from refitted query predictions; approximating the inner-fit derivative has a small effect in this control.

Training only at budget 128 gives MSE 0.07829, close to the full design at that budget. Joint budget training therefore supplies a reusable ordering while preserving accuracy at the largest tested panel. Restricting candidates to the 1,000-gene PERSIST pool gives MSE 0.07886, retaining an advantage over the original PERSIST adaptation. The full candidate axis contributes additional predictive information.

Table 2: **Component controls on CD4 confirmation at 128 genes.** All controls retain the common decoder. The fixed initialization control skips design learning; the query-reuse control fits and evaluates each episode on overlapping identities. The 1,000-gene pool matches the PERSIST candidate restriction.

Design setting	Raw MSE ↓	Standardized MSE ↓	Signed top-20 ↑
Fixed initialization	0.08028	0.9409	0.2147
Support reused as query	0.07850	0.9314	0.2191
Stop inner-fit gradient	0.07824	0.9289	0.2191
Train only at 128 genes	0.07829	0.9290	0.2197
Reconstruction loss only	0.07823	0.9283	0.2183
1,000-gene candidate pool	0.07886	0.9449	0.2203
Full design	0.07821	0.9267	0.2189

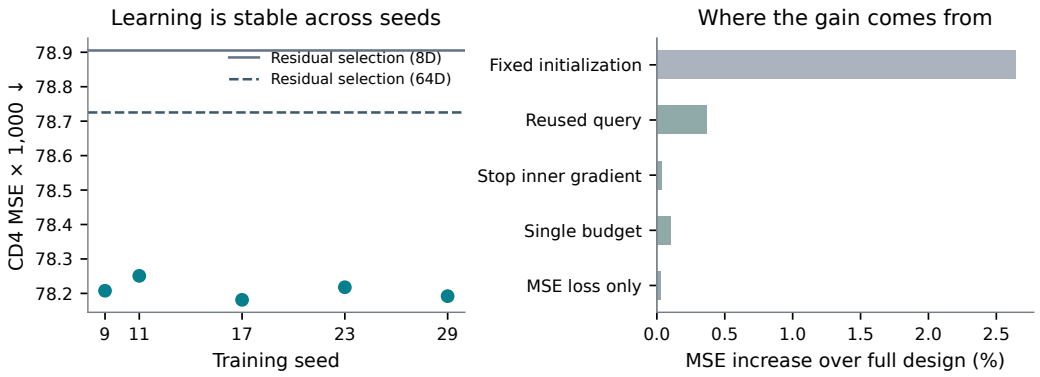

Figure 4: **The main effect survives stochastic training and component removal.** Left: five independent score initializations and episode sequences evaluated on the same CD4 confirmation cohort. Horizontal lines show the two residual controls. Right: changes in MSE when individual training choices are removed. The dominant effect is learning the design; separating support and query identities provides a further improvement.

The signed response term has a smaller effect. Removing it gives MSE 0.07823 and top-20 overlap 0.2183, compared with 0.07821 and 0.2189 for the full objective. Its paired MSE interval includes zero. We retain the term because its weight was selected by the construction-only development criterion; the component analysis identifies prediction-driven panel learning as the principal source of improvement.

5.5 Stability and Computational Behavior

Five training seeds give CD4 MSE values between 0.07818 and 0.07825. Every seed improves on both residual controls at 128 genes (Figure 4). These runs vary the initial score perturbation and all support/query draws while retaining the same construction atlas and response basis. The small variation in endpoint error shows that the primary result does not depend on the reported seed.

Predictive stability coexists with variation in the selected measurements. The five 128-gene panels have mean pairwise Jaccard overlap 0.364, whereas their MSE standard deviation is only 2.69×10^{-5}. Correlated genes can support similar response predictions through different panels. The measured benefit is therefore stable recovery quality; the selected gene set should be interpreted as a predictive measurement choice rather than a unique list of biological regulators.

The implementation evaluates all gene weights using batched matrix products and a 256×256 regularized system. On an NVIDIA A100, the 360-update CD4 learning stage takes 4.3–5.6 seconds across the five seeds, including tensor preparation and score initialization. Atlas loading, standardization, basis construction, and the final full-atlas decoder fit are separate stages. A serialized deployment model stores panel indices, construction statistics, a compact endpoint basis, and decoder

coefficients. Its panel-only interface accepts two B-column source arrays for the CD4 task. Numerical tests check the implicit gate derivative, primal–dual prediction equivalence, the gene-utility derivative, hard-rounding identity, and serialization.

A controlled benchmark with identical synthetic tensors at CD4 episode dimensions measures a median forward/backward step of 83.2 ms on four CPU threads and 9.14 ms on the A100, a $9.10\times$ speedup for that computation. The CPU and GPU losses agree to float32 precision, with maximum gradient difference 1.8×10^{-11}. Panel-only CPU inference for 1,699 responses takes 0.74 seconds in this benchmark.

6 Discussion

Measurement design and response modeling are naturally coupled: a panel is useful because of the predictor it enables. SENTRY-BD makes that relationship trainable by refitting a compact response model inside the design objective. Context grouping preserves the gene-level budget, query episodes reward transfer across perturbations, and one score ordering supports several measurement budgets. The exact derivative provides a concrete interpretation of measurement value after refitting.

The experiments show that moderate improvements in raw error can accompany clearer perturbation-specific recovery. Representation-matched controls and episode ablations locate the improvement in learning a useful panel for the refitted prediction task. This makes measurement design an explicit, testable part of the response-modeling pipeline.

The present evidence concerns aggregate responses from two source–endpoint tasks. Prospective targeted assays are needed to measure robustness to probe noise, missing measurements, and donor variation. Gene count is the measurement budget used here; assay-specific costs could be incorporated through weighted budget constraints. A further direction is to learn richer response representations within the design objective while preserving the analytic relationship between measurements and their predictive value. The current framework provides a reproducible basis for studying these extensions.

AI Use Statement

Generative AI was used substantially in research planning, implementation, experiments, mathematical derivation and checking, literature search, figure programming, and writing. Numerical claims use saved results; identities are tested independently and citations are verified against primary sources. Authors retain responsibility for the work.

Reproducibility Statement

The supplementary code includes configurations, per-perturbation metrics, panels, development candidates, component controls, and a trained model. Scripts reconstruct public inputs, run experiments, generate figures, and check numerical summaries. The appendix specifies preprocessing, metrics, and optimization.

Ethics Statement

This study uses public aggregate responses and involves no new recruitment or interventions. Prospective applications should account for assay quality and reference population coverage. Predictions support experimental analysis, with no clinical decisions evaluated here.

REFERENCES

Britt Adamson, Thomas M. Norman, Marco Jost, Min Y. Cho, James K. Nuñez, Yuwen Chen, et al. A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell*, 167(7):1867–1882.e21, 2016. doi: 10.1016/j.cell.2016.11.048.

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22(8): 1657–1661, 2025. doi: 10.1038/s41592-025-02772-6.

Brandon Amos, Vladlen Koltun, and J. Zico Kolter. The Limited Multi-Label Projection Layer. *arXiv preprint arXiv:1906.08707*, 2019.

Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete Autoencoders: Differentiable Feature Selection and Reconstruction. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 444–453, 2019. URL https://proceedings.mlr.press/v97/balin19a.html.

Stefano Blumberg, Paddy Slator, and Daniel Alexander. Experimental design for multi-channel imaging via task-driven feature selection. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39998–40024, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aec2dfc4f5e19acf05c15587c889dbc4-Paper-Conference.pdf.

Charlotte Bunne, Stefan G. Stark, Gabriele Gut, Jacobo Sarabia del Castillo, Mitch Levesque, Kjong-Van Lehmann, et al. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, 2023. doi: 10.1038/s41592-023-01969-x.

Brian Cleary, Brooke Simonton, Jon Bezney, Evan Murray, Shahul Alam, Anubhav Sinha, et al. Compressed sensing for highly efficient imaging transcriptomics. *Nature Biotechnology*, 39(8): 936–942, 2021. doi: 10.1038/s41587-021-00883-x.

Ian Covert, Rohan Gala, Tim Wang, Karel Svoboda, Uygar Sümbül, and Su-In Lee. Predictive and robust gene selection for spatial transcriptomics. *Nature Communications*, 14(1):2091, 2023. doi: 10.1038/s41467-023-37392-1.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, et al. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, 2024. doi: 10.1038/s41592-024-02201-0.

Paul Datlinger, André F Rendeiro, Christian Schmidl, Thomas Krausgruber, Peter Traxler, Johanna Klughammer, et al. Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods*, 14(3):297–301, 2017. doi: 10.1038/nmeth.4177.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, et al. Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, 167(7):1853–1866.e17, 2016. doi: 10.1016/j.cell.2016.11.038.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-Error CUR Matrix Decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008. doi: 10.1137/07070471x.

Bianca Dumitrascu, Soledad Villar, Dustin G. Mixon, and Barbara E. Engelhardt. Optimal marker gene selection for cell type discrimination in single cell analyses. *Nature Communications*, 12(1): 1186, 2021. doi: 10.1038/s41467-021-21453-4.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1), 1979. doi: 10.1214/aos/1176344552.

Isabelle Guyon and André Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. URL https://jmlr.org/papers/v3/guyon03a.html.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, 53(2):217–288, 2011. doi: 10.1137/090771806.

Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975. doi: 10.1016/0047-259x(75)90042-1.

Diego Adhemar Jaitin, Assaf Weiner, Ido Yofe, David Lara-Astiaso, Hadas Keren-Shaul, Eyal David, et al. Dissecting Immune Circuits by Linking CRISPR-Pooled Screens with Single-Cell RNA-Seq. *Cell*, 167(7):1883–1896.e15, 2016. doi: 10.1016/j.cell.2016.11.039.

S. Joshi and S. Boyd. Sensor Selection via Convex Optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009. doi: 10.1109/tsp.2008.2007095.

Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research*, 9:235–284, 2008. URL `https://jmlr.org/papers/v9/krause08a.html`.

Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scGen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019. doi: 10.1038/s41592-019-0494-8.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):MSB202211517, 2023. doi: 10.15252/msb.202211517.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550, 2014. doi: 10.1186/s13059-014-0550-8.

Stéphane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082.

Krithika Manohar, Bingni W. Brunton, J. Nathan Kutz, and Steven L. Brunton. Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns. *IEEE Control Systems*, 38(3):63–86, 2018. doi: 10.1109/mcs.2018.2810460.

Alsu Missarova, Jaison Jain, Andrew Butler, Shila Ghazanfar, Tim Stuart, Maigan Brusko, et al. geneBasis: an iterative approach for unsupervised selection of targeted gene panels from scRNA-seq. *Genome Biology*, 22(1):333, 2021. doi: 10.1186/s13059-021-02548-z.

Thomas M. Norman, Max A. Horlbeck, Joseph M. Replogle, Alex Y. Ge, Albert Xu, Marco Jost, et al. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019. doi: 10.1126/science.aax4438.

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005. doi: 10.1109/tpami.2005.159.

Joseph M. Replogle, Thomas M. Norman, Albert Xu, Jeffrey A. Hussmann, Jin Chen, J. Zachery Cogan, et al. Combinatorial single-cell CRISPR screens by direct guide RNA capture and targeted sequencing. *Nature Biotechnology*, 38(8):954–961, 2020. doi: 10.1038/s41587-020-0470-y.

Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexander Lenail, Alina Guna, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28, 2022. doi: 10.1016/j.cell.2022.05.013.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, 2024. doi: 10.1038/s41587-023-01905-6.

Daniel Schraivogel, Andreas R. Gschwind, Jennifer H. Milbank, Daniel R. Leonce, Petra Jakob, Lukas Mathur, et al. Targeted Perturb-seq enables genome-scale genetic screens in single cells. *Nature Methods*, 17(6):629–635, 2020. doi: 10.1038/s41592-020-0837-5.

Jordan W. Squair, Matthieu Gautier, Claudia Kathe, Mark A. Anderson, Nicholas D. James, Thomas H. Hutson, et al. Confronting false discoveries in single-cell differential expression. *Nature Communications*, 12(1):5692, 2021. doi: 10.1038/s41467-021-25960-2.

Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David D. Peck, Ted E. Natoli, Xiaodong Lu, et al. A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles. *Cell*, 171(6):1437–1452.e17, 2017. doi: 10.1016/j.cell.2017.10.049.

Maitreyi Swaroop, Tamar Krishnamurti, and Bryan Wilder. Distributionally robust feature selection. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 63046–63075. Curran Associates, Inc., 2025. doi: 10.52202/085713-2117. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5b3f001200198bb4ecaac0a1ea89dd99-Paper-Conference.pdf.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A. Teichmann, et al. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, 44(6):1050–1059, 2026. doi: 10.1038/s41587-025-02777-8.

Svante Wold, Michael Sjöström, and Lennart Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130, 2001. doi: 10.1016/s0169-7439(01)00155-1.

Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature Selection using Stochastic Gates. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 10648–10659, 2020. URL https://proceedings.mlr.press/v119/yamada20a.html.

Yifan Yang, Hao Ban, Minhui Huang, Shiqian Ma, and Kaiyi Ji. Tuning-free bilevel optimization: New algorithms and convergence analysis. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 75764–75814, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/bcaebb606df0c8c714756660e4a3bd7c-Paper-Conference.pdf.

Ronghui Zhu, Emma Dann, Jun Yan, Justine Reyes Retana, Ryunosuke Goto, Reese C. Guitche, et al. Genome-scale perturb-seq in primary human CD4+ T cells maps context-specific regulators of T cell programs and human immune traits. *Cell*, 2026. doi: 10.1016/j.cell.2026.08.002.

A Derivations and Numerical Verification

A.1 Entropy-constrained budget layer

For $0 < B < G$ and $\tau > 0$, the objective in Equation (2) is strictly concave on the feasible interior. Introducing multiplier ν for the budget yields $s_g + \tau \log((1 - w_g)/w_g) - \nu = 0$, hence $w_g = \sigma((s_g - \nu)/\tau)$. The sum of weights is continuous and strictly decreasing in ν, with limits G and zero, so the budget determines a unique threshold. Differentiating $\sum_g w_g = B$ gives

$$d\nu = \frac{\sum_g t_g \, ds_g}{\sum_g t_g}, \qquad dw_g = t_g(ds_g - d\nu), \qquad t_g = w_g(1 - w_g)/\tau.$$

This gives Equation (3). In particular, $J_B \mathbf{1} = 0$: a constant score shift leaves the design unchanged. The implementation finds ν by 48 bisection steps and uses the implicit derivative in the backward pass. This layer instantiates the projection of Amos et al. (2019) with a gene-level measurement budget.

A.2 Weighted ridge and its dual predictor

Concatenate the standardized support features in context-major order into $X \in \mathbb{R}^{n \times CG}$. Let W be diagonal with gene weights repeated once per context. Ridge fitting on the weighted features solves

$$\Theta(w) = \arg\min_{\Theta} \left\| Z_T - XW^{1/2}\Theta \right\|_F^2 + \alpha \|\Theta\|_F^2.$$

The normal equations give $\Theta = (W^{1/2}X^\top X W^{1/2} + \alpha I)^{-1} W^{1/2} X^\top Z_T$. Using the primal–dual identity and query features Q,

$$QW^{1/2}\Theta = QWX^\top (XWX^\top + \alpha I)^{-1} Z_T.$$

The grouped definitions of K and C in Equation (4) follow. The expression remains valid for zero weights because $\alpha > 0$. A hard indicator vector therefore gives exactly the common ridge model fitted on the selected genes.

A.3 Proof of the conditional measurement-utility derivative

Write $C_g = Q_g X_g^\top$ and $K_g = X_g X_g^\top$. Since $\partial A^{-1}/\partial w_g = -A^{-1} K_g A^{-1}$,

$$\begin{aligned} \frac{\partial F}{\partial w_g} &= C_g A^{-1} Z_T - C A^{-1} K_g A^{-1} Z_T \\ &= Q_g X_g^\top H - P X_g X_g^\top H \\ &= (Q_g - P X_g) X_g^\top H. \end{aligned}$$

For a differentiable outer loss, the chain rule gives the Frobenius inner product stated in Proposition 1. Multiplication by $J_B^\top$ gives the gradient with respect to scores. Automatic differentiation through the linear solve implements the same expression; the test suite checks it against the explicit grouped formula and finite differences on independent random matrices.

A.4 Finite changes and hard deployment

Let $A_v = A_w + \Delta K$ and $C_v = C_w + \Delta C$. The resolvent identity gives $A_v^{-1} - A_w^{-1} = -A_w^{-1} \Delta K A_v^{-1}$. Therefore

$$\begin{aligned} F(v) - F(w) &= \Delta C A_v^{-1} Z_T + C_w (A_v^{-1} - A_w^{-1}) Z_T \\ &= (\Delta C - P(w) \Delta K) A_v^{-1} Z_T. \end{aligned}$$

For nonnegative v, $A_v \succeq \alpha I$, which yields

$$\|F(v) - F(w)\|_F \le \frac{\|\Delta C - P(w)\Delta K\|_2 \, \|Z_T\|_F}{\alpha}.$$

This is a fixed-episode prediction identity and bound. It quantifies the effect of changing a design, including hard rounding, without assuming that gradient descent finds a globally optimal subset. The final model is refitted on the full construction atlas after the hard panel is selected.

A.5 TESTS

Six tests for the design module independently check the budget-layer Jacobian and shift invariance; equality of the dual predictor to a separately solved primal ridge model; the explicit gene derivative and finite differences; signed-loss direction and exclusion of the perturbed gene; the finite-change identity for soft-to-hard rounding; and save/load equivalence with panel-only inputs. Eight additional tests cover the underlying response model, feature-to-gene budget conversion, and inference validation. All 14 tests pass in the supplied environment.

B IMPLEMENTATION AND OPTIMIZATION

Algorithm. Given a construction atlas, standardize each source and the endpoint, replace the standardized on-target endpoint entry by zero, and compute a rank-64 randomized SVD. Initialize gene scores from log squared covariance with all 64 endpoint coordinates, standardize the log scores, multiply by 0.05, and add Gaussian noise with standard deviation 10^{-3}. At update t, choose budget B_t by cycling through 32, 64, and 128; draw 384 construction identities without replacement; use the first 256 for support and the remaining 128 for queries. Apply the budget layer, solve the support model, decode query responses, compute Equation (7), and update the scores. Center the scores after every update, which preserves the gate by shift invariance. After 360 updates, sort scores once and refit the common decoder at each requested hard budget.

Table 3: Implementation settings shared across tasks.

Setting	Value
Source contexts	CD4: 2; K562-to-RPE1: 1
Endpoint rank / randomized SVD	64 / 4 power iterations, seed 717
Standard-deviation floor	10^{-3}, sample standard deviation
Support / query size	256 / 128, disjoint identities
Budgets / updates	$\{32, 64, 128\}$ / 360
Optimizer / learning rate	Adam, default moment parameters / 0.04
Gate temperature	Geometric decay, $0.5 \to 0.05$
Ridge coefficient	$1.5\times$ number of fitting perturbations
Signed response weight / temperature	0.1 / square root of construction response variance
Primary seed / additional CD4 seeds	9 / 11, 17, 23, 29
Learning precision / final ridge solve	float32 / float64

Budget and candidate semantics. Each gene weight is shared across source contexts. The CD4 128-gene panel consequently uses 256 response values; the Replogle panel uses 128. Soft weights satisfy the budget numerically, while hard panels contain exactly B unique genes. All 10,282 CD4 genes or 7,226 Replogle genes are candidates in the full design. The same-pool control restricts CD4 learning to the 1,000 genes used by the PERSIST variance prefilter.

Computation. The support kernel requires $O(n^2CG)$ work, its regularized factorization requires $O(n^3)$ work, and query prediction adds $O(mnCG + mnr)$ work. Decoding the response loss costs $O(mrG)$. The support system has only $n = 256$ rows, while candidate groups are processed in dense matrix products. The implementation uses PyTorch's linear solver and automatic differentiation; no explicit matrix inverse is formed. The constant-size support solve removes the need to optimize a large neural decoder at every design step. Full-atlas standardization and endpoint decomposition remain necessary preprocessing costs.

Software and artifacts. Experiments use Python 3.10, NumPy 1.26.4, SciPy 1.11.4, scikit-learn 1.7.2, PyTorch 2.1.2, Matplotlib 3.8.4, and an NVIDIA A100 PCIe with 40 GB of memory. The `BilevelSENTRY` estimator implements training and serialization. The `predict_panel` interface requires source arrays in selected-gene order and returns a full endpoint profile. The supplementary model is sufficient for inference without downloading the construction atlas. Training data can be reconstructed from the public sources using the supplied scripts.

C Data Partitions and Development Choices

CD4 source. The author-released GWCD4 analysis supplies differential-response statistics for rest, eight-hour stimulation, and 48-hour stimulation. We use the released log-fold-change layer, retaining targets with one eligible row per condition and at least 20 cells in every condition. The source object contains 10,282 measured genes and has 16,786,240,107 bytes. The preprocessing code reads only required matrix rows through HTTP range requests and checks the metadata, object size, and gene axis. These are author-estimated aggregate responses; differential-expression methodology and replication structure affect their uncertainty (Love et al., 2014; Squair et al., 2021).

The original eligible identity partition uses SHA-256 with salt `P22-CD4-v1`. Its hash fractions below 0.70 contain 7,780 eligible identities, from which the pre-existing construction procedure assigns 6,144 identities to three construction shards and the remaining 1,636 to the earlier evaluation cohort. Fractions in $[0.70, 0.85)$ contain 1,606 identities that are not used here. All 1,699 eligible identities with hash fraction at least 0.85 form the primary confirmation cohort. This cohort is disjoint from both earlier groups and does not inherit the construction-shard sampling rule.

Development. A separate salted hash of construction identities defines 4,904 fitting and 1,240 validation examples. Every development standardizer and endpoint basis is fitted on the 4,904 examples. We evaluate signed-loss weights $\eta \in \{0, 0.1, 0.5\}$ and two additional settings that add standardized MSE with coefficients 0.25 and 0.5 to the $\eta = 0.1$ objective. The criterion is mean raw validation MSE over budgets 32, 64, and 128. The selected setting is $\eta = 0.1$ with no standardized-MSE term. All settings and outcomes are retained in the supplementary results.

Table 4: All five trained development configurations. Values average the three budget-specific raw MSEs on construction validation identities.

Signed-loss weight	Standardized-MSE weight	Mean validation MSE ↓
0	0	0.074425
0.1	0	**0.074383**
0.5	0	0.074642
0.1	0.25	0.074435
0.1	0.5	0.074453

The confirmation recipe records the selected setting, budgets, metrics, seeds, component controls, and source-file hashes before computing the confirmation metrics. The 64D PERSIST and stopped-inner-gradient comparisons are additional controls after the main run; all SENTRY-BD settings remain unchanged. The latter refits the same model on every forward pass but detaches $H = A^{-1}Z_T$ during backpropagation. Its MSE difference has a paired 95% interval spanning zero (-0.046–0.122%). The protocol record and development grid distinguish method selection from these controls.

Secondary CD4 cohort. On the previously examined 1,636-perturbation cohort, SENTRY-BD gives raw MSE 0.07617, compared with 0.07686 for 8D residual selection and 0.07750 for PERSIST. The relative improvement over PERSIST is 1.716% (95% CI: 1.421–2.062%). The direction agrees with the primary confirmation cohort. In this secondary cohort, removing the signed term slightly improves MSE, further supporting its interpretation as a secondary design choice.

Replogle preparation. We use the author-released K562 essential-gene and RPE1 pseudobulk files. Rows mapping to the same perturbed Ensembl gene are averaged using released filtered-cell counts. Targets need at least 30 cells in both cell lines. The output axis is the sorted intersection of 7,226 Ensembl genes. Identity hashing assigns five outer folds over 1,676 perturbations. Each fit uses only its four training folds for standardization, basis construction, panel selection, and decoder fitting. Cell counts determine eligibility and within-target aggregation, whereas the evaluation average assigns equal weight to perturbation identities.

D METRICS AND ADDITIONAL COMPARISONS

Let Ω_i omit the directly perturbed endpoint gene for perturbation i. Raw MSE is $|\Omega_i|^{-1} \sum_{g \in \Omega_i} (\widehat{y}_{ig} - y_{ig})^2$. Standardized MSE divides each residual by its construction endpoint standard deviation before squaring. Pearson correlation is computed across the retained gene coordinates; ρ_Δ first subtracts the construction endpoint mean from both prediction and truth. The same construction reference is used for every method within a split.

Signed top-k overlap ranks genes by absolute response and counts genes appearing in both top-k sets with the same response sign, divided by k. We report $k = 20$ in the main text and retain $k = 100$ in the per-target outputs. These are response-magnitude rankings rather than statistical differential-expression calls.

For centroid accuracy, compare each predicted response with every measured endpoint in its evaluation cohort or outer fold. Its score is the fraction of other endpoints farther away than its own, using squared Euclidean distance. The directly perturbed gene of the query is excluded from every candidate distance for that query. This uses the discrimination principle of Systema (Viñas Torné et al., 2026) with our aggregate-response data and split-specific reference.

Table 5: Additional 128-gene metrics. Standardized MSE is lower-is-better; Pearson correlation and centroid accuracy (CA) are higher-is-better.

	CD4 confirmation			K562 → RPE1		
Selector	Std. MSE	Pearson	CA	Std. MSE	Pearson	CA
Variance	0.9644	0.2663	0.5689	0.8455	0.4763	0.6487
PLS-VIP (8D)	0.9316	0.2725	0.5811	0.8379	0.4660	0.6520
PLS-VIP (64D)	0.9315	0.2727	0.5816	0.8382	0.4663	0.6520
PERSIST (8D)	0.9553	0.2695	0.5726	0.8402	0.4707	0.6507
PERSIST (64D)	0.9546	0.2725	0.5759	0.8423	0.4706	0.6480
Residual selection (8D)	0.9263	0.2830	0.5911	0.8310	0.4706	0.6658
Residual selection (64D)	0.9248	0.2860	0.5942	0.8267	0.4739	0.6692
SENTRY-BD	0.9267	0.2950	0.6000	0.8288	0.4763	0.6665

Bootstrap and seed variation. Each bootstrap sample resamples perturbation identities with replacement, retaining paired method errors. Relative reduction is computed from the two resampled mean MSEs. Replogle resampling pools the stored out-of-fold predictions; it does not refit fold models. The five CD4 seeds vary score initialization and support/query sampling. They retain the endpoint basis seed and construction cohort, and consequently measure the stochasticity of panel learning rather than donor or cohort variation.

Baseline adaptation. PERSIST uses its public implementation at commit `a16a496141f28e495493361a1a127bb340c6517f`. The source design is restricted to 1,000 high-variance genes; all contexts of those genes are inputs. The network has two 128-unit hidden layers and uses supervised MSE, batches of 128, Adam learning rate 10^{-3}, and 80 epochs with temperature $10 \to 0.01$. Seeds 11, 17, 23, 29, and 37 are compared by an inner 10% identity validation split. Selected context features map to unique genes, with duplicate genes filled by the learned feature scores. All selected genes are then measured in all source contexts and refitted with the common decoder. Both the 8D and 64D variants follow this rule.

The 8D residual selector acquires batches by residual association with QR filtering; the 64D residual control uses the common endpoint coordinates and omits that filter. The supplementary implementation retains its batch size, shortlist, and ridge settings, so the control changes only representation and QR use. PLS-VIP ranks genes using its supervised coordinate target at the indicated rank. Variance selection uses pooled raw source variance before standardization.