WHEN NOT TO CORRECT: EVIDENCE-BASED INTERACTION PREDICTION FOR COMBINATORIAL PERTURBATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting the transcriptomic effects of unseen combinatorial genetic perturbations from observed single-gene responses is a central challenge in functional genomics. A key difficulty is not whether genetic interactions exist, but whether interaction structure learned from observed pairs transfers reliably to new combinations. We introduce Hierarchical Null Regression (HNR), an evidence-based framework that adds interaction corrections only when supported by the training data. HNR separates residual interactions into a low-rank Core and full-rank Tail, with structure selected by permutation testing and predictions regularized through empirical Bayes estimation and uncertainty-aware shrinkage. When transferable structure is unsupported, HNR suppresses the interaction correction rather than forcing it. Across four Perturb-seq datasets, HNR consistently improves prediction of unseen combinations over strong linear, neural, graph-based, and pretrained baselines, while requiring no pretraining, external knowledge graph, validation-set hyperparameter tuning, or GPU computation. These results suggest that reliable combinatorial perturbation prediction depends not only on modeling interactions, but on determining when those interactions can be trusted. Code is provided as supplemental material for review and will be publicly released upon acceptance.

1 INTRODUCTION

High-throughput perturbation screens have become an important tool for studying how genetic interventions reshape cellular states at single-cell resolution (Dixit et al., 2016; Adamson et al., 2016; Norman et al., 2019; Replogle et al., 2022; Frangieh et al., 2021). These datasets are also driving efforts to build virtual cell models that can predict responses to perturbations that have not been directly measured (Bunne et al., 2024). An important next step is to extend such predictions to unseen perturbation combinations. Here, each perturbation may have been observed individually, while their joint effect remains unknown. Reliable predictions could reduce the number of combinations that need to be tested experimentally (Huang et al., 2024) and help agentic screening systems prioritize candidates for follow-up (Roohani et al., 2025; Hao et al., 2025; Qu et al., 2026). As illustrated in Figure 1, The challenge is that combined perturbations can produce interaction effects that are not evident from the individual responses alone.

A wide range of computational models have been developed for perturbation prediction. These include graph-based methods that incorporate biological prior knowledge (Roohani et al., 2024), compositional latent-variable models (Lotfollahi et al., 2023), and single-cell foundation models pretrained on large transcriptomic datasets (Cui et al., 2024; Hao et al., 2024; Theodoris et al., 2023). Despite this progress, recent benchmarks show that more complex models do not consistently outperform simple linear baselines on unseen perturbations and perturbation combinations (Ahlmann-Eltze et al., 2025; Bendidi et al., 2024; Wenteler et al., 2025; Kernfeld et al., 2025; Viñas Torné et al., 2026; Wei et al., 2026). This suggests that the main difficulty is not simply whether interaction effects are present, but whether they can be learned in a form that transfers to new combinations. In particular, a model may infer non-additive structure from observed pairs yet apply that structure unreliably when the combination changes. The problem is therefore one of compositional generalization: the individual perturbations are familiar, but their interaction must be extrapolated to a combination that was not observed during training (Lake & Baroni, 2018; Keysers et al., 2020).

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.

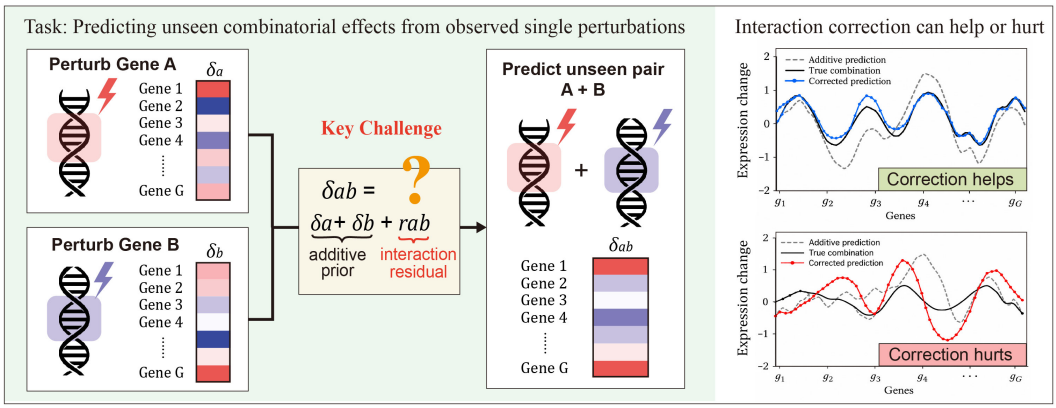

Figure 1: Overview of combinatorial perturbation prediction from observed single-perturbation effects. The unseen joint response is predicted relative to an additive reference, while the remaining interaction effect must be inferred from observed combinations.

This limitation suggests a different approach to combinatorial prediction. Rather than assuming that interaction structure learned from observed pairs will transfer to unseen ones, a predictor should first ask how strongly such corrections are supported by the training data. We therefore introduce **Hierarchical Null Regression (HNR)**, which uses a simple base prediction and adds interaction corrections only to the extent supported by the observed data. HNR separates shared low-rank interaction structure from the remaining residual variation, then predicts both components using similarities derived from observed single-perturbation effects. Uncertain interaction predictions are shrunk toward zero, limiting their influence when evidence for transfer is weak. In this way, the amount and complexity of interaction correction adapt to the dataset rather than being fixed in advance. These choices are determined from the training data without pretraining, external knowledge graphs, or validation-set hyperparameter tuning.

Our contributions are as follows:

- We introduce **HNR**, a framework that separates the **presence of interaction effects** from their **predictability on unseen combinations**. Rather than assuming that interactions learned from observed pairs will transfer, HNR controls the size and complexity of interaction corrections according to the evidence available in the training data.
- We evaluate HNR on **four combinatorial Perturb-seq datasets**: Norman, Wessels2023, DirectCapture (GSE146194), and GSE164996. For DirectCapture and GSE164996, we construct source-matched preprocessing protocols to provide additional benchmarks for compositional prediction. Across **24 dataset×metric evaluations**, HNR achieves the best score in **22 cases**. It requires no pretraining, external knowledge graph, or validation-set hyperparameter tuning.
- We show that HNR exhibits **adaptive complexity** across datasets without dataset-specific tuning. On Norman, Wessels2023, and GSE164996, it selects a Ridge base and detects low-rank interaction structure. On DirectCapture, it places greater weight on the nonlinear base and detects no low-rank interaction component. These choices are determined directly from the training data through closed-form estimation, permutation testing, and evidence maximization.

2 Related Work

Perturbation prediction and the linear baseline challenge. High-throughput perturbation screens have motivated a range of models for predicting cellular responses to unseen interventions (Dixit et al., 2016; Norman et al., 2019; Replogle et al., 2022). GEARS (Roohani et al., 2024) incorporates Gene Ontology and co-expression graphs to model combinatorial perturbations, while CPA (Lotfollahi et al., 2023) represents perturbation effects in a compositional latent space. More recently, pretrained single-cell models such as scGPT (Cui et al., 2024) have been adapted to perturbation prediction. Despite these advances, several benchmarks have found that complex models do not consistently

outperform simple linear baselines on unseen perturbations and perturbation combinations (Ahlmann-Eltze et al., 2025; Bendidi et al., 2024; Wenteler et al., 2025; Kernfeld et al., 2025; Viñas Torné et al., 2026; Wei et al., 2026). These findings suggest that increasing model capacity alone does not resolve the challenge of generalizing interaction effects to unseen combinations.

Genetic interactions and compositional generalization. Genetic interactions are commonly defined as deviations from an additive expectation and have long been used to characterize epistatic relationships between genes (Fisher, 1918; Mani et al., 2008; Phillips, 2008). Large-scale genetic interaction studies have shown that these effects often exhibit structured patterns across genes and pathways rather than varying independently from pair to pair (Boone et al., 2007; Costanzo et al., 2016; Norman et al., 2019). In combinatorial perturbation prediction, the test case is often a new pair whose two constituent perturbations have each been observed individually. The difficulty is therefore not predicting entirely unseen components, but predicting how known components behave when combined in a new configuration. This setting is closely related to compositional generalization, where models must recombine familiar elements in combinations not observed during training (Lake & Baroni, 2018; Keysers et al., 2020; Hupkes et al., 2020). For such pairs, the additive contribution is supported by observed single-perturbation responses, whereas the interaction effect must be inferred from patterns observed in other combinations.

3 METHOD

We consider the problem of predicting the transcriptomic effect of a combinatorial genetic perturbation from observed single-perturbation responses. Let $\delta_a, \delta_b \in \mathbb{R}^G$ denote the measured mean expression changes for single perturbations a and b relative to unperturbed controls, where G is the number of genes. The combinatorial effect $\delta_{ab} \in \mathbb{R}^G$ can be decomposed as

$$\delta_{ab} = \delta_a + \delta_b + r_{ab}, \tag{1}$$

where $r_{ab} = \delta_{ab} - \delta_a - \delta_b$ is the *interaction residual*. Given a training set $\mathcal{T}$ of observed perturbation pairs together with the corresponding single-perturbation effects, our goal is to predict δ_{ab} for an unseen pair $(a, b) \notin \mathcal{T}$, as illustrated in Figure 2.

The central challenge is that observed interaction residuals contain a mixture of transferable structure and pair-specific noise. Existing models parameterize interaction effects densely, permitting a nonzero correction for every pair without testing whether it is predictively supported. We observe empirically that such corrections often *increase* error on unseen pairs relative to the additive baseline $\delta_a + \delta_b$. HNR addresses this by treating additivity as the null model and allowing interaction corrections only when supported by statistical evidence.

3.1 BASE PREDICTOR

HNR constructs its base prediction by blending two complementary estimators.

Nonlinear single-effect calibration. The additive baseline $s_{ab} = \delta_a + \delta_b$ provides a natural first approximation to the joint response, but systematic nonlinear deviations can remain even when the two single-perturbation effects are known. To capture these gene-specific departures while keeping the correction simple, we fit a per-gene nonlinear calibration of the form

$$f(s)^{(g)} = s^{(g)} + \beta_1^{(g)} s^{(g)} + \beta_2^{(g)} (s^{(g)})^2, \tag{2}$$

where coefficients $\beta_1, \beta_2 \in \mathbb{R}^G$ are estimated by empirical Bayes evidence maximization (type-II maximum likelihood) on the concatenation of training pair effects and all single effects. This yields the nonlinear predictor $\hat{\delta}_{ab}^{\text{nl}} = f(s_{ab})$.

Global linear model. A Ridge regression provides a complementary linear estimate of the joint response. It maps one-hot perturbation indicators to expression effects and is jointly fitted on all singles and training pairs, yielding $\hat{\delta}_{ab}^{,\text{lin}}$.

Adaptive blending. The base prediction is a convex combination

$$\hat{\delta}_{ab}^{\text{base}} = \alpha\, \hat{\delta}_{ab}^{\text{nl}} + (1 - \alpha)\, \hat{\delta}_{ab}^{\text{lin}}, \tag{3}$$

where $\alpha \in [0, 1]$ is determined in closed form by projecting training residuals onto the direction $\hat{\delta}^{\text{nl}} - \hat{\delta}^{\text{lin}}$ (see Appendix A.1). No validation set is used; α is computed entirely from training pairs.

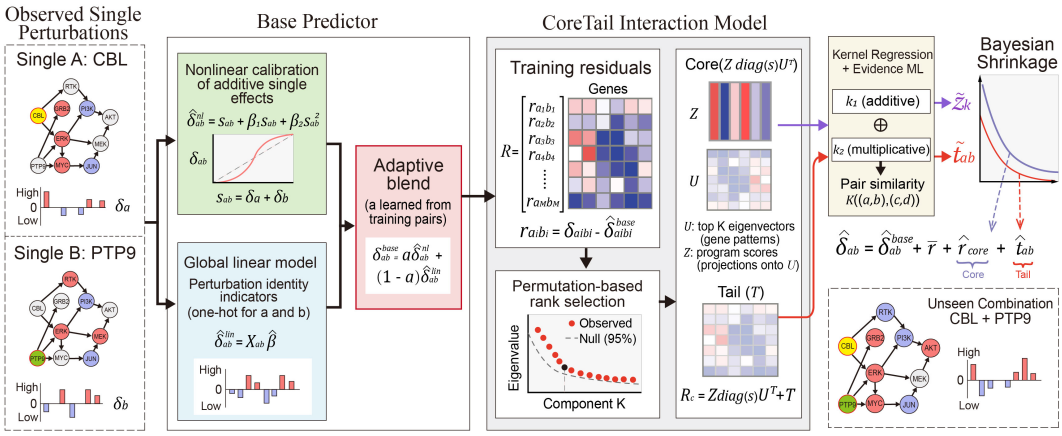

Figure 2: The overall framework of HNR for predicting unseen combinatorial perturbations from observed single effects and learned interaction structure across training pairs.

3.2 Interaction Modeling via CoreTail

Given the base predictor, let $Y_{\mathcal{T}} \in \mathbb{R}^{M\times G}$ denote the matrix whose rows are the observed combinatorial effects $\{\delta_{ab}\}_{(a,b)\in\mathcal{T}}$, and $\hat{Y}^{\text{base}}_{\mathcal{T}}$ the corresponding base predictions. We define the training interaction residual matrix

$$R = Y_{\mathcal{T}} - \hat{Y}^{\text{base}}_{\mathcal{T}} \in \mathbb{R}^{M\times G}, \tag{4}$$

where $M = |\mathcal{T}|$ is the number of training pairs. CoreTail decomposes R into a low-rank *core* capturing shared interaction programs and a full-rank *tail* capturing residual pair-specific effects.

Interaction rank selection via permutation testing. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ be the eigenvalues of $R_c R_c^\top$, where $R_c = R - \bar{r}$ is mean-centered. To determine which eigenvalues reflect genuine shared structure rather than noise, we compare each λ_k against a null distribution obtained by independently permuting gene values within each column of R_c. Specifically, for B permutations we compute null eigenvalues $\{\lambda_k^{(b)}\}_{b=1}^{B}$ and retain only directions where λ_k exceeds the 95th percentile of the null:

$$K = \max\{k : \lambda_j > \lambda_j^{(.95)},\ \forall\, j \leq k\}. \tag{5}$$

This yields K interaction programs $U \in \mathbb{R}^{G\times K}$ (the corresponding eigenvectors projected into gene space), normalized program scores $Z \in \mathbb{R}^{M\times K}$ for training pairs, and associated program scale factors $s_1, \ldots, s_K$, such that the reconstructed Core is $Z\,\mathrm{diag}(s)\,U^\top$. If $K = 0$, no low-rank Core correction is applied; the Tail component may still contribute subject to uncertainty-aware shrinkage.

Kernel regression with evidence maximization. To predict program scores for unseen pairs, we construct pair-level kernels from single-perturbation effects. Let $E \in \mathbb{R}^{N\times G}$ be the matrix of centered single effects for N perturbation genes. For pairs (a, b) and (c, d) with gene indices i_a, i_b, i_c, i_d, we define two kernel components:

$$k_1\big((a,b),(c,d)\big) = \tfrac{1}{2}(S_{i_a i_c} + S_{i_a i_d} + S_{i_b i_c} + S_{i_b i_d}), \tag{6}$$

$$k_2\big((a,b),(c,d)\big) = \tfrac{1}{2}(S_{i_a i_c} S_{i_b i_d} + S_{i_a i_d} S_{i_b i_c}), \tag{7}$$

where $S = E_c E_c^\top / \operatorname{tr}(E_c E_c^\top) \cdot N$ is the normalized single-effect Gram matrix. The additive kernel k_1 captures similarity in constituent genes; the multiplicative kernel k_2 captures combinatorial structure. For each program $k = 1, \ldots, K$, the kernel weights θ_k and noise variance σ_k^2 are learned by maximizing the log marginal likelihood:

$$\hat{\theta}_k, \hat{\sigma}_k^2 = \arg\max_{\theta,\sigma^2} \log p(z_k \mid X_{\mathcal{T}}, \theta, \sigma^2), \tag{8}$$

where $z_k \in \mathbb{R}^M$ is the vector of training scores for program k and the likelihood is Gaussian with covariance $\sum_j \theta_j K_j + \sigma^2 I$ (Appendix A.2). The tail residual $R_c - Z\,\mathrm{diag}(s)\,U^\top$ is modeled analogously with a single joint regression.

Uncertainty-aware shrinkage. For a test pair (a^*, b^*), the kernel regression produces predicted program scores $\tilde{z}_k$ and variances v_k. These are combined into an unshrunk gene-space core reconstruction $\tilde{r}_{\text{core}} = \sum_k \tilde{z}_k \, s_k \, u_k$ and an aggregate gene-space predictive variance v_{core} obtained by propagating the score-space variances through the program directions (Appendix A.3). We apply a single Bayesian shrinkage weight in gene space:

$$\hat{r}_{\text{core}} = \frac{\tau^2_{\text{core}}}{\tau^2_{\text{core}} + v_{\text{core}}} \, \tilde{r}_{\text{core}}, \tag{9}$$

where τ^2_{core} is the empirical signal variance of the core reconstruction on training pairs. When predictive uncertainty dominates ($v_{\text{core}} \gg \tau^2_{\text{core}}$), the shrinkage factor approaches zero and the interaction correction vanishes. The tail prediction is shrunk more aggressively, with the shrinkage weight squared (Appendix A.4). The squared shrinkage for the tail is a fixed architectural choice, not a tuned hyperparameter: the tail captures unstructured residual variation with lower signal-to-noise ratio than the core programs, and squaring the weight enforces a more conservative default that suppresses tail corrections unless predictive confidence is high.

3.3 Final Prediction and Adaptive Complexity

The complete HNR prediction combines the adaptive base predictor with the mean training residual, the shrunk core correction, and the shrunk tail correction:

$$\hat{\delta}_{ab} = \hat{\delta}^{\text{base}}_{ab} + \bar{r} + \hat{r}_{\text{core}} + \hat{t}_{ab}, \tag{10}$$

where $\bar{r}$ is the mean training residual, $\hat{r}_{\text{core}}$ is the shrunk gene-space core interaction (Equation 9), and $\hat{t}_{ab}$ is the shrunk tail prediction. HNR exhibits adaptive complexity without any dataset-specific tuning. When no transferable interaction structure is detected ($K = 0$) and all tail predictions are shrunk to zero, Equation 10 reduces to $\hat{\delta}^{\text{base}}_{ab} + \bar{r}$. When the base blending further selects $\alpha = 0$, the method reduces to Ridge regression plus a constant offset. The blending weight α, interaction rank K, kernel weights θ_k, and shrinkage factors are all determined from the training data using closed-form estimation or evidence maximization.

4 Experimental Setup

Datasets. We evaluate HNR on four combinatorial perturbation datasets: **Norman** (Norman et al., 2019) (CRISPRa, 128 eligible doubles), **Wessels2023** (Wessels et al., 2023) (Cas13-based CaRPool-seq, 158 doubles), **DirectCapture** (Replogle et al., 2020) (GSE146194, 37 doubles), and **GSE164996** (Zhao et al., 2021) (matched two-guide design, 54 doubles). Eligible double perturbations are split 60%/20%/20% into training, validation, and test sets, while all measured single-perturbation effects remain observed. Results are reported as mean $\pm$ std over 4 seeds. Dataset construction and preprocessing are described in Appendix B.1.

Metrics. All metrics are scaled by $\times 100$, with lower values indicating better performance.

- **MAE**: $\frac{1}{G}\sum_{g=1}^{G} |\hat{\delta}ab^{(g)} - \delta ab^{(g)}|$, the mean absolute error across all G genes.
- **Top-k** ($k \in 10, 20$): MAE evaluated on the k genes with the largest absolute observed effects $|\delta^{(g)}_{ab}|$, measuring prediction accuracy on the strongest expression responses.
- **NonAdd-k** ($k \in 10, 20$): MAE evaluated on the k genes with the largest absolute interaction residuals $|r^{(g)}_{ab}|$, where $r^{(g)}_{ab} = \delta^{(g)}_{ab} - \delta^{(g)}_{a} - \delta^{(g)}_{b}$, measuring accuracy on the most strongly non-additive genes.
- **IntWMAE**: $\sum_g w_g |\hat{\delta}ab^{(g)} - \delta ab^{(g)}| / \sum_g w_g$, where $w_g = |r^{(g)}_{ab}|$, assigning greater weight to genes with stronger non-additive effects.

Implementation details. HNR requires no pretraining, external knowledge graph, or validation-set hyperparameter tuning. The blending weight α is computed in closed form from training pairs, the interaction rank K is selected by permutation testing at the 95th percentile, and kernel weights and noise variance are estimated by maximizing the log marginal likelihood with L-BFGS-B. Shrinkage factors are then computed from the estimated signal and predictive variances. The same procedure is used across all four datasets. Further implementation details are provided in Appendix B.4.

Baselines. We compare HNR against eight baselines: Additive ($\hat{\delta}_{ab} = \delta_a + \delta_b$), Ridge (Hoerl & Kennard, 1970), MLP, Transformer (Vaswani et al., 2017), CPA (Lotfollahi et al., 2023), scGPT (Cui et al., 2024), GEARS (Roohani et al., 2024), and GEARS+scGPT. Baseline configurations follow their corresponding implementations, with neural models using the validation partition for model selection and early stopping where applicable. Related details are provided in Appendix B.3.

5 RESULTS

Table 1: Comparison of HNR with eight baselines across four combinatorial perturbation datasets. **Bold** and underline denote the best and second-best results, respectively. † indicates performance within one standard error of the best baseline. GEARS and GEARS+scGPT are evaluated only on conditions whose genes are present in the GO graph.

Method	MAE↓	Top10↓	Top20↓	NA10↓	NA20↓	IntWMAE↓
Norman (128 double perturbations)						
Additive	1.40±.12	24.77±2.52	20.84±2.49	40.06±3.11	33.71±2.70	11.35±0.87
Ridge	1.19±.06	29.67±0.92	24.04±1.12	24.03±1.44	19.85±1.07	7.58±0.23
MLP	2.11±.12	46.38±3.86	36.74±2.68	29.82±2.44	24.47±1.42	9.31±0.32
Transformer	2.10±.19	75.40±4.16	59.46±4.65	46.35±4.90	36.77±4.17	13.15±1.19
CPA	1.69±.13	40.15±2.72	32.21±2.25	27.05±1.46	22.63±1.31	8.67±0.37
scGPT	2.09±.20	76.00±4.15	60.02±4.56	46.55±5.00	36.98±4.27	13.18±1.25
GEARS+scGPT	3.27±.42	35.16±2.18	29.09±1.57	25.30±0.83	21.58±0.41	10.75±0.49
GEARS	3.34±.40	34.17±1.43	28.36±0.62	24.68±0.86	21.16±0.46	10.70±0.56
HNR (ours)	**1.11**±.06	25.53†±1.03	**20.57**±1.17	**21.55**±1.21	**17.93**±0.74	**6.94**±0.16
Wessels2023 (158 double perturbations)						
Additive	3.34±.09	23.66±2.12	20.63±1.18	50.65±2.12	45.18±1.70	11.56±0.31
Ridge	1.61±.07	15.32±1.18	13.24±0.99	15.27±0.42	13.06±0.33	4.14±0.17
MLP	1.87±.07	28.13±1.23	23.05±0.65	16.22±1.29	13.65±0.84	4.28±0.14
Transformer	2.05±.13	47.75±4.14	38.39±2.61	23.10±2.06	18.65±0.74	5.16±0.18
CPA	1.67±.08	21.58±1.78	18.18±1.33	15.52±0.72	13.15±0.33	4.13±0.18
scGPT	2.08±.09	49.25±3.92	39.62±2.35	23.24±2.14	18.75±0.78	5.19±0.19
GEARS+scGPT	2.75±.24	29.29±2.83	25.25±2.37	18.00±1.50	15.29±1.32	5.81±0.35
GEARS	2.72±.50	26.92±5.04	23.31±4.12	17.10±0.79	14.42±0.87	5.69±0.80
HNR (ours)	**1.60**±.07	**12.98**±0.93	**11.46**±0.94	**14.41**±0.38	**12.57**±0.26	**4.10**±0.17
DirectCapture (37 double perturbations)						
Additive	50.92±7.37	4765±887	2393±444	6622±893	3334±449	20643±3420
Ridge	77.49±7.86	6881±1587	3451±796	7493±1639	3769±823	24515±3837
MLP	110.89±20.41	14516±3639	7269±1820	14675±3668	7353±1831	53243±9369
Transformer	113.04±20.77	14507±3644	7264±1823	14637±3651	7334±1823	53198±9310
CPA	116.07±22.50	14146±3612	7084±1805	14325±3626	7176±1811	52225±9245
scGPT	112.94±19.75	14805±3661	7414±1832	14871±3668	7451±1831	53901±9332
GEARS+scGPT	115.24±20.48	15059±3697	7557±1853	15071±3695	7547±1847	54456±9362
GEARS	114.96±20.37	15058±3697	7556±1852	15072±3695	7547±1847	54456±9361
HNR (ours)	**42.89**±8.29	**3666**±1421	**1841**±713	**4336**±1286	**2184**±653	**11370**±4233
GSE164996 (54 double perturbations)						
Additive	1.94±.03	86.22±4.80	53.88±2.86	116.47±3.74	82.08±2.25	15.26±0.21
Ridge	**1.14**±.03	27.99±1.86	21.91±1.52	22.22±2.04	19.13±1.60	5.55±0.15
MLP	1.96±.15	70.52±4.95	46.10±3.97	34.48±0.82	27.17±1.61	6.15±0.21
Transformer	1.26±.08	86.51±3.52	55.94±3.06	39.41±1.03	29.97±1.25	6.09±0.29
CPA	1.52±.05	67.72±4.42	44.52±3.89	33.63±0.43	26.42±0.95	5.93±0.16
scGPT	1.51±.11	89.22±4.57	57.35±3.70	41.58±1.78	31.02±1.33	6.24±0.29
GEARS+scGPT	1.74±.08	83.60±3.56	56.57±2.06	47.88±3.21	32.79±0.90	7.89±0.06
GEARS	1.76±.14	84.93±5.49	57.79±3.94	46.86±2.87	32.18±2.11	7.89±0.41
HNR (ours)	1.14†±.03	**23.69**±2.08	**19.79**±1.45	**18.92**±1.92	**17.58**±1.52	**5.40**±0.14

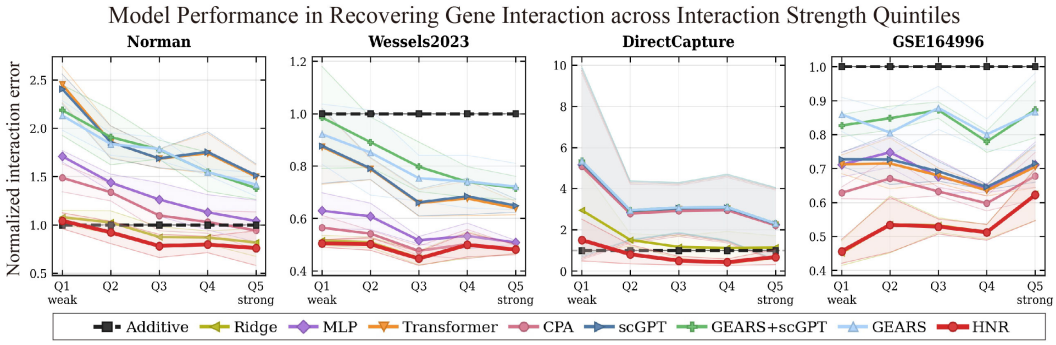

Figure 3: Interaction recovery across perturbation pairs grouped by true interaction-strength quintile. The x-axis ranges from weak (Q1) to strong (Q5) interactions, while the y-axis shows normalized interaction error. The dashed line marks the additive reference of predicting zero interaction.

Overall Results. Table 1 summarizes predictive performance across the four datasets, with HNR achieving the best result on 22 of 24 dataset×metric evaluations. A consistent pattern is that HNR preserves or improves overall prediction accuracy while providing larger gains on interaction-sensitive metrics, indicating that its advantage is concentrated on recovering non-additive effects rather than merely reducing average error. On Norman, HNR improves over Ridge on both MAE (1.11 vs. 1.19) and NA20 (17.93 vs. 19.85), with a corresponding improvement in IntWMAE. On Wessels2023, the two methods are nearly identical on MAE (1.60 vs. 1.61), but HNR performs better on the interaction-sensitive metrics, suggesting that the interaction module adds information beyond the strong Ridge baseline. DirectCapture represents a different regime: HNR selects zero low-rank interaction components, yet still achieves the lowest MAE (42.89 vs. 50.92 for Additive), showing that the framework does not depend on forcing a low-rank interaction structure when it is unsupported by the data. On GSE164996, HNR again matches Ridge on overall MAE while reducing NA20 from 19.13 to 17.58. Figure 3 further shows that these aggregate improvements extend across interaction-strength quintiles rather than being driven by a narrow subset of perturbation pairs. Statistical significance tests are provided in Appendix C.1.

Table 2: Ablation of HNR components. All values ×100 (↓); averaged over 4 seeds. R = adaptive Ridge blend, C = Core low-rank interaction programs, T = Tail residual interaction, and S = Bayesian shrinkage. IntW = IntWMAE. HNR_g uses only the nonlinear calibration base, while HNR_{ct} additionally incorporates the CoreTail interaction module without the Ridge blend.

					Norman		Wessels2023		DirectCapture		GSE164996	
Config	R	C	T	S	NA20	IntW	NA20	IntW	NA20	IntW	NA20	IntW
HNR_g	×	×	×	×	28.44	9.75	23.06	5.79	2828	18670	34.36	7.12
HNR_{ct}	×	✓	✓	✓	22.48	8.15	13.63	4.37	2793	18573	23.24	6.11
main (+ Ridge blend)	✓	×	×	×	19.85	7.58	13.06	4.14	2210	11441	19.13	5.55
w/o Core	✓	×	✓	✓	19.22	7.38	13.30	4.14	2184	11370	17.33	5.38
w/o Tail	✓	✓	×	✓	17.94	6.95	12.57	4.10	2184	11370	17.52	5.40
w/o Shrinkage	✓	✓	✓	×	18.78	7.25	12.38	4.11	2184	11370	17.74	5.44
Full (HNR)	✓	✓	✓	✓	**17.93**	**6.94**	**12.57**	**4.10**	**2184**	**11370**	**17.58**	**5.41**

Module Ablation Analysis. Table 2 isolates the contribution of each HNR component. The adaptive Ridge blend accounts for the largest single gain, reducing NA20 by 30–44% relative across datasets. On Norman, the Core low-rank programs capture most of the remaining interaction signal: removing Core raises NA20 from 17.93 to 19.22, while removing Tail changes it only to 17.94. Shrinkage provides a modest additional correction, with its effect varying across datasets. DirectCapture serves as a limiting case: its estimated interaction rank is zero, so the Core, Tail, and shrinkage ablations collapse to identical predictions.

Table 3: Computational cost and predictive performance on Norman (128 eligible pairs, 8,564 genes). † indicates methods requiring a GO gene interaction graph; ‡ indicates methods requiring scGPT pretrained weights.

	Computational requirements					Performance	
Method	Params	GPU	Graph	Pretrain	Time	MAE↓	IntWMAE↓
Additive	0	×	×	×	<1s	1.40±.12	11.35±.87
Ridge	closed-form	×	×	×	<1s	1.19±.06	7.58±.23
MLP	2.29M	✓	×	×	∼10s	2.11±.12	9.31±.32
Transformer	2.39M	✓	×	×	∼9s	2.10±.19	13.15±1.19
CPA	6.87M	✓	×	×	∼21s	1.69±.13	8.67±.37
scGPT‡	2.78M	✓	×	✓	∼14s	2.09±.20	13.18±1.25
GEARS†	2.31M	✓	✓	×	∼14min	3.34±.40	10.70±.56
GEARS+scGPT†‡	2.31M	✓	✓	✓	∼14min	3.27±.42	10.75±.49
HNR (ours)	0 (closed-form)	×	×	×	**3.8s**	**1.11**±.06	**6.94**±.16

Computational Efficiency. Table 3 shows that HNR achieves its predictive gains with substantially lower computational requirements than the neural baselines. On Norman, HNR runs in 3.8 s on CPU with no gradient-trained parameters, compared with 2.29–6.87M parameters for the neural models. Despite this lightweight design, HNR reduces MAE from 1.19 to 1.11 and IntWMAE from 7.58 to 6.94 relative to Ridge, corresponding to improvements of 6.7% and 8.4%, respectively. Compared with CPA, the strongest neural baseline on these two metrics, HNR lowers MAE by 34.3% and IntWMAE by 20.0%. The neural baselines require GPU execution and take approximately 9–21 s, while GEARS-based methods additionally rely on a GO interaction graph and require roughly 14 min, over 200× the runtime of HNR. Thus, the performance gains of HNR do not arise from greater model capacity or additional external information, but from selectively introducing interaction structure when supported by the data.

Table 4: Automatically selected HNR structure and sensitivity to the permutation threshold. α is the base blending weight (0 = Ridge, 1 = nonlinear). Core and Tail report fractions of interaction variance. Rank K is shown across permutation thresholds; downstream MAE is reported at the 90th and 99th percentiles. These counts reflect unique test pairs pooled across all four seeds.

				Variance split		Interaction rank K					MAE↓	
Dataset	n_{train}	n_{test}	α	Core %	Tail %	90th	92.5th	**95th**	97.5th	99th	90th	99th
Norman	76	26	0.00	64.6	35.4	7.0	7.0	7.0	6.5	6.5	1.11	1.11
Wessels2023	94	32	0.00	63.3	36.7	26.0	26.0	26.0	25.8	25.5	1.58	1.58
DirectCapture	23	7	0.70	0.0	100.0	0.5	0.2	0.0	0.0	0.0	0.40	0.40
GSE164996	32	11	0.00	69.6	30.4	10.2	10.2	10.2	10.0	9.8	1.19	1.19

Adaptive Structure Selection and Threshold Robustness. Table 4 shows that HNR automatically adapts both its base predictor and interaction decomposition to the structure of each dataset. Norman, Wessels2023, and GSE164996 all select a Ridge-only base predictor ($\alpha = 0$) while retaining a nonzero low-rank Core, which consistently explains about 63–70% of the interaction variance. However, the selected Core rank differs substantially across datasets, ranging from 7 on Norman to 26 on Wessels2023 at the default 95th-percentile threshold, indicating that HNR adjusts the complexity of the interaction representation rather than imposing a fixed rank. DirectCapture exhibits a qualitatively different regime: it assigns substantial weight to the nonlinear branch ($\alpha = 0.70$) and selects no Core component at the default threshold, so its interaction correction is captured entirely by the Tail. Importantly, these structural choices are stable with respect to the permutation threshold. Increasing the threshold from the 90th to the 99th percentile changes the selected rank by at most 0.5 on the three datasets with a nonzero Core, while downstream MAE remains unchanged at the two extremes for all four datasets. This indicates that HNR adapts its internal structure to dataset-specific interaction patterns without requiring a finely tuned threshold for reliable prediction.

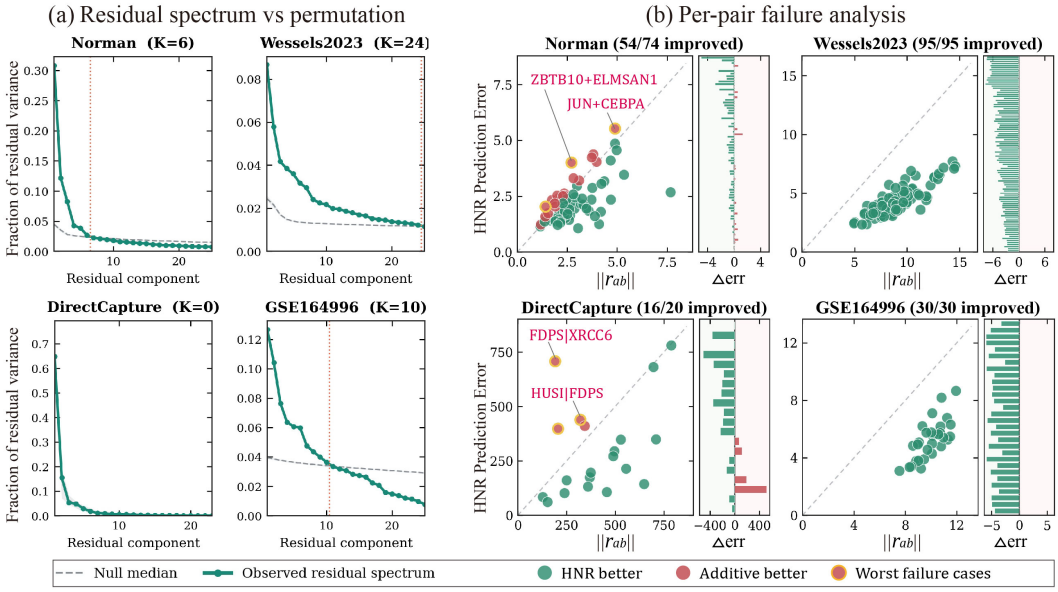

Figure 4: **Residual structure and failure analysis. (a)** Residual eigenspectra across four datasets. The x-axis is the residual component index and the y-axis is the fraction of residual variance; dashed curves show the permutation null and dotted lines mark the selected rank K. **(b)** Per-pair HNR errors. The x-axis shows interaction magnitude $\|r_{ab}\|$ and the y-axis shows HNR prediction error. Adjacent bars show $\Delta\text{err} = \text{error}_{\text{HNR}} - \text{error}_{\text{Additive}}$.

Interaction Recovery and Pair-Level Failure Analysis. Figure 4 links the recovered interaction structure to pair-level predictive behavior. In Fig. 4(a), Norman, Wessels2023, and GSE164996 retain residual spectral components above the permutation null, supporting the nonzero low-rank Core selected by HNR, whereas DirectCapture shows little persistent low-rank structure and selects $K = 0$. The pair-level analysis in Fig. 4(b) shows that the gains are broadly distributed rather than driven by a few favorable cases: HNR improves all 95 pairs on Wessels2023 and all 30 pairs on GSE164996, as well as 54/74 pairs on Norman and 16/20 on DirectCapture. The remaining errors are concentrated in a small number of pair-specific failures. On Norman, ZBTB10+ELMSAN1 and JUN+CEBPA are among the largest failures relative to the additive reference, while DirectCapture exhibits more pronounced outliers, including FDPS|XRCC6 and HUS1|FDPS. This contrast is consistent with the structural analysis: datasets with a retained low-rank Core show more uniformly transferable interaction structure, whereas the $K = 0$ DirectCapture regime remains more vulnerable to isolated pair-specific failures.

6 Conclusion

We introduced Hierarchical Null Regression (HNR), an evidence-based framework for predicting unseen combinatorial perturbations. Rather than assuming that interaction effects learned from observed pairs are transferable, HNR treats additivity as the null model and introduces interaction corrections only when supported by the training data. Across four Perturb-seq datasets, HNR achieved the best result on 22 of 24 dataset$\times$metric evaluations while requiring no pretraining, external knowledge graph, validation-set hyperparameter tuning, or GPU computational requirement. Its automatically selected structure also varied substantially across datasets, from nonzero low-rank interaction programs to a zero-rank regime when transferable structure was not detected. These results suggest that reliable combinatorial prediction depends not only on modeling interactions, but on determining when and how strongly those interactions should be trusted. More broadly, HNR provides a simple example of how statistical evidence and uncertainty can be used to control model complexity in compositional biological prediction.

AI USE STATEMENT

Generative AI tools were used during manuscript preparation for language editing, LaTeX formatting, and assistance with code review and debugging. All AI-assisted text and code were reviewed by the authors, and all reported experimental results were independently verified against the executed code and saved outputs. Generative AI was not used to generate experimental data or to determine reported numerical results. The authors take full responsibility for the final content of this work, including all text, claims, analyses, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses previously published and publicly available genetic perturbation datasets and does not involve new human-subject recruitment, animal experiments, or biological sample collection. Ethical approvals, consent procedures, and data-release practices associated with the original datasets are described in their respective source publications. This study introduces a computational prediction framework and does not generate new biological interventions or experimental samples.

REPRODUCIBILITY STATEMENT

We provide the methodological derivations and estimation procedures in Appendix A, detailed dataset construction, preprocessing, evaluation metrics, baseline configurations, and implementation details in Appendix B, and additional statistical and robustness analyses in Appendix C. All experiments use fixed random seeds and are reported over four train/validation/test splits. HNR does not use the validation partition for model selection; its blending weight, interaction rank, kernel parameters, and shrinkage factors are estimated from the training data. The datasets used in this study are publicly available through their original publications and accession records. Code and scripts required to reproduce the experiments are provided in the anonymous supplementary material.

REFERENCES

Britt Adamson, Thomas M. Norman, Marco Jost, Min Y. Cho, James K. Nuñez, Yuwen Chen, Jacqueline E. Villalta, Luke A. Gilbert, Max A. Horlbeck, Marco Y. Hein, Ryan A. Pak, Andrew N. Gray, Carol A. Gross, Atray Dixit, Oren Parnas, Aviv Regev, and Jonathan S. Weissman. 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.

Ihab Bendidi, Shawn Whitfield, Kian Kenyon-Dean, Hanene Ben Yedder, Yassir El Mesbahi, Emmanuel Noutahi, and Alisandra K. Denton. Benchmarking transcriptomics foundation models for perturbation analysis: One PCA still rules them all. *arXiv preprint arXiv:2410.13956*, 2024.

Charles Boone, Howard Bussey, and Brenda J. Andrews. Exploring genetic interactions and networks with yeast. *Nature Reviews Genetics*, 8(6):437–449, 2007. doi: 10.1038/nrg2085.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B. Burkhardt, Andrea Califano, Jonah Cool, Abby F. Dernburg, Kirsty Ewing, Emily B. Fox, Matthias Haury, Amy E. Herr, Eric Horvitz, Patrick D. Hsu, Viren Jain, Gregory R. Johnson, Thomas Kalil, David R. Kelley, Shana O. Kelley, Anna Kreshuk, Tim Mitchison, Stephani Otte, Jay Shendure, Nicholas J. Sofroniew, Fabian Theis, Christina V. Theodoris, Srigokul Upadhyayula, Marc Valer, Bo Wang, Eric Xing, Serena Yeung-Levy, Marinka Zitnik, Theofanis Karaletsos, Aviv Regev, Emma Lundberg, Jure Leskovec, and Stephen R. Quake. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024. doi: 10.1016/j.cell.2024.11.015.

Michael Costanzo, Benjamin VanderSluis, Elizabeth N. Koch, Anastasia Baryshnikova, Carles Pons, Guihong Tan, Wen Wang, Matej Usaj, Julia Hanchard, Susan D. Lee, Vicent Pelechano, Erin B. Styles, Maximilian Billmann, Jolanda van Leeuwen, Nydia van Dyk, Zhen-Yuan Lin, Elena Kuzmin, Justin Nelson, Jeff S. Piotrowski, Tharan Srikumar, Sondra Bahr, Yiqun Chen, Raamesh Deshpande, Christoph F. Kurat, Sheena C. Li, Zhijian Li, Mojca Mattiazzi Usaj, Hiroki Okada, Natasha Pascoe, Bryan-Joseph San Luis, Sara Sharifpoor, Emira Shuteriqi, Scott W. Simpkins, Jamie Snider, Harsha Garadi Suresh, Yizhao Tan, Hongwei Zhu, Noel Malod-Dognin, Vuk Janjic, Natasa Przulj, Olga G. Troyanskaya, Igor Stagljar, Tian Xia, Yoshikazu Ohya, Anne-Claude Gingras, Brian Raught, Michael Boutros, Lars M. Steinmetz, Claire L. Moore, Adam P. Rosebrock, Amy A. Caudy, Chad L. Myers, Brenda Andrews, and Charles Boone. A global genetic interaction network maps a wiring diagram of cellular function. *Science*, 353(6306):aaf1420, 2016. doi: 10.1126/science.aaf1420. URL `https://www.science.org/doi/abs/10.1126/science.aaf1420`.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21(8):1470–1480, 2024.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M Norman, Eric S Lander, Jonathan S Weissman, Nir Friedman, and Aviv Regev. Perturb-seq: Dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *Cell*, 167(7):1853–1866.e17, 2016.

Ronald A Fisher. The correlation between relatives on the supposition of Mendelian inheritance. *Transactions of the Royal Society of Edinburgh*, 52(2):399–433, 1918.

Chris J. Frangieh, Johannes C. Melms, Pratiksha I. Thakore, Kathryn R. Geiger-Schuller, Patricia Ho, Adrienne M. Luoma, Brian Cleary, Livnat Jerby-Arnon, Shruti Malu, Michael S. Cuoco, Maryann Zhao, Casey R. Ager, Meri Rogava, Lila Hovey, Asaf Rotem, Chantale Bernatchez, Kai W. Wucherpfennig, Bruce E. Johnson, Orit Rozenblatt-Rosen, Dirk Schadendorf, Aviv Regev, and Benjamin Izar. Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion. *Nature Genetics*, 53(3):332–341, 2021. doi: 10.1038/s41588-021-00779-1.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21(8):1481–1491, 2024. doi: 10.1038/s41592-024-02305-7.

Minsheng Hao, Yongju Lee, Hanchen Wang, Gabriele Scalia, and Aviv Regev. Perturboagent: An LLM-based agent for designing iterative perturb-seq experiments. In *Proceedings of the 20th Machine Learning in Computational Biology Meeting*, volume 311 of *Proceedings of Machine Learning Research*, pp. 44–64, 2025.

Arthur E Hoerl and Robert W Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Kexin Huang, Romain Lopez, Jan-Christian Hütter, Takamasa Kudo, Antonio Rios, and Aviv Regev. Sequential optimal experimental design of perturbation screens guided by multi-modal priors. In *Research in Computational Molecular Biology*, volume 14758 of *Lecture Notes in Computer Science*, pp. 17–37. Springer Nature Switzerland, 2024. doi: 10.1007/978-1-0716-3989-4_2.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020. doi: 10.1613/jair.1.11674.

Eric Kernfeld, Yunxiao Yang, Joshua S. Weinstock, Alexis Battle, and Patrick Cahan. A comparison of computational methods for expression forecasting. *Genome Biology*, 26:388, 2025. doi: 10.1186/s13059-025-03840-y.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

Brenden M Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *PMLR*, pp. 2873–2882, 2018.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L. Ibarra, Sanjay R. Srivatsan, Mohsen Naghipourfar, Riza M. Daza, Beth Martin, Jay Shendure, José L. McFaline-Figueroa, Pierre Boyeau, F. Alexander Wolf, Nafissa Yakubova, Stephan Günnemann, Cole Trapnell, David Lopez-Paz, and Fabian J. Theis. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, 2023. doi: 10.15252/msb.202211517.

David J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992. doi: 10.1162/neco.1992.4.3.415.

Ramamurthy Mani, Robert P. St. Onge, John L. Hartman, IV, Guri Giaever, and Frederick P. Roth. Defining genetic interaction. *Proceedings of the National Academy of Sciences*, 105(9):3461–3466, 2008. doi: 10.1073/pnas.0712255105.

Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

Patrick C. Phillips. Epistasis—the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, 9(11):855–867, 2008. doi: 10.1038/nrg2452.

Yuanhao Qu, Kaixuan Huang, Ming Yin, Kanghong Zhan, Dyllan Liu, Di Yin, Henry C. Cousins, William A. Johnson, Xiaotong Wang, Mihir Shah, Russ B. Altman, Denny Zhou, Mengdi Wang, and Le Cong. CRISPR-GPT for agentic automation of gene-editing experiments. *Nature Biomedical Engineering*, 10(2):245–258, 2026. doi: 10.1038/s41551-025-01463-z.

Joseph M. Replogle, Thomas M. Norman, Albert Xu, Jeffrey A. Hussmann, Jin Chen, J. Zachery Cogan, Elliott J. Meer, Jessica M. Terry, Daniel P. Riordan, Niranjan Srinivas, Ian T. Fiddes, Joseph G. Arthur, Luigi J. Alvarado, Katherine A. Pfeiffer, Tarjei S. Mikkelsen, Jonathan S. Weissman, and Britt Adamson. 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, Lauren Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-Yanai, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L. Bonnar, Marco Jost, Thomas M. Norman, and Jonathan S. Weissman. 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*, 42:927–935, 2024.

Yusuf Roohani, Andrew Lee, Qian Huang, Jian Vora, Zachary Steinhart, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. Biodiscoveryagent: An ai agent for designing genetic perturbation experiments. In *The Thirteenth International Conference on Learning Representations*, 2025.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, and Patrick T Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008, 2017.

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

Zhiting Wei, Yiheng Wang, Yicheng Gao, Shuguang Wang, Ping Li, Duanmiao Si, Yuli Gao, Siqi Wu, Danlu Li, Kejing Dong, Xingbo Yang, Chen Tang, Shaliu Fu, Xiaohan Chen, Wannian Li, Yuzhou You, Chen Zhang, Aibin Liang, Guohui Chuai, and Qi Liu. Benchmarking algorithms for generalizable single-cell perturbation response prediction. *Nature Methods*, 23:451–464, 2026. doi: 10.1038/s41592-025-02980-0.

Aaron Wenteler, Martina Occhetta, Nikhil Branson, Victor Curean, Magdalena Huebner, William Dee, William Connell, Siu Pui Chung, Alex Hawkins-Hooker, Yasha Ektefaie, César Miguel Valdez Córdova, and Amaya Gallagher-Syed. Perteval-scfm: Benchmarking single-cell foundation models for perturbation effect prediction. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66633–66677, 2025.

Hans-Hermann Wessels, Alejandro Méndez-Mancilla, Yuhan Hao, Efthymia Papalexi, William M. Mauck, III, Lu Lu, John A. Morris, Eleni P. Mimitou, Peter Smibert, Neville E. Sanjana, and Rahul Satija. Efficient combinatorial targeting of RNA transcripts in single cells with Cas13 RNA Perturb-seq. *Nature Methods*, 20:86–94, 2023. doi: 10.1038/s41592-022-01705-x.

Xiaoyu Zhao, Jinyu Li, Zhimin Liu, and Scott Powers. Combinatorial crispr/cas9 screening reveals epistatic networks of interacting tumor suppressor genes and therapeutic targets in human breast cancer. *Cancer Research*, 81(24):6090–6105, 2021. doi: 10.1158/0008-5472.CAN-21-2555.

A Derivations and Proofs

A.1 Closed-form Blending Weight

Let $D = \hat{Y}^{\text{nl}} - \hat{Y}^{\text{lin}} \in \mathbb{R}^{M\times G}$ denote the direction of disagreement between the two base predictors and $E = Y_{\mathcal{T}} - \hat{Y}^{\text{lin}}$ the Ridge residual on training pairs. The blending objective $J(\alpha) = \|Y_{\mathcal{T}} - [\alpha\hat{Y}^{\text{nl}} + (1-\alpha)\hat{Y}^{\text{lin}}]\|_F^2$ rewrites as

$$J(\alpha) = \|E - \alpha D\|_F^2 = \|E\|_F^2 - 2\alpha\langle E, D\rangle_F + \alpha^2\|D\|_F^2. \tag{11}$$

Differentiating and setting $dJ/d\alpha = 0$ gives $\alpha^* = \langle E, D\rangle_F / \|D\|_F^2$, i.e.,

$$\alpha^* = \frac{\sum_{i,g}(y_i^{(g)} - \hat{y}_i^{\text{lin},(g)})(\hat{y}_i^{\text{nl},(g)} - \hat{y}_i^{\text{lin},(g)})}{\sum_{i,g}(\hat{y}_i^{\text{nl},(g)} - \hat{y}_i^{\text{lin},(g)})^2}. \tag{12}$$

Provided $\|D\|_F^2 > 0$, J is strictly convex and this is the unique minimum; if $D = 0$ the two predictors coincide and any $\alpha \in [0,1]$ is optimal (we default to $\alpha = 0.5$). The final weight is $\alpha = \text{clip}(\alpha^*, 0, 1)$, computed in $O(MG)$ time with no validation set.

A.2 Evidence Maximization for Kernel Regression

Marginal likelihood. We model a target column $y \in \mathbb{R}^M$ as a zero-mean Gaussian process with covariance $\Sigma = \sum_{j=1}^{J}\theta_j K_j + \sigma^2 I_M$, where K_j are the pair-level kernel matrices (Equations 6–7). The log marginal likelihood follows directly from the multivariate Gaussian density:

$$\log p(y \mid \theta, \sigma^2) = -\tfrac{1}{2}\big[y^\top\Sigma^{-1}y + \log|\Sigma| + M\log(2\pi)\big]. \tag{13}$$

For a multi-output target $Y \in \mathbb{R}^{M\times Q}$ with shared kernel weights and independent columns (as in the tail regression, where $Q = G$), summing equation 13 across columns and applying the trace identity $\sum_q y_q^\top A y_q = \text{tr}(AYY^\top)$ yields

$$\mathcal{L}(\theta, \sigma^2) = -\tfrac{1}{2}\big[\text{tr}(\Sigma^{-1}YY^\top) + Q\log|\Sigma| + MQ\log(2\pi)\big]. \tag{14}$$

Gradient. Let $v = (\theta_1, \ldots, \theta_J, \sigma^2)^\top$ and define basis matrices $B_j = K_j$ for $j \le J$, $B_{J+1} = I_M$. Using the matrix identities $\partial\log|\Sigma|/\partial v_j = \text{tr}(\Sigma^{-1}B_j)$ and $\partial\Sigma^{-1}/\partial v_j = -\Sigma^{-1}B_j\Sigma^{-1}$, the gradient in log-space ($\ell_j = \log v_j$) is

$$\frac{\partial\mathcal{L}}{\partial\ell_j} = \frac{v_j}{2}\big[\text{tr}(\Sigma^{-1}YY^\top\Sigma^{-1}B_j) - Q\,\text{tr}(\Sigma^{-1}B_j)\big]. \tag{15}$$

Parameters are initialized uniformly and optimized via L-BFGS-B with these analytic gradients. A diagonal jitter $\max(\bar{\Sigma}_{ii}\cdot\epsilon_{\text{mach}}\cdot M,\ \epsilon_{\text{mach}})\cdot I_M$ is added for numerical stability.

Predictive distribution. Given optimized parameters and Cholesky factor $LL^\top = \Sigma$, the conditional Gaussian formula gives the predictive mean and variance for a test pair (a^*, b^*):

$$\tilde{y}_* = k_*^\top\mathbf{c}, \quad \mathbf{c} = L^{-\top}L^{-1}Y, \qquad v_* = k_{**} - \|L^{-1}k_*\|^2, \tag{16}$$

where $\mathbf{c} \in \mathbb{R}^{M\times Q}$ collects the posterior weights, $k_* = \sum_j \hat{\theta}_j K_j(*, \mathcal{T})$ is the cross-kernel vector, and $k_{**} = \sum_j \hat{\theta}_j K_j(*,*)$ the self-kernel. The variance v_* enters the shrinkage mechanism (Equation 9).

A.3 Core Shrinkage Derivation

The kernel regression produces, for each program k, a predictive mean $\tilde{z}_k$ and variance v_k in score space. Rather than applying per-program shrinkage (which would require K separate signal variance estimates from limited training data), we work directly in gene space with an isotropic moment-matched approximation.

Gene-space model. Let $\tilde{r}_{\text{core}} = \sum_k \tilde{z}_k s_k u_k$ denote the unshrunk core reconstruction in gene space. We model the true core interaction as $r^*_{\text{core}} \sim \mathcal{N}(0, \tau^2_{\text{core}} I_G)$ and the reconstruction as a noisy observation $\tilde{r}_{\text{core}} \mid r^*_{\text{core}} \sim \mathcal{N}(r^*_{\text{core}}, v_{\text{core}} I_G)$. The posterior mean under this model is

$$\hat{r}_{\text{core}} = \frac{\tau^2_{\text{core}}}{\tau^2_{\text{core}} + v_{\text{core}}} \tilde{r}_{\text{core}} = w_{\text{core}} \tilde{r}_{\text{core}}, \tag{17}$$

which is the shrinkage applied in practice. A single scalar weight w_{core} is applied to the entire gene-space core vector rather than shrinking each program score independently; this avoids estimating K separate signal variances from limited training data.

Signal variance. The signal variance τ^2_{core} is estimated as the mean squared magnitude of the core reconstruction on training pairs:

$$\tau^2_{\text{core}} = \frac{1}{MG} \| Z \operatorname{diag}(s) U^\top \|^2_F, \tag{18}$$

where s_k are the program scale factors and $U \in \mathbb{R}^{G \times K}$ the program directions.

Predictive variance. The gene-space predictive variance v_{core} is obtained by propagating the score-space variances through the program directions. Since the gene-space prediction for gene g is $\tilde{r}^{(g)} = \sum_k \tilde{z}_k s_k u_{kg}$, independence across programs gives $\operatorname{Var}(\tilde{r}^{(g)}) = \sum_k v_k s_k^2 u_{kg}^2$. Averaging over genes yields the scalar summary:

$$v_{\text{core}} = \frac{1}{G} \sum_{g=1}^{G} \sum_{k=1}^{K} v_k \, s_k^2 \, u_{kg}^2. \tag{19}$$

The shrinkage weight $w_{\text{core}} = \tau^2_{\text{core}} / (\tau^2_{\text{core}} + v_{\text{core}}) \in [0, 1]$ suppresses the core correction when predictive uncertainty dominates the empirical signal scale.

A.4 Tail Shrinkage

The tail $T = R_c - Z \operatorname{diag}(s) U^\top \in \mathbb{R}^{M \times G}$ captures variation not supported as significant shared structure by the permutation test and is expected to carry weaker transferable signal than the core programs. The tail signal variance is $\tau^2_{\text{tail}} = \frac{1}{MG} \|T\|^2_F$, and we apply a conservative squared shrinkage:

$$\hat{t}_{ab} = w^2_{\text{tail}} \, \tilde{t}_{ab}, \quad w_{\text{tail}} = \frac{\tau^2_{\text{tail}}}{\tau^2_{\text{tail}} + v_{\text{tail}}}, \tag{20}$$

which suppresses tail corrections unless predictive confidence is high. The precise exponent is immaterial: varying it from 1 to 3 changes all metrics by $< 0.1\%$ on every dataset (Appendix C.4), confirming that the Bayesian shrinkage mechanism already drives tail contributions to a negligible level independently of this architectural choice.

A.5 Permutation Test for Interaction Rank

We test whether the mean-centered interaction residual $R_c \in \mathbb{R}^{M \times G}$ contains shared structure across training pairs. Let $\lambda_1 \geq \cdots \geq \lambda_M$ be the eigenvalues of $R_c R_c^\top$. The procedure is:

1. For each of $B = \max(200, 2M)$ permutations, independently permute entries within each column of R_c to obtain $R_c^{(b)}$, preserving marginal gene distributions but destroying cross-pair structure. Compute null eigenvalues $\lambda_1^{(b)} \geq \cdots \geq \lambda_M^{(b)}$.
2. Compute thresholds $\lambda_k^{(0.95)} = Q_{0.95}(\{\lambda_k^{(b)}\}_{b=1}^B)$.
3. Retain directions consecutively: $K = \max\{k : \lambda_j > \lambda_j^{(0.95)}, \ \forall\, j \leq k\}$.
4. Extract gene-space programs $U_k = R_c^\top u_k / \sqrt{\lambda_k}$ and normalized scores $z_{ik} = u_k^{(i)} \sqrt{\lambda_k} / s_k$ where $s_k = \text{rms}(u_k^{(\cdot)} \sqrt{\lambda_k})$. The full core reconstruction is then $Z \operatorname{diag}(s) U^\top$.

Gene correlation. Column-wise permutation destroys gene-gene covariance, yielding a flatter null eigenspectrum that could in principle overestimate K. Two mechanisms mitigate this: (1) the base predictor already captures the dominant covariance structure, so R has substantially reduced gene correlations; (2) downstream Bayesian shrinkage (Equation 9) suppresses the aggregate Core correction when its propagated predictive uncertainty exceeds the empirical Core signal variance, regardless of whether individual eigenvalues exceeded the permutation null.

A.6 Nonlinear Calibration via Empirical Bayes

The per-gene calibration (Equation 2) models the deviation from additivity as $t^{(g)} = y^{(g)} - s^{(g)} = \beta_1^{(g)}\phi_1^{(g)} + \beta_2^{(g)}\phi_2^{(g)} + \varepsilon^{(g)}$, where $\phi_1^{(g)} = s^{(g)}/c_1$ and $\phi_2^{(g)} = (s^{(g)})^2/c_2$ are RMS-normalized features. Training data consist of (1) training pairs with input $s_{ab} = \delta_a + \delta_b$ and target δ_{ab}, and (2) single effects with input = target = δ_g, which anchors the correction toward the identity at observed single-effect magnitudes.

We place Gaussian priors $\beta_j^{(g)} \sim \mathcal{N}(0, \alpha_j^{-1})$ and a shared likelihood precision β. Since each gene's design matrix $\Phi^{(g)} \in \mathbb{R}^{N\times 2}$ yields a 2×2 posterior precision $\Lambda^{(g)} = \mathrm{diag}(\alpha) + \beta\Phi^{(g)\top}\Phi^{(g)}$, the posterior covariance $\Sigma^{(g)} = (\Lambda^{(g)})^{-1}$ and mean $m^{(g)} = \beta\Sigma^{(g)}\Phi^{(g)\top}t^{(g)}$ are computed in closed form via 2×2 matrix inversion.

The hyperparameters are updated by the standard evidence maximization fixed-point equations (MacKay, 1992):

$$\gamma_j = \textstyle\sum_g (1 - \alpha_j \Sigma_{jj}^{(g)}), \qquad \text{(effective parameters for feature } j\text{)} \tag{21}$$

$$\alpha_j^{\text{new}} = \gamma_j \big/ \textstyle\sum_g (m_j^{(g)})^2, \qquad \text{(prior precision update)} \tag{22}$$

$$\beta^{\text{new}} = (NG - \gamma_1 - \gamma_2) \big/ \textstyle\sum_g \|t^{(g)} - \Phi^{(g)} m^{(g)}\|^2, \qquad \text{(noise precision update)} \tag{23}$$

iterated until convergence (maximum relative change in log-parameters below $\sqrt{\epsilon_{\text{mach}}}$, typically 50–100 iterations). Genes where the linear or quadratic features explain significant variance receive large coefficients; genes with no systematic nonlinearity are shrunk toward zero.

A.7 Kernel Construction

Normalized Gram matrix. Let $E \in \mathbb{R}^{N\times G}$ be the single-effect matrix and $E_c = E - \mathbf{1}\bar{e}^\top$ its column-centered version. We set $S = N \cdot E_c E_c^\top / \mathrm{tr}(E_c E_c^\top)$ so that $\mathrm{tr}(S) = N$ (average self-similarity is 1), ensuring kernel weights are scale-invariant across datasets.

Additive kernel k_1. Representing pair (a, b) by $\mathbf{p}_{ab} = \mathbf{e}_a + \mathbf{e}_b$, we define $k_1((a, b), (c, d)) = \frac{1}{2}\mathbf{p}_{ab}^\top S\, \mathbf{p}_{cd} = \frac{1}{2}(S_{ac} + S_{ad} + S_{bc} + S_{bd})$. Since $S = LL^\top$ for some L, this is the inner product $\frac{1}{2}\langle L^\top \mathbf{p}_{ab}, L^\top \mathbf{p}_{cd}\rangle$, hence positive semi-definite (PSD).

Multiplicative kernel k_2. We define $k_2((a, b), (c, d)) = \frac{1}{2}(S_{ac}S_{bd} + S_{ad}S_{bc})$, which is large when both genes of one pair are simultaneously similar to the respective genes of another. This is the symmetrization of the Kronecker product $(S \otimes S)_{(a,b),(c,d)}$; since the Kronecker product of PSD matrices is PSD and symmetrization preserves PSD-ness, k_2 is PSD.

Combined kernel. The full kernel $k_\theta = \theta_1 k_1 + \theta_2 k_2 + \sigma^2 I$ is PSD for any $\theta_1, \theta_2 \geq 0$, $\sigma^2 > 0$. The marginal likelihood (Equation 14) automatically determines the relative contribution of additive versus combinatorial similarity. Computational cost is $O(M^2)$ for kernel evaluation plus $O(M^3)$ for Cholesky decomposition; negligible for $M \leq 94$.

A.8 Full Prediction Pipeline

For a test pair (a^*, b^*), prediction proceeds in four stages.

Base prediction. Compute $s = \delta_{a^*} + \delta_{b^*}$, apply the nonlinear calibration $\hat{\delta}^{\text{nl}} = f(s)$ (Equation 2), the Ridge predictor $\hat{\delta}^{\text{lin}} = W^\top(\mathbf{e}_{a^*} + \mathbf{e}_{b^*})$, and blend: $\hat{\delta}^{\text{base}} = \alpha\hat{\delta}^{\text{nl}} + (1 - \alpha)\hat{\delta}^{\text{lin}}$.

Core interaction. For each program $k = 1, \ldots, K$, predict score $\tilde{z}_k$ and variance v_k via kernel regression. Reconstruct the unshrunk core prediction in gene space as $\tilde{r}_{\text{core}} = \sum_k \tilde{z}_k \, s_k \, u_k$, propagate score-space variances to gene-space variance v_{core} (Equation 19), and apply shrinkage: $\hat{r}_{\text{core}} = w_{\text{core}} \, \tilde{r}_{\text{core}}$ with $w_{\text{core}} = \tau^2_{\text{core}} / (\tau^2_{\text{core}} + v_{\text{core}})$.

Tail interaction. Predict $\tilde{t}$ and v_{tail} via the tail kernel regression, then apply squared shrinkage: $\hat{t} = w^2_{\text{tail}} \, \tilde{t}$.

Assembly. $\hat{\delta}_{a^* b^*} = \hat{\delta}^{\text{base}} + \bar{r} + \hat{r}_{\text{core}} + \hat{t}$.

A.9 Adaptive Complexity

Proposition A.1 (Reduction to base-plus-offset predictor)**.** *If $K = 0$ and $w^2_{\text{tail}} \to 0$, then $\hat{\delta}_{ab} = \hat{\delta}^{\text{base}}_{ab} + \bar{r}$.*

Proof. $K = 0$ implies the core sum is empty. $w^2_{\text{tail}} \to 0$ implies $\hat{t} \to 0$. Equation 10 gives $\hat{\delta}_{ab} = \hat{\delta}^{\text{base}} + \bar{r} + 0 + 0$. This is the behavior observed on DirectCapture (Table 4). □

Proposition A.2 (Reduction to Ridge-plus-offset predictor)**.** *If additionally $\alpha = 0$, $K = 0$, and $w^2_{\text{tail}} \to 0$, then $\hat{\delta}_{ab} = \hat{\delta}^{\text{lin}}_{ab} + \bar{r}$.*

Proof. $\alpha = 0$ in Equation 3 gives $\hat{\delta}^{\text{base}} = \hat{\delta}^{\text{lin}}$. Combining with Proposition A.1 yields the result. Note that on Norman, Wessels2023, and GSE164996, the base blending selects $\alpha = 0$ (the Ridge branch), but these datasets retain a nonzero interaction rank $K > 0$ (Table 4), so the full HNR prediction includes additional Core and Tail interaction corrections beyond the Ridge-plus-offset form. □

Proposition A.3 (Reduction to additive predictor)**.** *If $K = 0$, $w^2_{\text{tail}} \to 0$, $\alpha = 1$, $\bar{r} = 0$, and $\beta_1 = \beta_2 = 0$, then $\hat{\delta}_{ab} = \delta_a + \delta_b$.*

Proof. $\beta_1 = \beta_2 = 0$ makes the calibration f the identity, so $\hat{\delta}^{\text{base}} = f(\delta_a + \delta_b) = \delta_a + \delta_b$. With $K = 0$, $\hat{t} = 0$, $\bar{r} = 0$, the result follows. □

Adaptive model family. The base predictor interpolates between two branches via α:

$$\text{Additive} \xleftarrow[\beta_1 = \beta_2 = 0]{\alpha = 1} \text{Calibrated additive} \xrightarrow{\text{blend}} \text{Base} \xleftarrow{\alpha = 0} \text{Ridge}, \tag{24}$$

and the full model adds interaction corrections on top:

$$\text{Base} + \bar{r} \xrightarrow{K > 0} \text{Base} + \bar{r} + \text{Core} \xrightarrow{+\,\text{Tail}} \text{Full HNR}. \tag{25}$$

The blending weight α selects which branch dominates the base; the interaction rank K and the shrinkage factors determine whether and how much interaction correction is applied. All selections are automatic; no hyperparameter controls which configuration is active.

B Experimental Details

B.1 Datasets

Norman (Norman et al., 2019). CRISPRa perturbations in K562 cells. The dataset contains 105 single-gene and 131 combinatorial perturbation conditions. Among the combinatorial conditions, 128 double perturbations have both constituent genes observed as singles and are eligible for compositional evaluation. Perturbation effects are computed as the difference between the mean expression of perturbed cells and the mean expression of unperturbed controls.

Wessels2023 (Wessels et al., 2023). Cas13-based combinatorial RNA Perturb-seq (CaRPool-seq) in THP-1 cells. After filtering for double perturbations where both constituent genes have corresponding single-gene conditions, 158 eligible doubles remain. Effects are computed relative to non-targeting control cells.

DirectCapture (GSE146194) (Replogle et al., 2020). Direct-capture Perturb-seq experiment 6 (GSM4367984). Preprocessing follows a source-matched protocol independent of train/validation/test membership: (i) cells are restricted to those with published perturbation identities; (ii) each cell is rescaled by its total gene-expression UMI count; (iii) genes with mean raw expression ≥ 1 UMI/cell are retained; (iv) each gene is z-normalized against the non-targeting control population; (v) normalized expression is averaged by perturbation condition (pseudobulk); and (vi) unordered cross-gene doubles for which both singles are observed are retained. This yields 37 eligible double perturbations.

GSE164996 (Zhao et al., 2021). Matched two-guide combinatorial CRISPR screen (sample S2). Only exact two-guide cells are used, ensuring singles and doubles share the same experimental context. Single-gene effects are defined as $\mu(\text{gene}+\text{CTRL})-\mu(\text{CTRL}+\text{CTRL})$, and double effects as $\mu(\text{geneA}+\text{geneB})-\mu(\text{CTRL}+\text{CTRL})$. The interaction residual is therefore

$$I_{ab} = \mu(a{+}b) - \mu(a{+}\text{CTRL}) - \mu(b{+}\text{CTRL}) + \mu(\text{CTRL}{+}\text{CTRL}), \tag{26}$$

keeping single and double effects in the same two-guide context. Normalization uses counts per 10,000 (CP10K) followed by log1p. Same-gene doubles and hRosa26 pairs are excluded. This yields 54 eligible double perturbations.

Splitting protocol. On all four datasets, eligible double perturbations are randomly split 60%/20%/20% into training, validation, and test partitions using fixed random seeds. For Norman, this yields 76 training, 26 validation, and 26 test pairs. All measured single-perturbation effects remain observed across all partitions. The validation partition is not used by HNR: the base blending weight α, interaction rank K, kernel weights, and shrinkage factors are estimated using only training pairs and observed single-perturbation data. Neural baselines use the validation partition for model selection and early stopping where applicable. No dataset-specific model hyperparameter tuning is performed for HNR.

B.2 Metrics

All metrics are computed on held-out double perturbations and averaged across evaluations. Values are scaled by $\times 100$ for readability; lower is better throughout.

- **MAE**: $\frac{1}{G}\sum_{g=1}^{G}|\hat{\delta}_{ab}^{(g)} - \delta_{ab}^{(g)}|$.
- **Top-k** ($k \in \{10, 20\}$): MAE restricted to the k genes with the largest absolute observed effects $|\delta_{ab}^{(g)}|$, selected independently for each evaluation.
- **NonAdd-k** ($k \in \{10, 20\}$): MAE restricted to the k genes with the largest absolute interaction residuals $|r_{ab}^{(g)}| = |\delta_{ab}^{(g)} - \delta_{a}^{(g)} - \delta_{b}^{(g)}|$, selected independently for each evaluation.
- **IntWMAE**: $\frac{\sum_g w_g |\hat{\delta}_{ab}^{(g)} - \delta_{ab}^{(g)}|}{\sum_g w_g}$, where $w_g = |r_{ab}^{(g)}|$.

B.3 BASELINES

Table 5 summarizes the baseline methods and their main implementation choices. The comparison spans simple additive and linear predictors, standard neural architectures, pretrained single-cell models, and graph-based perturbation models. To ensure a fair comparison under the same condition-level compositional prediction setting, non-graph models, including CPA and scGPT, are trained on condition-level mean perturbation effects, whereas GEARS is trained on single-cell expression profiles using its original data pipeline. GEARS and GEARS+scGPT are evaluated only on conditions whose constituent genes are represented in the GO graph.

Table 5: Baseline method details.

Model	**Paradigm**	**Key details**
Additive	Prior	$\hat{\delta}_{ab} = \delta_a + \delta_b$; no training
Ridge	Linear	One-hot indicators, L_2 regularization (λ=1.0)
MLP	Feedforward	64-dim gene embeddings, 2×256 hidden, ReLU, LayerNorm, dropout 0.1
Transformer	Attention	64-dim embeddings, 4 heads, 2 layers, 256-dim FFN
CPA	Comp. VAE	64-dim latent, KL annealing to 0.1 over 100 epochs, max 300 epochs
scGPT	Foundation	Frozen 512-dim scGPT embeddings → 128-dim, 3-layer Transformer head
GEARS	Graph	GO + co-expression graphs, hidden dim 64, 15 epochs
GEARS+scGPT	Graph+FM	GEARS with scGPT embeddings reduced to 64-dim via PCA

B.4 IMPLEMENTATION DETAILS

HNR is a lightweight statistical framework requiring no GPU acceleration, pretraining, external knowledge graph, or validation-set hyperparameter tuning.

Nonlinear calibration. The per-gene coefficients β_1, β_2 in Eq. 2 are estimated by empirical Bayes evidence maximization. Features are standardized by their root-mean-square values before fitting. The procedure runs for up to 512 iterations and stops when the maximum absolute change in log-parameters falls below $\sqrt{\epsilon}$, where ϵ is machine epsilon ($\approx 2.2 \times 10^{-16}$). Training data consist of the concatenation of all training pair effects and all observed single-perturbation effects.

Interaction rank selection. The interaction residual matrix $R \in \mathbb{R}^{M \times G}$ is mean-centered and its eigenvalues are compared against a null distribution obtained by independently permuting gene values within each column. The number of permutations is $B = \max(200, 2M)$ and the retention threshold is the 95th percentile. Eigenvalues are retained consecutively from the largest:

$$K = \max\{k : \lambda_j > \lambda_j^{(0.95)},\ \forall\, j \le k\}.$$

Kernel regression. Kernel weights and noise variances are initialized and optimized via L-BFGS-B on the negative log marginal likelihood. A small diagonal jitter is added for numerical stability:

$$\max(\mathrm{mean}(\mathrm{diag}(\Sigma)) \cdot \epsilon \cdot n, \epsilon)\, I$$

Shrinkage. The unshrunk Core reconstruction $\tilde{r}_{\text{core}} = \sum_k \tilde{z}_k s_k u_k$ is regularized by a single gene-space Bayesian shrinkage weight $w_{\text{core}} = \tau^2_{\text{core}}/(\tau^2_{\text{core}} + v_{\text{core}})$, where τ^2_{core} is the empirical signal variance and v_{core} is the aggregate predictive variance propagated from score space (Appendix A.3). The Tail prediction is shrunk more aggressively using the squared shrinkage weight.

Base blending. The blending weight $\alpha \in [0, 1]$ is computed in closed form by projecting training residuals $y - \hat{\delta}^{\text{lin}}$ onto the direction $\hat{\delta}^{\text{nl}} - \hat{\delta}^{\text{lin}}$ and clipping the result to $[0, 1]$.

Computational cost. On the Norman training split (76 training pairs, 8,564 genes), full HNR model fitting completes in 3.8 s on CPU.The kernel evidence optimization requires an $O(M^3)$ Cholesky decomposition over the M observed combinations; while negligible at the scales considered here ($M \le 94$), this may become a computational bottleneck for substantially larger combinatorial screens.

C Additional Results

C.1 Statistical Significance

Table 6 reports paired permutation tests comparing HNR with representative baselines across all four datasets. HNR shows statistically significant improvements over all neural baselines on full-transcriptome MAE. Relative to Ridge, the difference is negligible on Wessels2023, whereas the largest gains over Ridge occur on Norman and DirectCapture. The table also reports the number of held-out evaluations on which HNR achieves lower MAE, providing a complementary view of the consistency of the improvement.

Table 6: Paired permutation tests with 100,000 permutations, pooled across seeds. ΔMAE is HNR minus the corresponding baseline ($\times 100$; negative values favor HNR). Wins denotes the number of held-out evaluations for which HNR achieves lower MAE than the baseline.

Dataset	Baseline	ΔMAE	p	Wins
Norman	Additive	-0.29	< 0.001	84/104
	Ridge	-0.07	< 0.001	80/104
	Best neural (CPA)	-0.58	< 0.001	103/104
	GEARS	-2.22	< 0.001	104/104
Wessels2023	Additive	-1.74	< 0.001	128/128
	Ridge	-0.00	0.504	67/128
	Best neural (CPA)	-0.07	< 0.001	103/128
	GEARS	-1.10	< 0.001	105/105
DirectCapture	Additive	-8.03	0.027	20/28
	Ridge	-34.6	< 0.001	28/28
	Best neural (MLP)	-68.0	< 0.001	27/28
	GEARS	-72.1	< 0.001	28/28
GSE164996	Additive	-0.80	< 0.001	44/44
	Ridge	$+0.00$	0.014	20/44
	Best neural (Transformer)	-0.12	< 0.001	42/44
	GEARS	-0.63	< 0.001	37/37

C.2 Biological Interpretation of Interaction Programs

Table 7: Interaction programs discovered on Norman (seed 42, $K=6$; top three shown). Programs are identified automatically from the interaction residual by permutation-based decomposition.

Program	Var.%	Top positive genes	Top negative genes
1	30.8	SRM, NME1, PTMA, RANBP1	BTG1, TYROBP, LST1, CSF3R
		Top pairs: CEBPE+KLF1, FOSB+CEBPB, CEBPE+SPI1	
2	12.2	HBZ, HBG2, HBG1, ALAS2	AIF1, LY6E, IFITM2, GMFG
		Erythroid: hemoglobin synthesis + heme biosynthesis	
3	8.2	GYPB, GYPA, ID3, ID1	CFD, LST1, TYROBP, MS4A3
		Erythroid surface markers vs. monocyte/myeloid identity	

Table 7 summarizes the three leading interaction programs discovered on Norman for a representative seed. For each program, we report the fraction of interaction variance explained, the genes with the strongest positive and negative loadings, and representative high-scoring perturbation pairs. These programs are obtained directly from the interaction residual and are not defined using external biological annotations. Program 1 explains 30.8% of the interaction variance and separates proliferative genes such as SRM, NME1, and PTMA from myeloid differentiation markers including BTG1, TYROBP, LST1, and CSF3R. Its strongest perturbation scores are associated with combinations of hematopoietic transcription factors, including CEBPE+KLF1, FOSB+CEBPB, and CEBPE+SPI1. Program 2 explains 12.2% of the interaction variance and captures an erythroid-associated axis dominated by hemoglobin and heme-biosynthesis genes such as HBZ, HBG2, HBG1, and ALAS2.

Program 3 explains 8.2% and contrasts erythroid surface markers and differentiation-related genes with monocyte/myeloid-associated markers.

C.3 GEARS-ELIGIBLE SUBSET COMPARISON

GEARS can evaluate only perturbations whose constituent genes are represented in its GO graph, resulting in a smaller evaluation subset on some datasets. Table 8 therefore evaluates all methods on exactly the same GEARS-eligible test conditions. HNR retains strong performance under this restricted comparison, indicating that its advantage is not explained by differences in evaluation coverage.

Table 8: All methods evaluated on the GEARS-eligible subset of test conditions. All metrics are scaled by $\times 100$ ($\downarrow$); results are mean $\pm$ std over four seeds. Numbers in parentheses indicate the number of GEARS-eligible test pairs relative to the full test set.

	Norman (26/26)			Wessels2023 (26/32)			DirectCapture (7/7)			GSE164996 (9/11)		
Method	MAE	NA20	IntW	MAE	NA20	IntW	MAE	NA20	IntW	MAE	NA20	IntW
Additive	1.40	33.71	11.35	3.28	45.23	11.39	50.92	3334	20643	1.90	81.02	15.06
Ridge	1.19	19.85	7.58	1.61	13.50	4.18	77.49	3769	24515	1.13	18.93	5.50
MLP	2.11	24.47	9.31	1.88	14.50	4.33	110.89	7353	53243	1.96	27.87	6.17
Transformer	2.10	36.77	13.15	2.08	19.96	5.29	113.04	7334	53198	1.25	30.69	6.13
CPA	1.69	22.63	8.67	1.67	13.81	4.16	116.07	7176	52225	1.51	27.06	5.94
scGPT	2.09	36.98	13.18	2.11	20.09	5.32	112.94	7451	53901	1.50	31.74	6.28
GEARS+scGPT	3.27	21.58	10.75	2.75	15.29	5.81	115.24	7547	54456	1.74	32.79	7.89
GEARS	3.34	21.16	10.70	2.72	14.42	5.69	114.96	7547	54456	1.76	32.18	7.89
HNR (ours)	**1.11**	**17.93**	**6.94**	**1.61**	**12.87**	**4.13**	**42.89**	**2184**	**11370**	**1.13**	**17.30**	**5.34**

C.4 TAIL SHRINKAGE EXPONENT SENSITIVITY

The Tail prediction in HNR is shrunk by raising the Bayesian shrinkage weight to an exponent, with a default value of 2. We evaluate exponents in $\{1.0, 1.5, 2.0, 2.5, 3.0\}$ across all four datasets and all reported metrics. The maximum relative change from the default is less than 0.1% on every dataset. For example, Norman MAE remains at 1.11 and NA20 varies only between 17.92 and 17.93 across all five exponent values, while Wessels2023, DirectCapture, and GSE164996 are unchanged to displayed precision. On DirectCapture, where the interaction rank is zero, the exponent is irrelevant by construction. On the remaining three datasets, Bayesian shrinkage already suppresses Tail contributions to a negligible magnitude, making the precise exponent effectively immaterial. These results support the squared shrinkage as a fixed architectural default rather than a tuned hyperparameter.