# BioJEPA-AC: Self-Supervised "World Model" for Cells


## Abstract

This report presents BioJEPA-AC (Biological Joint-Embedding Predictive Architecture - Action Conditioned), a joint-embedding predictive architecture uses a shared latent space to learn cell dynamics and perturbation response. By integrating the representation learning capabilities of the Joint-Embedding Predictive Architecture (JEPA) with a multi-modal Action-Conditioned (AC) mechanism, BioJEPA-AC operates in a learned latent space to simulate the trajectory of cellular states under perturbation. We detail the theoretical underpinnings, including the transition from generative to predictive paradigms. Training is framed as masked latent prediction with an EMA teacher, VICReg-style regularization, and an auxiliary action-state alignment objective.

We show that with training limited to just the K562-essential dataset [1] using the test splits mirroring GEARS[9] we're able to achieve a MSE 0.4979, top-20 Pearson correlation 0.9266, a $R^2$ over all genes with a mean 0.9175 (median 0.9272), and a $R^2$ over the top 50 differentially expressed genes with a mean 0.0962 (median 0.3211).


## 1. Introduction

### 1.1 The Challenge of Simulating Cell Dynamics

Large-scale single-cell perturbation screens (e.g., CRISPRi Perturb-seq) produce paired information containing a perturbation identity and a high-dimensional transcriptional readout. The K562 Perturb-seq resource associated with Replogle et al. provides a reference setting for this problem[1]. With BioJEPA-AC, the modeling goal is to learn relationship dynamics between control and perturbed cells, in relation to a perturbation, through a shared latent space. By learning in the latent space instead of on discrete expression counts, the goal is to create a model with a better understanding of the physics of cells leading to better generalization.

### 1.2 Why Predicting Perturbation Responses is Hard

Predicting perturbation response helps validate how well a model has learned cell dynamics. If a model can predict the impact on a wide range of unseen perturbations, we can see that the model is simulating cell environments. The key data for learning and evaluating these models starts with perturbSeq based on single-cell seuqnecing. Because of the following properties, using scRNA-seq perturbation screen on a model that learnes in the latent space rather than directly on the expression-space seems fruitful:

- Dimensionality: expression vectors span thousands of genes, and the effective degrees of freedom depend on preprocessing and feature selection.
- Heterogeneity: cells under the same perturbation condition can occupy multiple transcriptional states [1].
- Measurement variation: scRNA-seq readouts include technical and sampling variation; our approach treats gene expression as an observation used to learn stable latent representations.

## 1.3 Context in Related Work

Supervised perturbation-response predictors such as CPA[8] and GEARS [9] model perturbation effects directly in expression space with different inductive biases (covariate factoring vs gene network structure). Pretrained single-cell encoders (e.g., scGPT[10], Geneformer[11]) provide transferable representations that can be adapted to downstream prediction tasks. BioJEPA-AC differs in its primary objective: masked latent prediction with an EMA teacher, plus explicit action conditioning[3-7].

More recent work has emphasized distribution matching over cell populations and inference-time use of cellular context. STATE frames perturbation response prediction as a transition between distributions of cells conditioned on covariates (e.g. cell line, batch), and trains a transformer-based state-transition model using distributional discrepancy objectives (maximum mean discrepancy, commonly instantiated with an energy-distance kernel).[12] STACK is a self-supervised cell-set foundation model that refines cell embeddings using context provided by other cells in the same set. It further defines an in-context “cell prompting” setup in which a prompt population conditions predictions for a query population, trained with distributional alignment losses (energy distance) on embeddings and gene expression. [13]

## 1.4 The Joint-Embedding Predictive Architecture (JEPA) Paradigm

To target some of the shortcomings of existing approaches, BioJEPA-AC adopts the Joint-Embedding Predictive Architecture (JEPA) paradigm, pioneered in computer vision and grounded in the energy-based model theory of LeCun. The fundamental insight of JEPA is to abandon the reconstruction of raw inputs entirely. Instead, the model learns to predict the latent representation of the system based on identifying missing information based on the latent representation of the context.[2]

In this framework, the loss function operates on the representations in the abstract latent space, not the input space. The latent space allows the model to learn representations of similar concepts and the interplay between concepts in a more compact space. Because of this, the objective shifts from "What are the exact counts of the masked genes?" to "What is the biological state represented by the masked genes?".

## 1.5 BioJEPA-AC: The Action-Conditioned World Model

BioJEPA-AC can be interpreted through a JEPA/energy-based learning lens in which training minimizes a discrepancy between latents without specifying an explicit likelihood in expression space[2]. In BioJEPA-AC, the predictor is conditioned on an action embedding and is trained to assign low discrepancy to teacher latents consistent with the observed perturbed cell and the provided perturbation metadata.

We posit that a cell is a dynamical system, and perturbations (e.g. CRISPRi knockouts) are "actions" that transition the system from one state to another. By conditioning the predictor network on an embedding of the perturbation, BioJEPA-AC learns a causal world model of the cell. It approximates the transition function:

$$z_{t+1} \approx P(z_t, a_t) \tag{1}$$

where $z_t$ is the current cellular state (control), $a_t$ is the perturbation, and $z_{t+1}$ is the resulting phenotype (case).

# 2. Theoretical Framework

## 2.1 Energy-Based Models (EBMs) and Representation Learning

At its core, BioJEPA-AC is an Energy-Based Model (EBM). The goal of an EBM is to learn an energy function $E(x, y)$ that assigns low energy values to compatible pairs of variables (e.g., a cell state and its valid perturbation response) and high energy values to incompatible pairs. In the JEPA formulation, the energy is defined as the distance between the predicted representation and the target representation in latent space[2].

Unlike contrastive methods (e.g., CLIP, SimCLR), which require "negative pairs" to push incompatible states apart, JEPA relies on regularized predictive learning[2]. Contrastive learning is particularly challenging in single-cell biology because defining a "negative" is semantically ambiguous; two cells from different batches or ostensibly different types might be biologically identical in their regulatory state (e.g., two T-cells in different phases of the cell cycle).

BioJEPA-AC avoids this by minimizing the prediction error for positive pairs while using architectural constraints (asymmetric teachers) and minimizing masked latent prediction error together with VICReg regularization to discourage representational collapse[6] This can be viewed as learning a compatibility function between predicted and target latents without specifying an explicit likelihood in observation space[2]. The loss can be expressed as minimizing a reconstruction-like discrepancy in latent space:

$$\min_{\theta} \ \mathbb{E}\left[\|z_t - \hat{z}_t\|_1\right] \tag{2}$$

## 2.2 The Manifold Hypothesis in Single-Cell Data

The manifold hypothesis suggests that while scRNA-seq data exists in a high-dimensional Euclidean space ( $\mathbb{R}^{20000}$), the biologically valid states and relationships lie on a much more compressed, lower-dimensional manifold. Generative models often struggle because they attempt to approximate the probability density of the entire high-dimensional space. BioJEPA-AC, by projecting cell state data into a low-dimensional latent bottleneck (e.g., $\mathbb{R}^{256}$ in implementation), targets learning the topology of this manifold.

The "Action-Conditioned" component essentially learns vector fields on this manifold. If the manifold represents the landscape of possible cellular identities similar to Waddington's epigenetic landscape, the action embeddings represent the forces that push cells along specific trajectories, into valleys within that landscape.

## 2.3 Causal Inference vs. Correlation

Standard correlation networks (e.g., weighted gene co-expression network analysis) capture symmetric relationships: if Gene A and Gene B are co-expressed, they are linked. However, correlation does not imply causality. Perturbation data, such as the K562 CRISPR screen [1], provides ground-truth causal information: "If I break Gene A, Gene B changes."

BioJEPA-AC attempts to internalize this causality through its learning of the latent space and how actions impact a cell representation in the latent space. By forcing the model to predict the consequence of an action $a$ on the context $z$, the model must learn the directional edges representative of pseudo regulatory network and pathways. The action-conditioned formulation treats a perturbation label (and its metadata) as an intervention variable $a$ that adjusts, or "conditions", the mapping from control to perturbed states[2,5]. *This moves the field from descriptive "atlasing" to predictive "simulation".*

# 3. BioJEPA-AC Architecture

The BioJEPA-AC architecture is designed to handle permutation invariance (genes have no inherent order), continuous values (expression counts), and extreme sparsity inherent to Perturb-seq data. The architecture comprises three main modules:

- **Cell State Encoder:** maps the latent space representing cell states, currently based on the gene expression. This is used for both the student and teacher
- **Action Composer:** maps perturbation metadata to a fixed-dimensional action latent allowing different perturbation modalities to co-embed.
- **Action-Conditioned Predictor:** predicts perturbed latent conditioned on control latent and the action latent.

## 3.1 Linear Projection with Gene Identity

Each cell is comprised of a set of gene identities represented via learned embeddings scaled based on the level of observed expression. The total expression magnitude is incorporated via an explicit value encoding (including a total-count feature) before passing tokens through transformer layers. Each gene $i$ has a learned embedding vector $e_i \in \mathbb{R}^d$. The scalar expression value $x_i$ scales the gene embedding: $h_i = x_i \cdot e_i$. The total mRNA count (library size) is projected to a vector and added as a bias term to every gene embedding. This explicitly models total detected genes as a global covariate allowing models to learn when cells are no longer viable. This "bag-of-genes" approach preserves the continuous nature of the data and allows the create a cell state latent by differentiatiating between genes solely based on their learned identity embedding $e_i$.

## 3.2 Cell State Encoder

The cell-state encoder is a transformer encoder operating over gene tokens and uses a feature-map (linear) attention variant to reduce quadratic scaling[16]. The encoder produces contextualized latent tokens $z \in \mathbb{R}^{G \times D}$ representing the input cell in the shared latent space.

### 3.2.1 Linear Attention Mechanism

A major bottleneck in applying transformers to scRNA-seq is the quadratic complexity of self-attention ($O(L^2)$), where $L$ is the number of genes. Since a cell can express >20,000 genes, standard attention is computationally prohibitive. The implementation utilizes a shallow set transformer (6 layers, 4 heads, 256 embedding dims) with an efficient linear attention mechanism. Instead of softmax attention, the encoder uses a linear attention variant based on a non-negative feature map (e.g., $\phi(x) = \mathrm{ELU}(x) + 1$), reducing attention complexity to $O(N \cdot L)$ (linear in the number of genes).[16] The encoder treats genes as an unordered set (no positional

encoding), aligning with the biological reality that the physical order of genes on a chromosome does not strictly dictate their regulatory function.

### 3.2.2 The Student-Teacher Framework

BioJEPA-AC maintains a student encoder and, to provide stable targets for the predictor, an exponential moving average (EMA) teacher encoder that is structurally identical to the student encoder. This shared sturcture allows for a unified embedding space.

The teacher parameters $\xi_{\mathrm{t}}$ are not updated via gradient descent. Instead, they track the student parameters $\theta_{\mathrm{s}}$ via exponential moving average based on a specified momentum $m$ (set to 0.996[14]):

$$\xi_{\mathrm{t}} \leftarrow m\xi_{\mathrm{t}} + (1-m)\theta_{\mathrm{s}} \tag{3}$$

In practice, this means teacher processes the target view ($x_{target}$) to generate the ground-truth embedding $z_{target}$ which we calcualte loss against, preventing "target chasing" instabilities.

### 3.2.3 Masking Strategy

Pretraining and action-conditioned training randomly masks a subset of gene positions and predicts their teacher latents. We use a default mask ratio is 0.6 (60%), which is consistent with masked-reconstruction regimes used in other domains (e.g., masked autoencoders)[15].

## 3.3 The Action Composer

To enable our "AC" variant to predict perturabation impact, we use an action composer to encode heterogeneous perturbations into a unified latent space and use them to shift our cell state latent.

### 3.3.1 Multi-Modal Perturbation Embeddings (Preprocessing)

The composer ingests raw features depending on the perturbation source:

- **DNA (CRISPR Guides):** Processed by a pretrained Nucleotide Transformer v3[18] yielding 1,536-dimensional vectors. This captures sequence-specific efficiency and off-target potential.
- **Protein (Overexpression/Targets):** Processed by ESM-2[19], yielding 320-dimensional vectors based on amino acid sequences. This provides a rich prior on the function of the target gene and the function of introduced protein perturbations (e.g. mABs)
- **Chemical (Small Molecules):** When introduced, these will be [processed via Morgan fingerprints or SMILES embeddings (768-dimensional).

These features are projected via linear layers into a shared content latent space $c \in \mathbb{R}^{320}$. Since we limited our current training to the K562 Essential dataset, our only perturbation at this point is CRISPRi[1]. To process these perturbation we created embeddings for the sgRNA using the DNA modality above and converted the target gene identifiers to protein sequences which we embedded with ESM-2.

### 3.3.2 Mode-of-Action and FiLM Conditioning

To distinguish between different mechanisms targeting the same gene (e.g., CRISPRi knockdown vs. CRISPRa activation), the model uses a learnable mode embedding $m \in \mathbb{R}^{64}$. This is combined with the content latent using Feature-wise Linear Modulation (FiLM) to map perturbation embeddings to a shared action latent[17]:

$$\begin{gathered} s = \mathrm{Linear}(m), \quad t = \mathrm{Linear}(m) \\ a = c \odot (1 + s) + t \end{gathered} \tag{4}$$

The scale ($s$) and shift ($t$) parameters modulate the action vector. This allows the model to "invert" the direction of a perturbation vector if the mode switches from activation to inhibition, providing a flexible grammar for perturbation.

## 3.4 The Action-Conditioned Predictor (Causal Simulator)

A key component of our "AC" variant is the Action-Coditioed Predictor. The predictor network functions as the causal simulator. It consumes control latents $z_c$ and action latent $a$, forms learnable target queries for masked genes, and applies attention blocks to produce per-token level vectors to shift our cell state representation in the shared latent space.

- **Cross-Attention Injection:** The predictor is a transformer decoder to the shared latent space. The action latent $a$ is treated as the **Key (K)** and **Value (V)**, while the cell state tokens act as **Queries (Q)**. This "injects" the perturbation information globally into every token.
- **Dynamics Propagation:** Subsequent self-attention layers allow the tokens to update their states based on the injected perturbation and their neighbors, modeling the propagation of the signal through the learned interactions.
- **Stochastic Output:** The predictor outputs two heads for each token latent: a mean $\mu$ and a log-variance $\log \sigma^2$. This allows the model to estimate epistemic uncertainty, identifying features where the prediction is low-confidence (e.g., due to sparsity or lack of training data).

## 3.5 Production Configuration

| Item | Value |
|---|---|
| Encoder (JEPA student/teacher) | embd = 256<br>heads = 4<br>layers = 6 |
| Action Composer | lat_dim = 320<br>mod_dim = 64 |
| Parameter counts | JEPA: 6,019,840<br>AC: 7,881,216<br>Pert: 882,880 |
| Training Length | Pretraining: 100 Epochs<br>Action Comp: 1000 Epochs<br>Training: 20 Epochs |

# 4. Data Sources

The capabilities of BioJEPA-AC are inextricably linked to the scale and quality of its training data. While the model architecture is built to scale to different modalities of cell state representation and perturbations, we've limited initial runs to just a single dataset.

## 4.1 The K562 Essential Perturb-seq Dataset

BioJEPA-AC is trained and evaluated on the GEARS packaging of the K562 genome-scale Perturb-seq CRISPRi dataset ("K562 Essential").[1,9] The dataset contains pooled single-cell profiles with genetic perturbations and matched controls, together with guide/target annotations. GEARS provides standardized train/validation/test splits and preprocessing utilities that are used in this work.[9] Preprocessing follows the GEARS pipeline, including selection of the Top-5,000 variable genes and log-normalization of expression counts (as configured by the GEARS dataset loader).[9] After processing, the dataset contained 141,555 total cells and 28% were held out for test/val splits. For these cells, expression levels were available for 5,000 total genes and had perturbations targetting 1,087 unique genes.

# 5. Training Dynamics and Optimization

## 5.1 Self-Supervised Pretraining

The pretraining objective predicts masked teacher latents from student context latents:

$$\mathcal{L}_{\text{JEPA}} = \mathbb{E}\left[\|z_t - \hat{z}_t\|_1\right] \tag{5}$$

We utilize VICReg regularization to discourage collapse[6]. The pretraining updates both the student and teacher.[14],[15]. The objective is to learn robust token embeddings and cellular representations without perturbation labels to build a latent space representative of cell systems.

For training, for a given cell, 60% of the gene tokens are masked. The student encoder processes the masked input, while the Teacher encoder processes the full input. When comparing the generated latent representation, we leverage VICReg (Variance-Invariance-Covariance Regularization) to prevent mode collapse. The student is then updated via backprop. The teacher's weights are updated from the student via exponential moving average based on a specified momentum set to 0.996.

### 5.1.1 The VICReg Loss Function

The pretraining loss $\mathcal{L}$ is a weighted sum of three loss terms.

$$\mathcal{L} = \lambda \mathcal{L}_{inv} + \mu \mathcal{L}_{var} + \nu \mathcal{L}_{cov} \tag{6}$$

1. **Invariance ($\mathcal{L}_{inv}$):** Minimizes the Mean Squared Error (MSE) between the Student's predicted context latent and the Teacher's target latent. This forces the model to capture the semantic content of the cell state.

$$\mathcal{L}_{inv} = \frac{1}{N} \sum_{j=1}^{N} \left\| \hat{Z}_j - Z_j \right\|_2^2 \tag{7}$$

2. **Variance ($\mathcal{L}_{var}$):** A hinge loss that forces the standard deviation of the embeddings along the batch dimension to be at least 1. This prevents hypersphere collapse where the model maps all inputs to a single constant vector.

$$\mathcal{L}_{var} = \frac{1}{d} \sum_{j=1}^{d} \max(0, 1 - \sqrt{\mathrm{Var}(Z_{:,j}) + \epsilon}) \tag{8}$$

3. **Covariance ($\mathcal{L}_{cov}$):** Penalizes the off-diagonal elements of the covariance matrix $C(Z)$. This forces each dimension of the latent vector to encode unique, independent information, maximizing the information density of the representation.

$$\mathcal{L}_{cov} = \frac{1}{d} \sum_{j \neq k} [C(Z)]_{j,k}^2 \tag{9}$$

The current implementation uses $\lambda = 25, \mu = 25, \nu = 1$):

## 5.2 Perturbation Embedding Training

The Action Composer is trained to ensure multi-modal consistency. Contrastive learning through InfoNCE-style objective is used to align action embeddings across modalities (e.g., anchor/positive action pairs)[7]. In this process positive pairs are constructed from different descriptions of the same perturbation (e.g., the DNA sequence of a guide RNA and the protein sequence of its target gene)[7]. The model minimizes the InfoNCE loss to pull the embeddings of corresponding DNA and Protein representations together. This ensures that the predictor receives a consistent "action" vector regardless of whether the input is defined by sequence or protein ID.

$$\mathcal{L}_{\text{InfoNCE}}(q \to k) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp\left(s(q_i, k_i)/\tau\right)}{\sum_{j=1}^{N} \exp\left(s(q_i, k_j)/\tau\right)} \tag{10}$$

For training, we use a temperature $\tau = 0.07$.

## 5.3 Action-Conditioned Training

After pretraining and perturbation training, the Action-Conditioned predictor is trained to learn how to shift control latents and to  the perturbed latents based on the action embeddings using masked negative log likelihood (NLL) loss:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \frac{1}{2} \left( \log \sigma_i^2 + \frac{\|z_{t,i} - \mu_i\|_2^2}{\sigma_i^2} \right) \tag{11}$$

The paired (Control, Perturbation, Case) data passes through the full model. The Student receives the control cell state. The Teacher receives the perturbed cell state. The action composer receives the perturbation representation. The action-conditioned predictor receives the control latent and the pertrubation latent to then generate a perturbed cell latent. Crucially, the student and teacher encoders are frozen during this phase. This treats the encoders as a fixed feature extractor, preserving the general manifold structure learned during pretraining and preventing overfitting to the smaller perturbation dataset. To calculate loss we use combination of reconstruction loss (NLL on masked gene latents) and VICReg consistency loss on the difference between the teacher's perturbed cell latent and the ACpredictor's perturbed cell latent.

$$\mathcal{L}_{AC} = \lambda_{sim} \mathcal{L}_{NLL} + \mathcal{L}_{VICReg} \tag{12}$$

# 6. Evaluation and Results

## 6.1 Evaluation Protocol

Since BioJEPA-AC is an encoder and trains on comparing latents, the evaluation pipeline trains a lightweight evaluation head that decodes gene-level expression and reports metrics comparing predicted vs. observed perturbation effects. Standard metrics like global Mean Squared Error (MSE) can be misleading in single-cell perturbation settings because many genes exhibit small or no changes for a given perturbation. As a result, an identity-style predictor can achieve low global MSE while failing to capture causal perturbation effects. For this reason, the report includes metrics beyond global MSE.

1. **Global Mean Squared Error (MSE):** The average squared error between the predicted post-perturbation expression $\hat{x}_{pert}$ and the true post-perturbation expression $x_{pert}$ across genes and samples (in the evaluation head's output space).
2. **Pearson Delta Correlation ($\rho_\Delta$):** The Pearson correlation between the *predicted change* $\Delta\hat{x} = \hat{x}_{pert} - x_{ctrl}$ and the *true change* $\Delta x = x_{pert} - x_{ctrl}$. In this report, the correlation is computed on the Top-20 differentially expressed genes (DEGs) to emphasize perturbation-driven signal.
3. $R^2$ **(Coefficient of Determination):** The coefficient of determination between $\hat{x}_{pert}$ and $x_{pert}$. The report summarizes this metric with mean and median over (i) all genes and (ii) the Top-50 DEGs.
4. **Predicted Shift Magnitude:** A summary of how large the model's predicted perturbation shift is on

average, defined as $\mathbb{E}[|\Delta\hat{x}|] = \mathbb{E}[|\hat{x}_{pert} - x_{ctrl}|]$. This helps detect degenerate solutions that minimize error by predicting overly small shifts.

## 6.2 Perturbation Expression Impact Evaluation on the K562 Essential Dataset

> To calculate gene-level metrics (e.g., gene-wise correlations), a lightweight decoder is trained on top of the frozen latent representations. The decoder is a secondary source of error that is distinct from the BioJEPA-AC model itself.

The model is evaluated on the K562 Essential dataset [1] using held-out perturbations defined by GEARS-style splits.[9] The perturbation encoder is trained with sgRNA sequence embeddings and target-gene embeddings.

1. **Global MSE = 0.4979:** A value below 0.5 indicates the model's reconstructed cell states are mathematically very close to the ground truth expression manifold.
2. **Pearson R (Top 20 DEG) = 0.9266:** This confirms that the model flawlessly preserves the fundamental cellular identity and viability signals.
3. $R^2$ **(All Genes) = Mean 0.9175 | Median 0.9272:** Capturing >91% of the variance implies the model has essentially solved the baseline state of the cell and understands general gene-gene correlations.
4. $R^2$ **(Top 50 DEGs)= Mean 0.0962 | Median 0.3211:** The positive median proves the model is successfully predicting the causal effects of the drug/CRISPR, distinguishing signal from noise.
5. **Mean Pred Shift Magnitude = 0.5379:** A value of ~0.54 confirms the model is confident enough to predict significant biological changes rather than conservatively predicting near-zero updates (posterior collapse). Note that these numbers are "log" so a value of 0.5 represents half an order of magnitude.

These metrics are reported in the same evaluation space used to train and apply the evaluation head. As noted above, gene-level results depend on both the frozen BioJEPA-AC latents and the separately trained decoder. While some of these numbers may seem high compared to comparable models in the space, note that this model has been trained on a single dataset and not yet generalized across many different datasets.

## 6.3 Items Not Present For Eval Completeness

The the current release do not include:

- Evaluate BioJEPA-AC against standard baselines (e.g., CPA [8], GEARS [9], scGPT [10], Geneformer [11]) using the same GEARS splits and preprocessing.
- Evaluate generalization to settings such as unseen target genes, unseen guides for known targets, and combinatorial perturbations (if available in the benchmark).
- Test transfer across perturbation datasets or across cell types/cell lines to assess robustness beyond a single benchmark.
- Robustness checks across alternative gene sets and model scales.

# 7. Future Directions: In Silico Biology

1. Expand our validation suite to other tasks to better represent the flexibility of our model.
2. Expand our "unseen" validation to fully excluded cell lines and perturbations to highlight the generalization capabilities of the model.
3. Expand our dataset to include different cell lines, different perturbation types/modalities, and more cell state measurements.

For questions, comments, and inquiries, reach out to gptomics@gmail.com

# Works cited


[1] Replogle, J. M., Saunders, R. A., Pogson, A. N., et al. (2022). *Mapping information-rich genotype–phenotype landscapes with genome-scale Perturb-seq.* Cell. DOI: 10.1016/j.cell.2022.05.013.

[2] LeCun, Y. (2022). *A Path Towards Autonomous Machine Intelligence.* arXiv:2205.12723.

[3] *PerturbNet predicts single-cell responses to unseen chemical and genetic perturbations.* Molecular Systems Biology (2025). DOI: 10.1038/s44320-025-00131-3.

[4] *Scouter predicts transcriptional responses to genetic perturbations with large language model embeddings.* Nature Computational Science (2025). DOI: 10.1038/s43588-025-00912-8.

[5] Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Komeili, M., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al. (2025). *V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning*. *arXiv preprint arXiv:2506.09985*.

[6] Bardes, A., Ponce, J., & LeCun, Y. (2022). VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. *arXiv preprint arXiv:2105.04906*.

[7] van den Oord, A., Li, Y., & Vinyals, O. (2018). *Representation Learning with Contrastive Predictive Coding.* arXiv:1807.03748.

[8] Lotfollahi M, Klimovskaia Susmelj A, De Donno C, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Mol Syst Biol*. 2023;19(6):e11517. doi:10.15252/msb.202211517

[9] *Predicting transcriptional outcomes of novel multigene perturbations with graph neural networks (GEARS).* Nature Biotechnology (2024). DOI: 10.1038/s41587-023-01905-6.

[10] *scGPT: Towards building a foundation model for single-cell multi-omics using generative pre-training.* Nature Methods (2024). DOI: 10.1038/s41592-024-02201-0.

[11] Theodoris, C. V., Xiao, L., Chopra, A., et al. (2023). *Transfer learning enables predictions in network biology.* Nature. DOI: 10.1038/s41586-023-06139-9.

[12] Adduri, A. K., Gautam, D., Bevilacqua, B., Imran, A., Shah, R., Naghipourfar, M., Teyssier, N., Ilango, R., Nagaraj, S., Dong, M., Ricci-Tam, C., Carpenter, C., Subramanyam, V., Winters, A., Tirukkovular, S., Sullivan, J., Plosky, B. S., Eraslan, B., Youngblut, N. D., Leskovec, J., Gilbert, L. A., Konermann, S., Hsu, P. D., Dobin, A., Burke, D. P., Goodarzi, H., & Roohani, Y. H. (2025). Predicting cellular responses to perturbation across diverse contexts with State. *bioRxiv*. DOI: 10.1101/2025.06.26.661135.

[13] Dong, M., Adduri, A., Gautam, D., Carpenter, C., Shah, R., Ricci-Tam, C., Kluger, Y., Burke, D. P., & Roohani, Y. H. (2026). *Stack: In-Context Learning of Single-Cell Biology*. *bioRxiv*. DOI: 10.64898/2026.01.09.698608.

[14] Grill, J.-B., Strub, F., Altché, F., et al. (2020). *Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning.* arXiv:2006.07733.

[15] He, K., Chen, X., Xie, S., et al. (2021). *Masked Autoencoders Are Scalable Vision Learners.* arXiv:2111.06377.

[16] Katharopoulos, A., Vyas, A., Pappas, N., & Fleuret, F. (2020). *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention.* arXiv:2006.16236.

[17] Perez, E., Strub, F., de Vries, H., Dumoulin, V., & Courville, A. (2017). *FiLM: Visual Reasoning with a General Conditioning Layer.* arXiv:1709.07871.

[18] *The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics.* Nature Methods (2024). DOI: 10.1038/s41592-024-02523-z.

[19] *Evolutionary-scale prediction of atomic-level protein structure with a language model.* Science (2023). DOI: 10.1126/science.ade2574.