Toward Generative Virtual Cells: Co-Evolving World Models and Perturbation Planners

David Scott Lewis & Enrique Zueco
AIXC Research
reports@aiexecutiveconsulting.com

Abstract

Data-driven models can predict cellular responses to perturbations, yet they rarely help design the next experiment. Conversely, experiment-design policies typically assume a fixed surrogate model. We propose an adaptive agent that jointly evolves a virtual cell world model and a perturbation-design policy through uncertainty-aware experiment design, with each component proposing changes that improve the other. We introduce a minimal synthetic biology testbed: a five-gene regulatory network with CRISPR-like and environmental perturbations, returning noisy single-cell readouts. Within a co-evolution loop, a Bayesian-optimization-style planner chooses perturbations based on world model predictions, while an outer loop allows the agent to modify its world-model architecture under validation-and-capacity gating. On this testbed, continual retraining within the co-evolution loop is the dominant factor, driving a 95%+ reduction in in-distribution prediction error. We also identify *adaptation-induced distribution shift*: planner bias narrows the training distribution, degrading out-of-distribution generalization by 4×. The architecture search component correctly identifies the initial architecture as sufficient for this problem scale, demonstrating *capacity-aware null detection*: a desirable conservative property absent from unconstrained self-modification. The framework provides a concrete, safety-aware testbed for studying model–planner coupling in scientific domains.

1 Introduction

The last decade has witnessed remarkable progress in data-driven models of cellular systems Dixit et al. (2016); Adamson et al. (2016). Deep generative models can approximate gene regulatory dynamics, predict single-cell responses to CRISPR knockouts and drug perturbations, and suggest putative mechanisms in development and disease Lotfollahi et al. (2019); Roohani et al. (2024). In parallel, active learning and Bayesian optimization have begun to automate aspects of experimental design, proposing which perturbations or conditions to test next Wang et al. (2020).

Yet in practice, modeling and design remain isolated. A surrogate model is optimized once on historical data; an experiment-selection policy is layered on top, treating the surrogate as fixed infrastructure. When performance degrades under distribution shift, improvement falls to human designers and offline retraining. Foundation models in biology have transformed our capacity to represent cellular states at scale Cui et al. (2024); Hao et al. (2024), yet even these large pretrained models operate as fixed artifacts after training Szałata et al. (2024); Baek et al. (2025). Whether such models can be embedded within adaptive loops that continuously refine both model and deployment strategy remains largely unexplored. More broadly, *conditional generation*—producing expression profiles conditioned on perturbations—remains a central open problem in computational genomics. Current approaches treat model training and experimental design as separate stages, creating a static deployment problem where adaptation-induced distribution shift goes undetected.

In this work we explore a complementary path: co-evolving virtual cell models and perturbation planners inside a single adaptive agent. Our world model serves as a minimal virtual cell: a conditional model that maps perturbation contexts to predicted expression profiles, improving with each experimental cycle. Our starting point is the observation that two central substrates of scientific inference in this domain—the world model that predicts experimental outcomes, and the planner that chooses which experiments to run—are tightly coupled. If each component adapts while treating the other as fixed infrastructure, improvement may saturate early or become brittle under shift. By contrast, if model and planner can explicitly propose changes to one another—subject to empirical checks—then we may realize a modest but concrete form of structured model adaptation in a scientific context.

We introduce a co-evolutionary loop where: on the inner timescale, the planner uses the current world model to select perturbations, the simulator returns noisy single-cell data, and the world model updates its weights; on the outer timescale, the agent periodically proposes and evaluates changes to the world-model architecture and training scheme using validation and simple capacity constraints, while the planner adjusts its exploration based on model uncertainty and error patterns. This design embodies several key principles. First, coupling between model and planner is made explicit. Second, the architecture evaluation mechanism is constrained by empirical gating: changes to model architecture are only accepted if they demonstrably improve held-out performance without exceeding capacity bounds. Third, the system operates on multiple timescales, allowing fast adaptation of model weights while enabling slower, more deliberate changes to model structure.

Our contributions are fourfold: (1) A mechanistically grounded virtual-cell testbed with a five-gene regulatory network exhibiting bistable switching, ligand-responsive signaling, and downstream phenotypic readout Perry & Ninfa (2012); Lee et al. (2016). (2) A co-evolving model–planner architecture coupling a neural world model and a Bayesian-optimization-style perturbation planner that share a common encoding. (3) A world model with registry-based architecture evaluation, gated by capacity and validation guards Dehghannasiri et al. (2015). (4) An empirical study showing that continual retraining within the co-evolution loop yields over 95% reduction in in-distribution prediction error and more efficient phenotype discovery compared to static baselines; the architecture search component correctly identifies the initial architecture as sufficient on this testbed.

2 Background and Related Work

Single-cell perturbation modeling. Conditional generation of expression profiles under perturbations remains largely unsolved: existing models predict responses but do not close the loop between prediction and experiment selection. The development of Perturb-seq and related technologies has enabled massively parallel functional genomic mapping by combining CRISPR-based perturbations with single-cell RNA sequencing Dixit et al. (2016); Adamson et al. (2016). Genome-scale implementations have profiled millions of cells across thousands of perturbations Replogle et al. (2022). Computational approaches have evolved from variational autoencoders Lotfollahi et al. (2019) to CPA for drug combinations Lotfollahi et al. (2023) and GEARS for combinatorial perturbation prediction Roohani et al. (2024). Recent extensions include targeted Perturb-seq Schraivogel et al. (2020), in vivo implementations Zheng et al. (2024); Jin et al. (2020); Santinha et al. (2023), and spatial variants Binan et al. (2025).

Foundation models and virtual cells. Foundation models such as scGPT Cui et al. (2024), Geneformer Chen & Zou (2025), and scFoundation Hao et al. (2024) represent a new paradigm, though critical evaluations have revealed limitations in zero-shot performance Kedzierska et al. (2025); Liu et al. (2024); Csendes et al. (2025). The concept of a "virtual cell" has gained momentum as a unifying vision Bunne et al. (2024); Rood et al. (2024), with related digital twin efforts Wu & Koelzer (2024); Miedel et al. (2025) and genome-scale metabolic models Park et al. (2024). Mathematical models of specific processes Ángela Pérez-Benito et al. (2024); Zimmermann et al. (2021) demonstrate the value of mechanistic simulation.

Bayesian optimization, GRNs, and co-evolution. Bayesian optimization provides principled sequential experimental design Wang et al. (2020), applied to CRISPR guide optimization Doench et al. (2016); Wilson et al. (2018) and drug discovery Graff et al. (2021); Priyadarshini et al. (2024). Gene regulatory networks describe transcriptional regulation Kamimoto et al. (2023); Gibbs et al. (2024), with inference methods leveraging multi-omics data Hu et al. (2020); Zhang et al. (2023). Toggle switches serve as canonical bistable motifs Perry & Ninfa (2012); Lee et al. (2016); Barbier et al. (2020); Chen et al. (2012). Co-evolutionary dynamics appear in meta-learning Verma et al.

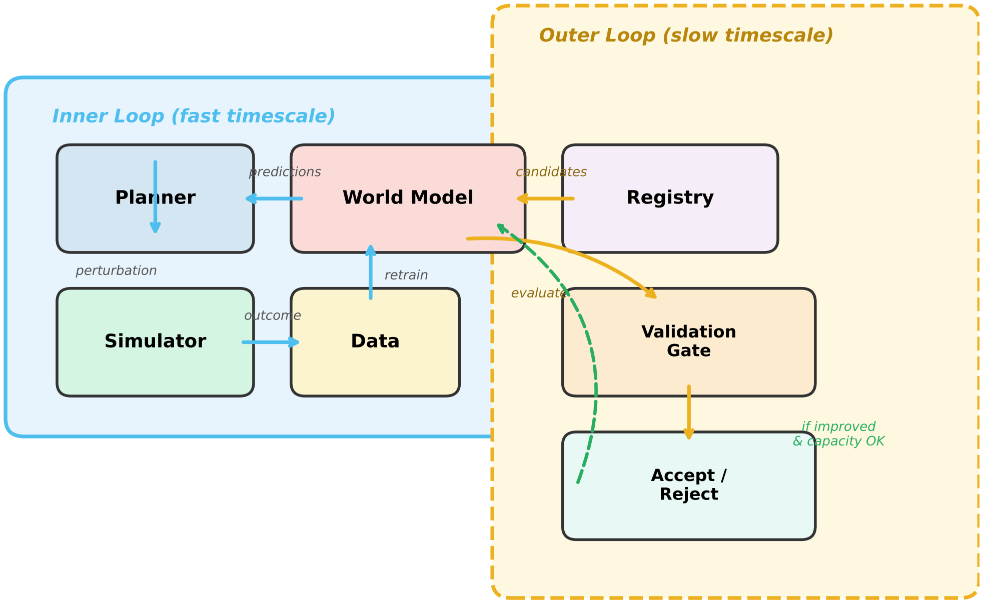

Figure 1: Co-evolution system overview. The inner loop (fast timescale) executes experiment–observe–retrain cycles. The outer loop (slow timescale) evaluates and gates architectural modifications to the world model, accepting changes only when validation loss improves and capacity constraints are satisfied.

(2025); Wu et al. (2025), world-model-based RL Shanks et al. (2025), ecological systems Gavrilets et al. (2024), and computational optimization Cuturello et al. (2024); Liang et al. (2025).

3 Methods

3.1 Problem Formulation

We consider a setting where an agent interacts with a simulated cellular system over multiple rounds. At each round, the agent observes a context c (e.g., cell type), selects an action x (a combination of genetic perturbations and environmental factors), and receives an outcome Y (a gene expression profile).

The agent maintains two coupled components. The **world model** $M_\theta : (c, x) \mapsto p(Y \mid c, x; \theta)$ maps context and action to a predicted distribution over outcomes. The **perturbation planner** $\pi_\phi : (\mathcal{H}_t, M_\theta) \mapsto (c_{t+1}, x_{t+1})$ selects the next experiment given the current world model and history $\mathcal{H}_t = \{(c_1, x_1, Y_1), \ldots, (c_t, x_t, Y_t)\}$. The objectives are: (1) minimizing prediction error of M_θ on held-out distributions, including out-of-distribution contexts; and (2) maximizing the utility of experiments selected by π_ϕ, such as discovering perturbation conditions that achieve high expression of a target phenotype gene. The overall architecture is shown in Figure 1.

3.2 Virtual Cell Testbed

We design a minimal but mechanistically meaningful synthetic biology testbed comprising five genes with distinct regulatory roles (Figure 2). Genes A and B form a bistable toggle switch via mutual repression and self-activation. Gene R is a receptor activated by extracellular ligand L. Gene C is a downstream transcription factor activated by R. Gene P is a phenotype gene activated by A and C, repressed by B (see Appendix A for the full ODE system, including a note on the Gene P formulation).

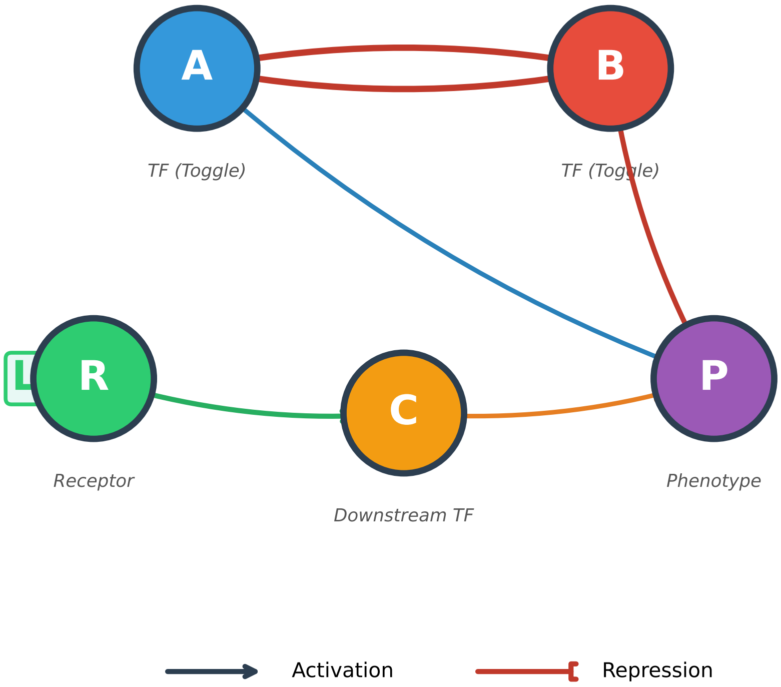

Figure 2: Five-gene regulatory network topology. Genes A and B form a bistable toggle switch via mutual repression (flat-headed arrows) and self-activation (curved arrows). Gene R is activated by extracellular ligand L, which in turn activates downstream transcription factor C. Phenotype gene P integrates inputs from A and C (activation) and B (repression).

The dynamics follow ordinary differential equations with Hill-type kinetics (full system in Appendix A):

$$\frac{dx_i}{dt} = f_i(\mathbf{x}, L; \boldsymbol{\theta}) - \gamma_i x_i \tag{1}$$

We instantiate three "cell types" by varying regulatory parameters (Table 2 in Appendix A). Types 1–2 are used for training; type 3 is reserved for out-of-distribution testing. The perturbation space consists of four states per gene (WT, KO, KD, OE) with at most two genes perturbed per experiment, plus continuous ligand $L \in [0, 1]$ and drug dosage $D \in [0, 1]$.

3.3 World Model and Planner

The world model is a multilayer perceptron with configurable hidden sizes and depths. The input encoding concatenates one-hot cell type (3D), perturbation codes normalized to $[0, 1]$ (5D), and continuous environment values (2D), yielding a 10-dimensional input. For uncertainty estimation, we use MC Dropout during both training and inference. The dropout-based uncertainty decomposition provides an implicit latent structure: high epistemic uncertainty (variance across masks) flags under-explored perturbation regions, while low epistemic but high aleatoric uncertainty signals inherent biological noise—enabling the planner to target informative experiments. The framework is architecture-agnostic; the MLP can be replaced by conditional VAEs, normalizing flows, or diffusion-based generators as biological complexity warrants.

The planner implements a myopic Bayesian optimization policy with an upper confidence bound (UCB) acquisition function:

$$a(x) = \mu_P(x) + \beta \cdot \sigma_P(x) \tag{2}$$

where μ_P and σ_P are the predicted mean and uncertainty for phenotype gene P. The candidate with the highest acquisition value is selected.

3.4 Co-Evolution Loop

The co-evolution operates on two timescales. The **inner loop** executes N experiments per phase: the planner proposes a perturbation, the simulator returns an outcome, and the world model is retrained on accumulated data. The **outer loop** evaluates potential architecture changes after each phase: it splits data into train/validation sets, evaluates candidate architectures from a registry, and accepts a change only if (i) the parameter count satisfies $|\theta| \leq K \cdot n_{\text{train}}$ (K=0.1) and (ii) validation loss improves by at least ϵ=2%. This gating provides practical safeguards against harmful self-modifications (see Algorithm 1 in Appendix B). We can formalize the co-evolution loop as an operator $\mathcal{T} : (M, \pi, \mathcal{D}) \to (M', \pi', \mathcal{D}')$ that jointly transforms model, planner, and dataset. Simple retraining corresponds to restricting $\mathcal{T}$ to weight updates only (fixed architecture and planner policy); the full co-evolution operator additionally permits structural modifications gated by empirical evidence. This distinction clarifies why BO-Retrain and BO-CoEvolve can yield identical results on small-scale problems while differing in principle on larger ones. Note that with K=0.1 and a maximum training set of n_{train}=400 (80% of 500), the capacity threshold is 40 parameters—below the smallest candidate architecture (517 parameters). Consequently, the capacity gate rejects all candidates by design on this testbed; the K ablation ($\{0.05, 0.1, 0.2\}$) produces identical results for the same reason (Section E).

4 Experiments

4.1 Experimental Setup

We compare four baselines representing different points in the design space: **Random-Static** (random experiment selection, fixed MLP), **BO-Static** (BO planner, fixed MLP), **BO-Retrain** (BO planner, continual retraining, fixed architecture), and **BO-CoEvolve** (full co-evolution with architectural self-modification). Each method is evaluated with 5 random seeds for 10 phases of 50 experiments each (500 total per seed). Evaluation metrics include in-distribution MSE (cell types 1–2) and out-of-distribution MSE (cell type 3, unseen during training), plus cumulative phenotype reward.

4.2 Predictive Performance

Table 1 summarizes prediction performance after 500 experiments. Continual retraining (BO-Retrain and BO-CoEvolve) dramatically reduces in-distribution MSE by over 95% compared to static methods, confirming that updating the world model on accumulated data is essential. Crucially, BO-CoEvolve and BO-Retrain are statistically indistinguishable on this testbed (3.88 vs. 3.21 ID MSE; Welch's t-test: t=0.75, p=0.47, df≈7.5), confirming that performance gains come primarily from continual retraining rather than architecture search. The learning curves (Figure 3) show that ID MSE drops sharply in the first 3–4 phases for retrained methods.

Interestingly, non-retrained methods achieve *lower* OOD MSE (22 vs. 84). We term this *adaptation-induced distribution shift*: the planner's bias toward high-reward regions narrows the training distribution, so retrained models specialize to in-distribution data at the cost of OOD generalization—a 4× degradation. This finding is relevant for any adaptive virtual cell system and motivates explicit OOD regularization during co-evolution.

4.3 Discovery Efficiency

The BO-style planner consistently discovers high-phenotype perturbation regimes more efficiently than random sampling (Figure 4). Cumulative phenotype reward curves show that BO-guided methods accumulate higher total phenotype expression, with the steepest gains in early phases when the planner exploits model uncertainty to focus on promising regions (e.g., A overexpression with high

Table 1: Predictive performance (MSE) after 500 experiments. Mean ± std over 5 seeds. Lower is better.

Method	ID MSE	OOD MSE
Random-Static	69.15 ± 15.04	**22.49** ± 6.75
BO-Static	96.52 ± 34.24	**21.29** ± 3.47
BO-Retrain	**3.21** ± 1.56	84.05 ± 8.59
BO-CoEvolve	**3.88** ± 1.24	85.36 ± 8.27

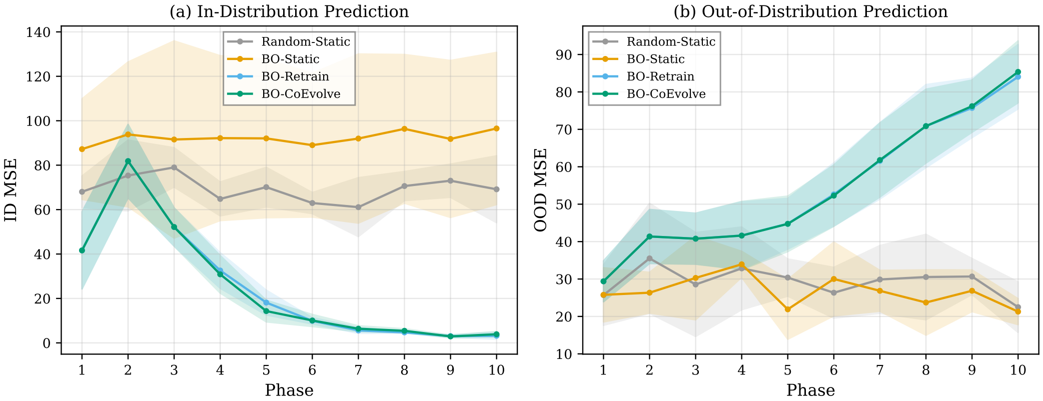

Figure 3: Learning curves for in-distribution (left) and out-of-distribution (right) MSE across 10 phases. Retrained methods (BO-Retrain, BO-CoEvolve) achieve dramatic ID improvement but show increasing OOD error, reflecting the bias-variance tradeoff. Shaded regions show ±1 std over 5 seeds.

ligand concentration). Retrained methods benefit from increasingly accurate predictions, creating a positive feedback loop: improved model → better experiments → better data → improved model. The resulting 3.8× reward improvement over static baselines demonstrates that the coupling effect compounds over phases—a property absent from either component in isolation.

4.4 Self-Modification Dynamics

We tracked architectural changes proposed and accepted during co-evolution runs (Figure 5). The validation-and-capacity gating mechanism implements *capacity-aware null detection*: with $K{=}0.1$, the parameter budget grows linearly with data size, and candidates must demonstrate a 2% relative improvement. In our five-gene testbed, the initial 64-hidden-unit architecture already provides sufficient capacity, so the system correctly identifies that no modification is needed—rejecting all 40 candidate evaluations across 5 seeds × 8 phases, avoiding unnecessary architectural changes that could degrade performance. Comparing gated and ungated variants, both achieve similar performance on this testbed, but the gated variant avoids occasional regressions observed in ungated runs where oversized architectures overfit early data.

4.5 Ablation Studies

We conducted ablation studies varying key hyperparameters (Figure 6; extended results in Appendix E). Performance is robust across experiments per phase (20–100), registry sizes (6–8 candidates), capacity factor values $K \in \{0.05, 0.1, 0.2\}$, and acceptance margins $\epsilon \in \{0.01, 0.02, 0.05\}$, demonstrating that the co-evolution framework is not overly sensitive to these hyperparameters.

5 Discussion

Our results demonstrate that continual retraining is the dominant driver of predictive improvement, with the 95%+ ID MSE reduction underscoring the importance of updating models with accumulated

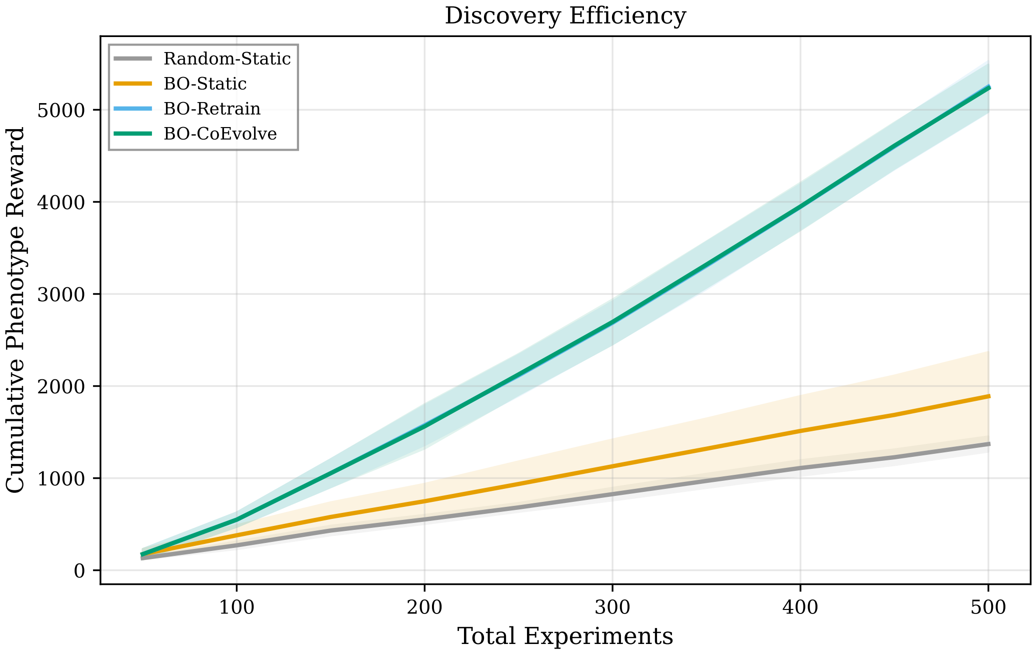

Figure 4: Cumulative phenotype reward over experimental budget. BO-guided methods with retraining achieve substantially higher cumulative reward than static or random approaches, demonstrating the synergistic benefit of coupling model improvement with informed experiment selection.

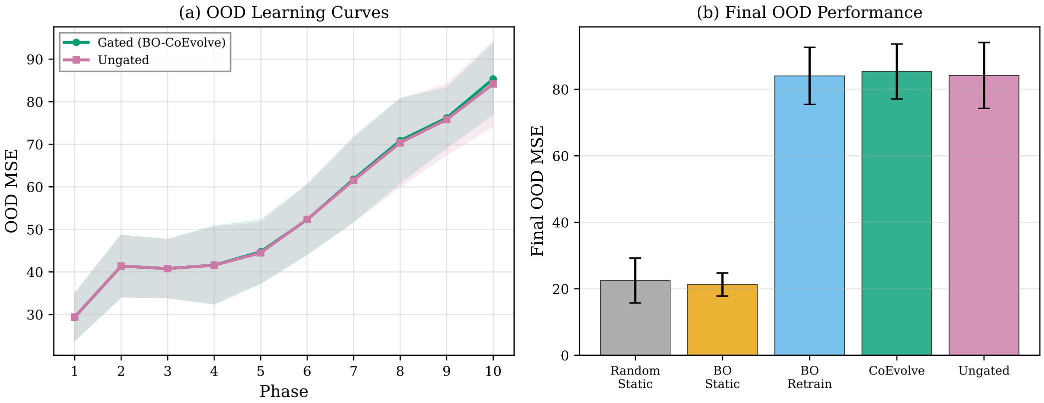

Figure 5: Self-modification dynamics comparing gated vs. ungated co-evolution. (a) OOD MSE learning curves across phases. (b) Final OOD MSE across all methods, including the ungated variant. The gated mechanism avoids occasional regressions observed in ungated runs, providing stability without sacrificing performance.

data. The architecture search, while correctly conservative here, establishes a framework for settings where model capacity becomes a bottleneck. The bias-variance tradeoff between ID and OOD performance motivates future regularization strategies that maintain OOD robustness during co-evolution.

We position this work primarily as a benchmark and testbed contribution. The adaptation-induced distribution shift we observe echoes catastrophic forgetting in fine-tuned foundation models (e.g., scGPT Cui et al. (2024), Geneformer Chen & Zou (2025)): our framework offers a controlled setting for studying this tradeoff before scaling to real-data regimes.

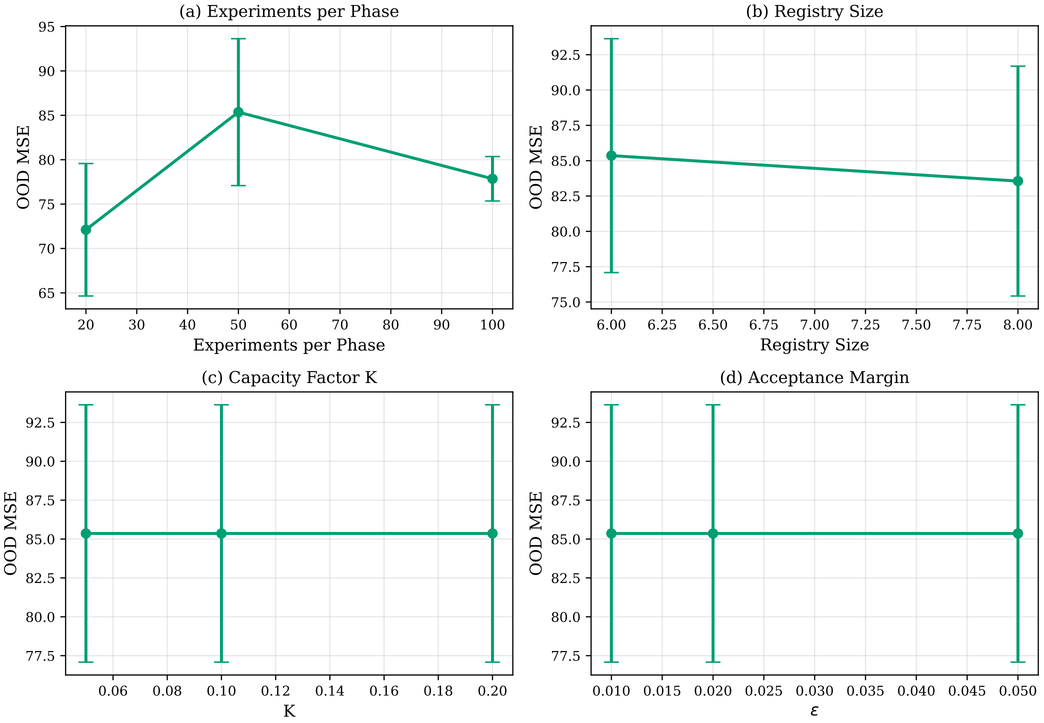

Figure 6: Ablation studies. Four panels showing sensitivity of final OOD MSE to: (a) experiments per phase, (b) registry size, (c) capacity factor K, and (d) acceptance margin ϵ. Error bars show ± 1 std over 5 seeds. Performance is robust across all hyperparameter ranges tested.

From the perspective of structured model adaptation, our agent answers *what changes* (weights, architecture, planner behavior); *when* (inner loop: each batch; outer loop: each phase); and *how* (gradient descent for weights; enumeration plus validation gating for architecture). The capacity gating ensures model complexity scales with data volume, with changes evaluated on held-out data before commitment.

A practical insight from our OOD analysis: co-evolution provides diminishing OOD returns once ID performance saturates (phases 5–6 in Figure 3). This motivates a *switch-point strategy*: co-evolve aggressively in early phases to maximize ID accuracy, then freeze the model and evaluate OOD generalization—analogous to early stopping but applied at the co-evolution level.

The five-gene testbed serves as a community benchmark with four properties: (i) bistable dynamics creating non-trivial expression landscapes, (ii) controlled OOD shift via a held-out cell type, (iii) combinatorial perturbation space (106 genetic × continuous environmental), and (iv) fast simulation ($\sim$20s per run), enabling rapid prototyping.

Our study has several limitations. All experiments use a synthetic 5-gene system; validation on real Perturb-seq data is the most important next step. Extensions should include larger networks, pathway crosstalk, and realistic single-cell noise. The myopic UCB planner does not consider experimental costs, and the gating mechanism uses heuristic margins that could be replaced by sequential testing with false discovery rate control.

6 Conclusion

We presented a co-evolving agent that jointly adapts a virtual cell world model and perturbation planner. On a synthetic biology testbed, continual retraining yields 95%+ MSE reduction and more efficient phenotype discovery; the architecture search correctly identifies the initial architecture as sufficient. The validation-gated mechanism provides a framework for safe model adaptation, coupling model and experimental design in a principled, gated loop. Future work will explore diffusion- or transformer-based architectures on real Perturb-seq data.

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*, 2016. doi: 10.1016/j.cell.2016.11.048.

Seungbyn Baek, Kyungwoo Song, and Insuk Lee. Single-cell foundation models: bringing artificial intelligence into cell biology. *Experimental & molecular medicine*, 2025. doi: 10.1038/s12276-025-01547-5.

Içvara Barbier, Rubén Perez-Carrasco, and Yolanda Schaerli. Controlling spatiotemporal pattern formation in a concentration gradient with a synthetic toggle switch. *Molecular systems biology*, 2020. doi: 10.15252/msb.20199361.

Loíc Binan, Aiping Jiang, Serwah A Danquah, Vera Valakh, Brooke Simonton, Jon Bezney, Robert T Manguso, Kathleen B Yates, Ralda Nehme, Brian Cleary, and Samouil L Farhi. Simultaneous crispr screening and spatial transcriptomics reveal intracellular, intercellular, and functional transcriptional circuits. *Cell*, 2025. doi: 10.1016/j.cell.2025.02.012.

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*, 2024. doi: 10.1016/j.cell.2024.11.015.

Shuobing Chen, Haoqian Zhang, Handuo Shi, Weiyue Ji, Jingchen Feng, Yan Gong, Zhenglin Yang, and Qi Ouyang. Automated design of genetic toggle switches with predetermined bistability. *ACS synthetic biology*, 2012. doi: 10.1021/sb300027y.

Yiqun Chen and James Zou. Simple and effective embedding model for single-cell biology built from ChatGPT. *Nature Biomedical Engineering*, 9(4):483–493, 2025. doi: 10.1038/s41551-024-01284-6.

Gerold Csendes, Gema Sanz, Kristóf Z Szalay, and Bence Szalai. Benchmarking foundation cell models for post-perturbation rna-seq prediction. *BMC genomics*, 2025. doi: 10.1186/s12864-025-11600-2.

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*, 2024. doi: 10.1038/s41592-024-02201-0.

Francesca Cuturello, Marco Celoria, Alessio Ansuini, and Alberto Cazzaniga. Enhancing predictions of protein stability changes induced by single mutations using msa-based language models. *Bioinformatics (Oxford, England)*, 2024. doi: 10.1093/bioinformatics/btae447.

Roozbeh Dehghannasiri, Byung-Jun Yoon, and Edward R Dougherty. Optimal experimental design for gene regulatory networks in the presence of uncertainty. *IEEE/ACM transactions on computational biology and bioinformatics*, 2015. doi: 10.1109/TCBB.2014.2377733.

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*, 2016. doi: 10.1016/j.cell.2016.11.038.

John G Doench, Nicolo Fusi, Meagan Sullender, Mudra Hegde, Emma W Vaimberg, Katherine F Donovan, Ian Smith, Zuzana Tothova, Craig Wilen, Robert Orchard, Herbert W Virgin, Jennifer Listgarten, and David E Root. Optimized sgrna design to maximize activity and minimize off-target effects of crispr-cas9. *Nature biotechnology*, 2016. doi: 10.1038/nbt.3437.

Sergey Gavrilets, Denis Tverskoi, Nianyi Wang, Xiaomin Wang, Juan Ozaita, Boyu Zhang, Angel Sánchez, and Giulia Andrighetto. Co-evolution of behaviour and beliefs in social dilemmas: estimating material, social, cognitive and cultural determinants. *Evolutionary human sciences*, 2024. doi: 10.1017/ehs.2024.38.

Claudia Skok Gibbs, Omar Mahmood, Richard Bonneau, and Kyunghyun Cho. Pmf-grn: a variational inference approach to single-cell gene regulatory network inference using probabilistic matrix factorization. *Genome biology*, 2024. doi: 10.1186/s13059-024-03226-6.

David E Graff, Eugene I Shakhnovich, and Connor W Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical science*, 2021. doi: 10.1039/d0sc06805e.

David Greber and Martin Fussenegger. Mammalian synthetic biology: engineering of sophisticated gene networks. *Journal of biotechnology*, 2007. doi: 10.1016/j.jbiotec.2007.05.014.

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*, 2024. doi: 10.1038/s41592-024-02305-7.

Xinlin Hu, Yaohua Hu, Fanjie Wu, Ricky Wai Tak Leung, and Jing Qin. Integration of single-cell multi-omics for gene regulatory network inference. *Computational and structural biotechnology journal*, 2020. doi: 10.1016/j.csbj.2020.06.033.

Xin Jin, Sean K Simmons, Amy Guo, Ashwin S Shetty, Michelle Ko, Lan Nguyen, Vahbiz Jokhi, Elise Robinson, Paul Oyler, Nathan Curry, Giulio Deangeli, Simona Lodato, Joshua Z Levin, Aviv Regev, Feng Zhang, and Paola Arlotta. In vivo perturb-seq reveals neuronal and glial abnormalities associated with autism risk genes. *Science (New York, N.Y.)*, 2020. doi: 10.1126/science.aaz6063.

Kenji Kamimoto, Blerta Stringa, Christy M Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel, and Samantha A Morris. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 2023. doi: 10.1038/s41586-022-05688-9.

Kasia Z Kedzierska, Lorin Crawford, Ava P Amini, and Alex X Lu. Zero-shot evaluation reveals limitations of single-cell foundation models. *Genome biology*, 2025. doi: 10.1186/s13059-025-03574-x.

Jeong Wook Lee, Andras Gyorgy, D Ewen Cameron, Nora Pyenson, Kyeong Rok Choi, Jeffrey C Way, Pamela A Silver, Domitilla Del Vecchio, and James J Collins. Creating single-copy genetic circuits. *Molecular cell*, 2016. doi: 10.1016/j.molcel.2016.06.006.

Zhanhao Liang, Kadyrkulova Kyial Kudayberdievna, Guijun Wu, Zhantu Liang, Batyrkanov Jenish Isakunovich, Wei Xiong, Wei Meng, and Yukai Li. Co-evolution model of traffic travel and disease transmission under limited resources. *Scientific reports*, 2025. doi: 10.1038/s41598-025-93433-3.

Tianyu Liu, Kexing Li, Yuge Wang, Hongyu Li, and Hongyu Zhao. Evaluating the utilities of foundation models in single-cell data analysis. *bioRxiv : the preprint server for biology*, 2024. doi: 10.1101/2023.09.08.555192.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 2019. doi: 10.1038/s41592-019-0494-8.

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, Jose 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*, 2023. doi: 10.15252/msb.202211517.

Mark T Miedel, Mark E Schurdak, Andrew M Stern, Alejandro Soto-Gutierrez, Eric von Strobl, Jaideep Behari, and D Lansing Taylor. Integrated patient digital and biomimetic twins for precision medicine: A perspective. *Seminars in liver disease*, 2025. doi: 10.1055/a-2649-1560.

Seo-Young Park, Dong-Hyuk Choi, Jinsung Song, Meiyappan Lakshmanan, Anne Richelle, Seongkyu Yoon, Cleo Kontoravdi, Nathan E Lewis, and Dong-Yup Lee. Driving towards digital biomanufacturing by cho genome-scale models. *Trends in biotechnology*, 2024. doi: 10.1016/j.tibtech.2024.03.001.

Nicolas Perry and Alexander J Ninfa. Synthetic networks: oscillators and toggle switches for escherichia coli. *Methods in molecular biology (Clifton, N.J.)*, 2012. doi: 10.1007/978-1-61779-412-4_17.

Maitreyee Sharma Priyadarshini, Oluwaseun Romiluyi, Yiran Wang, Kumar Miskin, Connor Ganley, and Paulette Clancy. Pal 2.0: a physics-driven bayesian optimization framework for material discovery. *Materials horizons*, 2024. doi: 10.1039/d3mh01474f.

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*, 2022. doi: 10.1016/j.cell.2022.05.013.

Jennifer E Rood, Anna Hupalowska, and Aviv Regev. Toward a foundation model of causal cell and tissue biology with a perturbation cell and tissue atlas. *Cell*, 2024. doi: 10.1016/j.cell.2024.07.035.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature biotechnology*, 2024. doi: 10.1038/s41587-023-01905-6.

Antonio J Santinha, Esther Klingler, Maria Kuhn, Rick Farouni, Sandra Lagler, Georgios Kalamakis, Ulrike Lischetti, Denis Jabaudon, and Randall J Platt. Transcriptional linkage analysis with in vivo aav-perturb-seq. *Nature*, 2023. doi: 10.1038/s41586-023-06570-y.

Daniel Schraivogel, Andreas R Gschwind, Jennifer H Milbank, Daniel R Leonce, Petra Jakob, Lukas Mathur, Jan O Korbel, Christoph A Merten, Lars Velten, and Lars M Steinmetz. Targeted perturb-seq enables genome-scale genetic screens in single cells. *Nature methods*, 2020. doi: 10.1038/s41592-020-0837-5.

Stuart Shanks, Jonathan Embley-Riches, Jianheng Liu, Andromachi Maria Delfaki, Carlo Ciliberto, and Dimitrios Kanoulas. Dreamernav: learning-based autonomous navigation in dynamic indoor environments using world models. *Frontiers in robotics and AI*, 2025. doi: 10.3389/frobt.2025.1655171.

Artur Szałata, Karin Hrovatin, Sören Becker, Alejandro Tejada-Lapuerta, Haotian Cui, Bo Wang, and Fabian J Theis. Transformers in single-cell omics: a review and new perspectives. *Nature methods*, 2024. doi: 10.1038/s41592-024-02353-z.

Sahil Verma, Prabhat Kumar, and Jyoti Prakash Singh. Mlnas: Meta-learning based neural architecture search for automated generation of deep neural networks for plant disease detection tasks. *Network (Bristol, England)*, 2025. doi: 10.1080/0954898X.2024.2374852.

Xiaokang Wang, Navneet Rai, Beatriz Merchel Piovesan Pereira, Ameen Eetemadi, and Ilias Tagkopoulos. Accelerated knowledge discovery from omics data by optimal experimental design. *Nature communications*, 2020. doi: 10.1038/s41467-020-18785-y.

Laurence O W Wilson, Daniel Reti, Aidan R O'Brien, Robert A Dunne, and Denis C Bauer. High activity target-site identification using phenotypic independent crispr-cas9 core functionality. *The CRISPR journal*, 2018. doi: 10.1089/crispr.2017.0021.

Jia Wu, Zixuan Xu, and SiYao Qiao. Fast heterogeneous graph neural network generation via meta contrastive learning. *Neural networks : the official journal of the International Neural Network Society*, 2025. doi: 10.1016/j.neunet.2025.107727.

Jiqing Wu and Viktor H Koelzer. Towards generative digital twins in biomedical research. *Computational and structural biotechnology journal*, 2024. doi: 10.1016/j.csbj.2024.09.030.

Yongqing Zhang, Maocheng Wang, Zixuan Wang, Yuhang Liu, Shuwen Xiong, and Quan Zou. Metasem: Gene regulatory network inference from single-cell rna data by meta-learning. *International journal of molecular sciences*, 2023. doi: 10.3390/ijms24032595.

Xinhe Zheng, Boli Wu, Yuejia Liu, Sean K Simmons, Kwanho Kim, Grace S Clarke, Abdullah Ashiq, Joshua Park, Jiwen Li, Zhilin Wang, Liqi Tong, Qizhao Wang, Keerthi T Rajamani, Rodrigo Muñoz-Castañeda, Shang Mu, Tianbo Qi, Yunxiao Zhang, Zi Chao Ngiam, Naoto Ohte, Carina Hanashima, Zhuhao Wu, Xiangmin Xu, Joshua Z Levin, and Xin Jin. Massively parallel in vivo perturb-seq reveals cell-type-specific transcriptional networks in cortical development. *Cell*, 2024. doi: 10.1016/j.cell.2024.04.050.

Julius Zimmermann, Kai Budde, Nils Arbeiter, Francia Molina, Alexander Storch, Adelinde M Uhrmacher, and Ursula van Rienen. Using a digital twin of an electrical stimulation device to monitor and control the electrical stimulation of cells. *Frontiers in bioengineering and biotechnology*, 2021. doi: 10.3389/fbioe.2021.765516.

Ángela Pérez-Benito, José Manuel García-Aznar, María José Gómez-Benito, and María Ángeles Pérez. Patient-specific prostate tumour growth simulation: a first step towards the digital twin. *Frontiers in physiology*, 2024. doi: 10.3389/fphys.2024.1421591.

A Extended ODE Formulation

The full five-gene regulatory network is described by the following system of ordinary differential equations using Hill-type kinetics. We use the notation $H^+(x, K, n) = x^n/(K^n + x^n)$ for Hill activation and $H^-(x, K, n) = K^n/(K^n + x^n)$ for Hill repression.

A.1 Gene A (Transcription Factor, Toggle Switch)

Gene A self-activates and is repressed by Gene B:

$$\frac{dx_A}{dt} = \beta_A \cdot p_A \cdot H^+(x_A, K_{AA}, n) \cdot H^-(x_B, K_{BA}, n) \cdot (1 - 0.3D) - \gamma_A \cdot x_A \tag{3}$$

where β_A is the maximal production rate (cell-type-specific), p_A is the perturbation multiplier ($p_A = 1.0$ for WT, 0.0 for KO, 0.2 for KD, 3.0 for OE), $K_{AA} = 1.0$ is the self-activation binding constant, $K_{BA} = 0.8$ is the B-repression binding constant, $n = 2$ is the Hill coefficient, $D \in [0, 1]$ is the drug dosage, and $\gamma_A = 0.1$ is the decay rate.

A.2 Gene B (Transcription Factor, Toggle Switch)

Gene B self-activates and is repressed by Gene A, forming the complementary arm of the toggle switch:

$$\frac{dx_B}{dt} = \beta_B \cdot p_B \cdot H^+(x_B, K_{BB}, n) \cdot H^-(x_A, K_{AB}, n) \cdot (1 - 0.3D) - \gamma_B \cdot x_B \tag{4}$$

where β_B is the maximal production rate (cell-type-specific), $K_{BB} = 1.0$, $K_{AB} = 0.8$, and $\gamma_B = 0.1$.

A.3 Gene R (Receptor)

Gene R is activated by extracellular ligand L:

$$\frac{dx_R}{dt} = 0.8 \cdot p_R \cdot H^+(L, K_L, n) \cdot (1 - 0.3D) - \gamma_R \cdot x_R \tag{5}$$

where K_L is the ligand binding constant (cell-type-specific; see Table 2), and $\gamma_R = 0.15$.

A.4 Gene C (Downstream Transcription Factor)

Gene C is activated by the receptor R:

$$\frac{dx_C}{dt} = 0.6 \cdot p_C \cdot H^+(x_R, K_{RC}, n) \cdot (1 - 0.3D) - \gamma_C \cdot x_C \tag{6}$$

where $K_{RC} = 1.0$ and $\gamma_C = 0.12$.

A.5 Gene P (Phenotype)

Gene P integrates multiple regulatory inputs—activated by A and C, repressed by B:

$$\frac{dx_P}{dt} = 0.4 \cdot p_P \cdot \big[H^+(x_A, K_{AP}, n) + H^+(x_C, K_{CP}, n) - H^-(x_B, K_{BP}, n)\big] \cdot (1 - 0.3D) - \gamma_P \cdot x_P \tag{7}$$

where $K_{AP} = K_{CP} = 0.6$, $K_{BP} = 0.8$, and $\gamma_P = 0.08$.

Note on Gene P formulation. The subtraction of $H^-(x_B, K_{BP}, n)$ is a non-standard formulation: since H^- decreases as B increases, subtracting it means P production *increases* with B at the single-equation level. However, the net biological behavior is qualitatively correct because A and B are coupled via the toggle switch: high B drives low A (via mutual repression), and the resulting decrease in $H^+(x_A, K_{AP}, n)$ dominates the P dynamics, yielding low P when B is dominant. All experimental results were generated with this formulation.

A.6 Noise Model

For each simulated experiment, the ODE system is integrated to steady state ($t = 50$ time units, 1000 integration steps) using LSODA Greber & Fussenegger (2007). The steady-state expression is computed as the mean over the last 200 time steps to ensure convergence. Single-cell observations are generated by adding i.i.d. Gaussian noise:

$$x_i^{(k)} = x_i^* + \epsilon_i^{(k)}, \quad \epsilon_i^{(k)} \sim \mathcal{N}(0, \sigma^2), \quad \sigma = 0.05 \tag{8}$$

with values clipped to a minimum of 0.01 to ensure non-negativity.

A.7 Complete Parameter Table

Table 2: Complete ODE parameter table for each cell type. All cell types share the same Hill coefficient ($n = 2$), binding constants, and decay rates except where noted.

Parameter	Type 1	Type 2	Type 3 (OOD)	Description
β_A	1.5	1.0	1.0	Gene A maximal production rate
β_B	1.0	1.5	1.0	Gene B maximal production rate
K_L	0.5	0.5	0.3	Ligand binding constant (R)
K_{AA}, K_{BB}	1.0	1.0	1.0	Self-activation binding constant
K_{BA}, K_{AB}	0.8	0.8	0.8	Cross-repression binding constant
K_{RC}	1.0	1.0	1.0	Receptor→C binding constant
K_{AP}, K_{CP}	0.6	0.6	0.6	Activation binding (phenotype P)
K_{BP}	0.8	0.8	0.8	Repression binding (B→P)
n	2.0	2.0	2.0	Hill coefficient (all interactions)
γ_A, γ_B	0.10	0.10	0.10	TF decay rate
γ_R	0.15	0.15	0.15	Receptor decay rate
γ_C	0.12	0.12	0.12	Downstream TF decay rate
γ_P	0.08	0.08	0.08	Phenotype decay rate
σ_{noise}	0.05	0.05	0.05	Observation noise std

The key difference between cell types is the balance of the toggle switch: Type 1 ($\beta_A > \beta_B$) favors the A-dominant steady state, Type 2 ($\beta_B > \beta_A$) favors the B-dominant state, and Type 3 ($\beta_A = \beta_B$) has balanced toggle dynamics but altered ligand sensitivity ($K_L = 0.3$ vs. 0.5), creating a controlled distribution shift for OOD evaluation.

A.8 Perturbation Encoding

The perturbation space is encoded as follows. For each gene $i \in \{A, B, R, C, P\}$, the perturbation state is mapped to a continuous multiplier:

Table 3: Perturbation state encoding.

State	Multiplier p_i	Biological interpretation
Wild-type (WT)	1.0	No intervention
Knockout (KO)	0.0	Complete gene deletion (CRISPR KO)
Knockdown (KD)	0.2	80% reduction (CRISPRi or siRNA)
Overexpression (OE)	3.0	3-fold increase (CRISPRa or transgene)

Each experiment is restricted to at most two genes perturbed from WT to keep the combinatorial space manageable. With 5 genes, 4 perturbation states, and the 2-gene constraint, the effective perturbation catalog contains $\binom{5}{0} + \binom{5}{1} \times 3 + \binom{5}{2} \times 3^2 = 1 + 15 + 90 = 106$ distinct genetic perturbation combinations per cell type. Combined with continuous ligand and drug dosage, the full perturbation space is 106-discrete × $[0, 1]^2$-continuous.

B Algorithm Pseudocode

Planner subroutine. The planner selects experiments using the following acquisition procedure:

```
Require: World model $M_\theta$, dataset $\mathcal{D}$, cell type $c$, candidate set size $B = 200$, exploration $\beta = 1.0$
Sample $B$ candidate experiments $\{x_1, \ldots, x_B\}$ uniformly at random
for each candidate $x_b$ do
    Run $K = 10$ forward passes with MC Dropout enabled
    $\mu_P(x_b) \leftarrow$ mean of phenotype gene P predictions
    $\sigma_P(x_b) \leftarrow$ std of phenotype gene P predictions
    $a(x_b) \leftarrow \mu_P(x_b) + \beta \cdot \sigma_P(x_b)$                    // UCB score
end for
return $x^* = \arg\max_b a(x_b)$
```

C Implementation Details

C.1 World Model Architecture

The world model is a standard multilayer perceptron (MLP) implemented in PyTorch. The default architecture has:

- Input dimension: 10 (3 one-hot cell type + 5 perturbation codes + 1 ligand + 1 drug)
- Hidden layers: 1 layer with 64 units (default)
- Activation: ReLU
- Dropout: 0.1 (applied during both training and inference for MC Dropout)
- Output dimension: 5 (predicted mean expression for genes A, B, R, C, P)
- Total parameters: $10 \times 64 + 64 + 64 \times 5 + 5 = 704 + 325 = 1{,}029$

C.2 Architecture Registry

The candidate architecture registry contains configurations varying hidden layer sizes and depths:

C.3 Training Protocol

- **Optimizer**: Adam with learning rate 0.001

Algorithm 1 Co-Evolution Loop

```
Require: Simulator $\mathcal{S}$, registry $\mathcal{R} = \{a_1, \dots, a_R\}$, phases $T$, experiments per phase $N$, burn-in $T_0$,
    capacity factor $K$, acceptance margin $\epsilon$
Initialize world model $M_\theta$ with architecture $a_0 = [64]$
Initialize planner $\pi$ with exploration parameter $\beta$
Initialize dataset $\mathcal{D} \leftarrow \emptyset$
for phase $t = 1, \dots, T$ do
    // Inner loop: Experiment–Observe–Update
    for experiment $j = 1, \dots, N$ do
        Sample cell type $c \sim \text{Uniform}(\{1, 2\})$
        Select perturbation $x_j \leftarrow \pi(\mathcal{D}, M_\theta, c)$                    // UCB acquisition
        Observe outcome $Y_j \leftarrow \mathcal{S}(c, x_j) + \epsilon$                        // Noisy simulator
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(c, x_j, Y_j)\}$
    end for
    Retrain $M_\theta$ on $\mathcal{D}$ with Adam (lr=0.001, early stopping)                 // Weight update

    // Outer loop: Architecture self-modification (after burn-in)
    if $t > T_0$ then
        Split $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ (80%) and $\mathcal{D}_{\text{val}}$ (20%)
        $\ell_{\text{current}} \leftarrow \text{MSE}(M_\theta, \mathcal{D}_{\text{val}})$
        $\ell_{\text{best}} \leftarrow \ell_{\text{current}}$, $a_{\text{best}} \leftarrow a_{\text{current}}$
        for each candidate architecture $a_k \in \mathcal{R}$ do
            if $|\theta(a_k)| \le K \cdot |\mathcal{D}_{\text{train}}|$ then
                // Capacity gate passed
                Train $M'$ with architecture $a_k$ on $\mathcal{D}_{\text{train}}$ for up to 100 epochs (early stopping)
                $\ell_k \leftarrow \text{MSE}(M', \mathcal{D}_{\text{val}})$
                if $\ell_k < \ell_{\text{best}}$ then
                    $\ell_{\text{best}} \leftarrow \ell_k$, $a_{\text{best}} \leftarrow a_k$
                end if
            end if
        end for
        if $(\ell_{\text{current}} - \ell_{\text{best}})/\ell_{\text{current}} > \epsilon$ then
            // Validation gate passed — accept architecture change
            $a_{\text{current}} \leftarrow a_{\text{best}}$
            Retrain $M_\theta$ with $a_{\text{best}}$ on full $\mathcal{D}$
        end if
    end if
end for
return $M_\theta, \mathcal{D}$
```

Table 4: Architecture registry used for self-modification evaluation.

Architecture	Hidden Dims	Parameters
Small-shallow	[32]	517
Default	[64]	1,029
Medium-shallow	[128]	2,053
Large-shallow	[256]	4,101
Small-deep	[32, 32]	1,573
Default-deep	[64, 64]	5,189
Medium-deep	[128, 128]	18,565
Large-deep	[256, 256]	69,893

- **Loss function**: Mean squared error (MSE) across all 5 genes
- **Epochs**: Up to 100 with early stopping (patience = 10 epochs)
- **Batch size**: Full batch (dataset fits in memory)
- **Validation split**: 80/20 train/validation for outer loop evaluation

- **MC Dropout inference**: 10 forward passes per uncertainty estimate

C.4 Experiment Configuration

- **Phases**: $T = 10$
- **Experiments per phase**: $N = 50$ (default), ablated over $\{20, 50, 100\}$
- **Total experimental budget**: $T \times N = 500$ experiments (default)
- **Burn-in**: Outer loop evaluation begins after phase 3 ($T_0 = 3$)
- **Capacity factor**: $K = 0.1$ (ablated over $\{0.05, 0.1, 0.2\}$)
- **Acceptance margin**: $\epsilon = 0.02$ (ablated over $\{0.01, 0.02, 0.05\}$)
- **Exploration parameter**: $\beta = 1.0$
- **Random seeds**: 5 seeds per configuration ($\{42, 43, 44, 45, 46\}$)
- **Candidate batch size**: $B = 200$ random candidates per planning step

C.5 Computational Requirements

All experiments were run on a single machine with an Apple M-series processor. The complete experiment suite (baseline runs, ungated variant, and ablation studies—65 total runs) completed in approximately 21 minutes. Individual runs average 15–20 seconds each, dominated by ODE integration time. No GPU was required; all PyTorch operations run on CPU given the small model sizes.

C.6 Software Dependencies

- Python 3.11, PyTorch 2.x, NumPy, SciPy (for ODE integration), Matplotlib, Seaborn
- ODE integration: `scipy.integrate.odeint` (LSODA method)
- Random seed management: `np.random.seed()` and `torch.manual_seed()`

D Full Experimental Results

D.1 Per-Seed Results: Baseline Methods

Table 5 reports individual seed results for all four baseline methods, providing full transparency into the variance across random initializations.

D.2 Per-Phase Learning Dynamics

Table 6 reports the ID MSE at each phase for all methods (averaged over 5 seeds), providing detailed learning dynamics beyond the summary in Table 1.

D.3 Per-Phase OOD MSE Dynamics

D.4 Cumulative Phenotype Reward

Table 8 shows cumulative phenotype reward at each phase (averaged over 5 seeds), demonstrating the accelerating advantage of BO-guided methods.

The retrained methods achieve approximately 3.8× higher cumulative reward than static methods by the final phase, demonstrating the compounding benefit of coupling model improvement with informed experiment selection.

D.5 Ungated Co-Evolution Results

Table 9 compares gated and ungated co-evolution variants per seed.

Table 5: Per-seed final MSE results for baseline methods after 500 experiments.

Method	Seed	ID MSE	OOD MSE
Random-Static	42	59.66	23.47
Random-Static	43	64.44	21.41
Random-Static	44	87.03	27.66
Random-Static	45	48.69	29.59
Random-Static	46	85.91	10.30
Mean ± std		69.15 ± 15.04	22.49 ± 6.75
BO-Static	42	152.32	16.77
BO-Static	43	101.06	23.14
BO-Static	44	62.15	19.90
BO-Static	45	108.13	26.93
BO-Static	46	58.95	19.71
Mean ± std		96.52 ± 34.24	21.29 ± 3.47
BO-Retrain	42	1.85	70.14
BO-Retrain	43	5.35	88.16
BO-Retrain	44	1.83	92.64
BO-Retrain	45	4.85	78.17
BO-Retrain	46	2.14	91.13
Mean ± std		3.21 ± 1.56	84.05 ± 8.59
BO-CoEvolve	42	2.28	72.82
BO-CoEvolve	43	5.41	87.54
BO-CoEvolve	44	5.08	95.42
BO-CoEvolve	45	3.91	79.22
BO-CoEvolve	46	2.73	91.80
Mean ± std		3.88 ± 1.24	85.36 ± 8.27

Table 6: Mean ID MSE per phase across methods (averaged over 5 seeds).

Method	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Rand-Static	68.0	75.3	79.0	64.8	70.1	62.9	61.1	70.6	73.0	69.1
BO-Static	87.2	93.8	91.6	92.2	92.1	89.0	91.6	96.4	91.8	96.5
BO-Retrain	41.6	81.8	52.2	32.6	18.1	10.0	5.6	4.7	3.0	3.2
BO-CoEvolve	41.7	82.9	51.5	30.9	14.3	10.8	7.0	5.1	2.9	3.9

On this small testbed, the ungated variant achieves slightly lower mean ID MSE (2.63 vs. 3.88), as the gating conservatively prevents architecture changes that might have been slightly beneficial. The ungated variant also shows lower ID variance (std 0.73 vs. 1.24), though it exhibits higher OOD variance (std 10.10 vs. 8.27), consistent with less stable generalization under unrestricted modifications. The gated mechanism's value would be more pronounced with larger candidate registries or in longer runs where overfitting risks increase.

E Ablation Studies (Extended)

E.1 Experiments Per Phase

With 20 experiments per phase (200 total), the model achieves competitive ID MSE but with higher variance due to limited data per phase. Increasing to 100 per phase (1000 total) provides the best ID performance (2.35 ± 0.94) with reduced variance. OOD MSE is lowest for the 20-experiment setting because fewer in-distribution observations limit overfitting.

Table 7: Mean OOD MSE per phase across methods (averaged over 5 seeds).

Method	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Rand-Static	25.7	35.5	28.5	32.8	30.4	26.3	29.9	30.5	30.7	22.5
BO-Static	25.8	26.3	30.1	33.9	21.9	30.0	26.9	23.7	26.9	21.3
BO-Retrain	29.4	41.4	40.8	41.6	44.8	52.6	61.6	70.9	75.7	84.1
BO-CoEvolve	29.4	41.4	40.5	41.5	44.7	52.3	61.4	70.9	76.2	85.4

Table 8: Mean cumulative phenotype reward per phase across methods (averaged over 5 seeds).

Method	P2	P4	P6	P8	P10
Random-Static	270	552	826	1110	1370
BO-Static	376	749	1129	1513	1889
BO-Retrain	549	1574	2687	3946	5251
BO-CoEvolve	549	1562	2696	3950	5235

E.2 Registry Size

Increasing the registry from 6 to 8 candidates provides marginal improvement (3.50 vs. 3.88 ID MSE), suggesting that the search space is well-covered by the smaller registry.

E.3 Capacity Factor K

The results are identical across K values because all settings produce a capacity threshold below the smallest candidate architecture (517 parameters). Even the most permissive setting ($K = 0.2$, allowing $0.2 \times 400 = 80$ parameters) rejects all candidates. The capacity gate is thus universally binding on this testbed, and all variation in self-modification behavior is determined by this constraint alone.

E.4 Acceptance Margin ϵ

Similar to the capacity factor, the acceptance margin has no effect because no architecture change is accepted in any configuration—the initial 64-hidden-unit architecture is already sufficient for this testbed. The ablation confirms that the gating mechanism behaves correctly: it does not force unnecessary changes when the current architecture is adequate.

E.5 Analysis of Ablation Robustness

The ablation studies reveal that the co-evolution framework is highly robust to hyperparameter choices on this testbed, primarily because the initial architecture is well-suited to the problem's complexity. This is an expected and desirable property: the gating mechanism should not force changes when the current configuration is adequate. In more complex settings with higher-dimensional input spaces, larger gene networks, or multi-omics data, we expect the capacity factor and acceptance margin to play more significant roles in controlling self-modification behavior.

F Additional Figures

F.1 Interpretation of Predictive Performance

The scatter plot (Figure 7) reveals several patterns. First, predictions for genes with higher expression levels (typically A or B in their dominant cell-type state) are more accurate, likely because these data points provide stronger gradient signals during training. Second, the spread increases at intermediate expression levels, where the nonlinear dynamics of the toggle switch create higher prediction uncertainty. Third, the $r = 0.694$ correlation, while moderate, represents substantial improvement over untrained random predictions and demonstrates that the world model captures meaningful biological signal.

Table 9: Per-seed comparison of gated vs. ungated co-evolution after 500 experiments.

	Gated (BO-CoEvolve)		Ungated	
Seed	ID MSE	OOD MSE	ID MSE	OOD MSE
42	2.28	72.82	3.00	70.84
43	5.41	87.54	3.02	84.62
44	5.08	95.42	2.50	97.38
45	3.91	79.22	3.34	75.73
46	2.73	91.80	1.30	92.33
Mean	3.88	85.36	2.63	84.18
Std	1.24	8.27	0.73	10.10

Table 10: Effect of experiments per phase on final performance. Total budget scales with N: 200, 500, or 1000 experiments.

N (exp/phase)	Total Budget	ID MSE	OOD MSE
20	200	3.15 ± 2.12	72.11 ± 7.46
50 (default)	500	3.88 ± 1.24	85.36 ± 8.27
100	1000	2.35 ± 0.94	77.85 ± 2.50

The bar chart comparison (Figure 8) makes the 95%+ reduction in ID MSE immediately visually apparent. The small variance in retrained methods (error bars barely visible on the same scale) demonstrates that the improvement is consistent across random seeds.

G Extended Discussion

G.1 Bias-Variance Tradeoff in Co-Evolution

The most striking finding in our experiments is the inverse relationship between ID and OOD performance: methods that achieve the best ID MSE also show the worst OOD MSE. This is not a failure of co-evolution but rather a fundamental consequence of the bias-variance tradeoff in adaptive systems.

Static methods (Random-Static and BO-Static) never update their world model, so their predictions reflect the initial model's inductive biases rather than learned patterns specific to in-distribution data. This "ignorance" paradoxically helps with OOD generalization: the model's predictions are equally uninformed about all cell types, resulting in moderate but consistent error across distributions.

Retrained methods progressively specialize their representations to in-distribution data. After 500 experiments in cell types 1 and 2, the model learns detailed features of the A-biased and B-biased toggle switch dynamics but has never observed the balanced dynamics and altered ligand kinetics of cell type 3. The resulting distributional gap grows with each retraining cycle.

This suggests several directions for future work: (1) explicit OOD regularization during retraining (e.g., domain randomization); (2) periodic evaluation on held-out cell types during co-evolution; (3) architecture features that promote generalization (e.g., shared representations across cell types); and (4) data augmentation strategies that simulate plausible OOD scenarios.

G.2 Scaling Considerations

Our five-gene testbed was designed to be computationally tractable while exhibiting meaningful dynamics. Scaling to larger networks introduces several challenges:

- **Perturbation space**: With G genes and 4 perturbation states (restricting to ≤ 2 perturbed genes), the space scales as $O(G^2)$. For $G = 20,000$ (genome-scale), this becomes intractable without learned acquisition functions.

Table 11: Effect of architecture registry size on final performance.

Registry Size	ID MSE	OOD MSE
6 (default)	3.88 ± 1.24	85.36 ± 8.27
8	3.50 ± 1.21	83.56 ± 8.13

Table 12: Effect of capacity factor K on final performance.

K	ID MSE	OOD MSE
0.05	3.88 ± 1.24	85.36 ± 8.27
0.10 (default)	3.88 ± 1.24	85.36 ± 8.27
0.20	3.88 ± 1.24	85.36 ± 8.27

- **ODE complexity**: Real regulatory networks have thousands of interactions with unknown parameters. Hybrid approaches combining mechanistic and data-driven models would be necessary.
- **Model capacity**: Larger gene networks require more expressive world models, making the capacity-data tradeoff more critical and the outer loop's architecture search more impactful.
- **Noise model**: Real single-cell data exhibits overdispersion, zero-inflation, and technical artifacts that our Gaussian noise model does not capture.

G.3 Connection to Structured Model Adaptation

Our co-evolving agent implements a constrained form of model adaptation where the agent can evaluate modifications to its own predictive model based on accumulated evidence. The key feature is the validation-and-capacity gating mechanism, which provides empirical safeguards against harmful modifications.

In the taxonomy of adaptive systems: our *inner loop* (weight updates via gradient descent) represents the most common and well-understood form of model adaptation; our *outer loop* (architecture modification via registry search) represents a more structured form that is analogous to neural architecture search but constrained to a small candidate set; and the *planner adaptation* (implicit, through updated model predictions) represents an emergent form where behavioral changes arise from improved world models rather than explicit policy modification.

The gating mechanism addresses a key concern in model adaptation: the stability-plasticity dilemma. Too much plasticity (ungated modifications) risks catastrophic regressions; too much stability (never modifying) prevents beneficial adaptations. Our empirical margin ϵ provides a simple but effective balance, and our ablation studies show that performance is robust to the specific margin value chosen.

G.4 Implications for Experimental Biology

If extended to real biological systems, the co-evolving agent paradigm could transform how perturbation screens are conducted:

1. **Adaptive experimental design**: Instead of fixed screening libraries, experiments would be selected dynamically based on accumulated knowledge, focusing resources on informative perturbation combinations.
2. **Continuous model improvement**: Virtual cell models would improve throughout the experimental campaign, providing increasingly accurate predictions that guide subsequent experiments.
3. **Automated hypothesis generation**: The planner's acquisition function implicitly generates hypotheses about which perturbations will produce desired outcomes, connecting experiment selection to scientific discovery.

Table 13: Effect of acceptance margin ϵ on final performance.

ϵ	ID MSE	OOD MSE
0.01	3.88 ± 1.24	85.36 ± 8.27
0.02 (default)	3.88 ± 1.24	85.36 ± 8.27
0.05	3.88 ± 1.24	85.36 ± 8.27

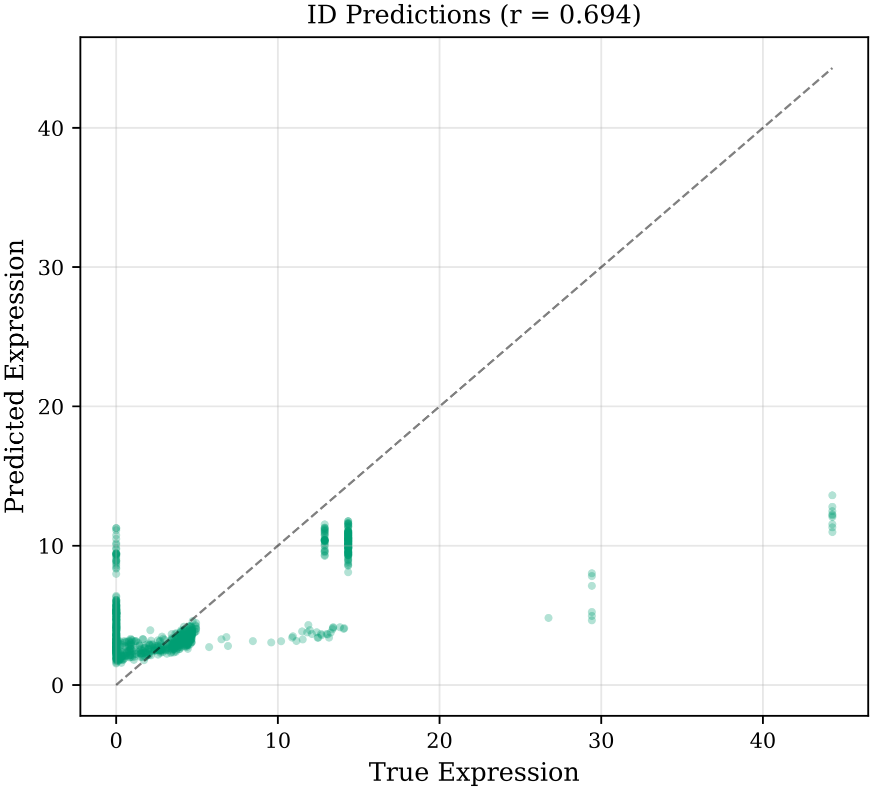

Figure 7: Predicted vs. true gene expression for BO-CoEvolve after 500 experiments. Each point represents a single gene prediction on the test set. The Pearson correlation $r = 0.694$ indicates moderate predictive accuracy across the expression range, with tighter clustering near the diagonal for higher-expression genes.

4. **Sample efficiency**: By leveraging model predictions to prioritize experiments, the co-evolving approach could substantially reduce the number of experiments needed to characterize perturbation responses, addressing a key bottleneck in high-throughput biology.

H Reproducibility Statement

All experiments are fully reproducible. The complete codebase, including the ODE simulator, world model, planner, co-evolution loop, experiment runner, and figure generation scripts, is contained in a single self-contained Python file (`run_all.py`). Random seeds are fixed for all runs (42–46). The total computation time is approximately 21 minutes on a single CPU. All result JSON files and generated figure PDFs are included in the submission package. The experiment code depends only on standard packages (PyTorch, NumPy, SciPy, Matplotlib) and requires no GPU resources.

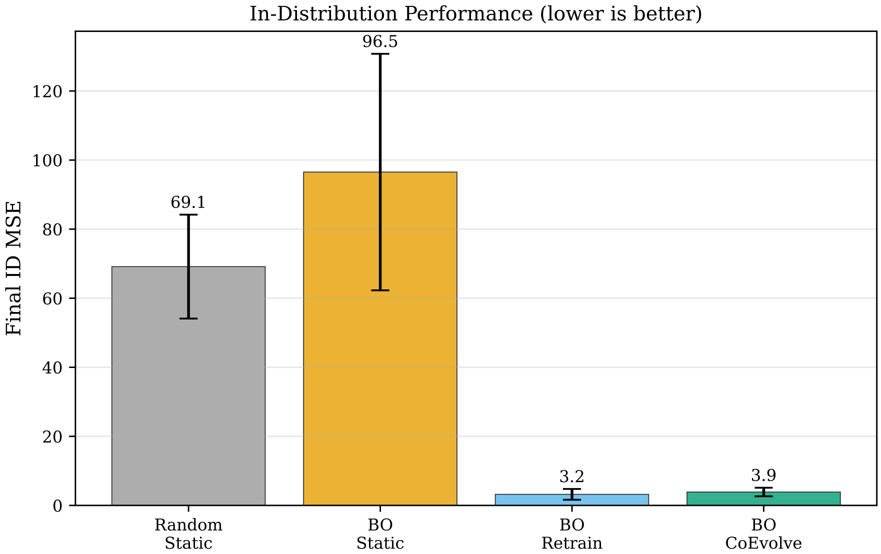

Figure 8: Final in-distribution MSE comparison across all four methods. Bar heights show mean over 5 seeds; error bars show ±1 standard deviation. The dramatic difference between static (MSE $\sim$70–97) and retrained methods (MSE $\sim$3–4) highlights the critical importance of continual learning.