PEAC: JOINT GENERATION OF PROMOTER-ENHANCER PAIRS WITH ACTIVITY CONDITIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing regulatory DNA with predictable gene-expression activity remains challenging since promoters and enhancers usually interact as pairs, while current approaches always focus on the design of individual regulatory elements. Ignoring promoter–enhancer interactions results in regulatory elements that perform well individually but fail to achieve the desired activity when combined. To address this problem, we propose PEAC, a **P**romoter-**E**nhancer **A**ctivity-**C**onditioned co-design framework that achieves joint modeling of promoter-enhancer pairs. Specifically, we first train a student model that is distilled from a promoter teacher and an enhancer teacher simultaneously, and then fine-tune the student model with the activity constraint to achieve the conditional generation of the target activity. Furthermore, we also incorporate pair-level fusion and latent distribution alignment to make the generated promoter-enhancer pairs better matched. Extensive experiments show that PEAC achieves excellent performance in generating activity conditioned promoter-enhancer pairs with sequence realism. In oracle-based evaluation, PEAC demonstrated the ability to generate pairs across low-, medium-, and high-activity regimes, with target regime fractions of 79.1%, 91.7% and 67.8%, compared with 31.8%, 68.6%, and 11.6% for baseline. This improvement is accompanied by realistic GC content, high motif enrichment, and low biological distance compared to reference. Together, these results establish PEAC as a native promoter-enhancer generator that makes pair-level activity the explicit design target and provides an oracle-evaluated framework for activity-conditioned regulatory sequence co-design.

1 INTRODUCTION

Cis-regulatory sequences (CREs) are segments of non-coding DNA that regulate the transcription of nearby genes (Wittkopp & Kalay, 2011). The activity of these sequences is determined by the presence and arrangement of specific DNA motifs, which serve as binding sites for transcription factors (TFBS). These interactions ultimately influence the recruitment of the transcriptional machinery and the initiation of gene expression. Engineering these sequences offers the potential to control gene expression, which is essential for applications in synthetic biology and gene therapy. However, designing effective regulatory sequences remains a significant challenge due to the complex interactions between DNA motifs, transcription factors, and chromatin structure (Kim & Wysocka, 2023).

Current approaches have primarily worked on generating single regulatory elements. Large language models have been applied to DNA sequence design and have shown promise in generating functional regulatory sequences. HyenaDNA, Evo and DNABERT already demonstrate the ability of understanding the DNA sequence language and potentially generating functional sequences (Nguyen et al., 2023; Brixi et al., 2026; Ji et al., 2021). Regulatory language models have been further trained to generate either promoters or enhancers in tools such as regLM, TACO, Ctrl-DNA, and DeepSEED (Lal et al., 2024; Yang et al., 2025; Chen et al., 2025; Zhang et al., 2023). Other methods like DFM and D3 tried to approach the problem with diffusion models (Stark et al., 2024; Sarkar et al., 2024). All the above methods have shown that machine learning models can help design single regulatory elements in various species and cell types. However, pair-level activity control benefits from modeling promoter and enhancer elements together, because independently optimized elements can yield mismatched activity regimes given the complex interactions between

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

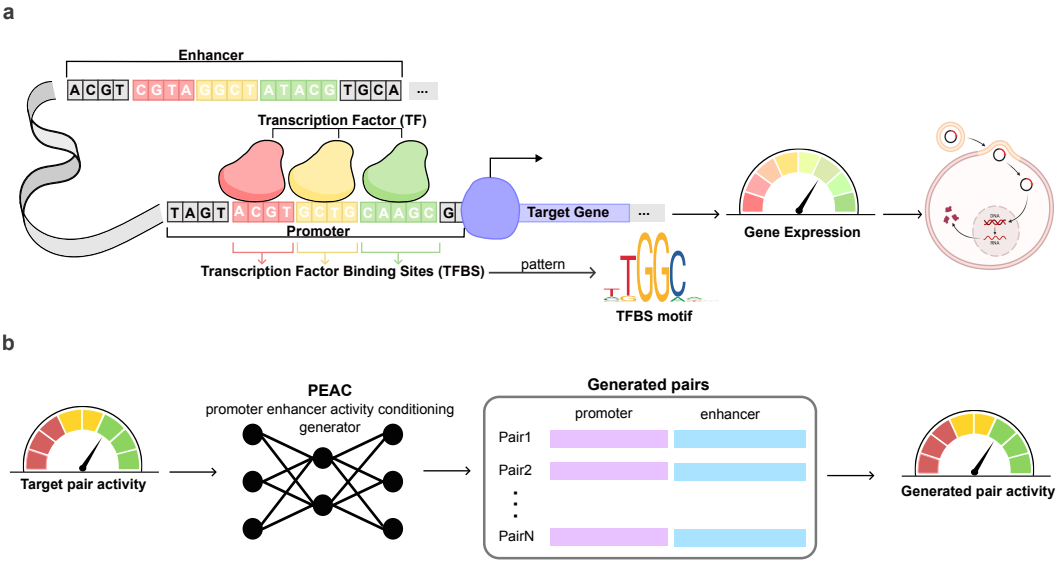

Figure 1: Conceptual overview of activity-conditioned promoter-enhancer co-design. (a) Regulatory sequence activity is shaped by promoter and enhancer elements, transcription factor binding sites, TFBS motifs, and their downstream effect on target gene expression. Designing these sequences has potential applications in cell engineering. (b) PEAC consists of a target pair-activity condition, a joint generator for regulatory sequences, a generated-pair module that outputs matched promoter and enhancer sequences, and a pair-activity evaluator that scores the generated promoter-enhancer pairs.

them (Bergman et al., 2022; Lin et al., 2026). Joint design may therefore provide additional value for controlling the activity distribution of paired regulatory sequences. Existing models could be extended to generate pairs, the technical challenge is learning the interactions between the two elements and demonstrating benefits beyond independently conditioned generation given current sequencing technologies.

Figure 1 summarizes the motivation for moving from single regulatory element design to paired regulatory sequence generation. Promoters are cis-regulatory DNA regions surrounding transcription start sites that recruit and organize the transcription machinery to initiate transcription. Enhancers are the regions that bind transcription factors and increase transcription from target promoters, acting over short or long genomic distances. Promoters and enhancers contain motif-level information, but their functional effect is ultimately expressed through pair-level regulatory activity.

Here, we propose PEAC, a Promoter–Enhancer Activity-Conditioned co-design framework for joint modeling of promoter–enhancer pairs. Instead of generating the two regulatory elements independently, PEAC first distills a shared student from a promoter teacher and an enhancer teacher in parallel, so that element-specific sequence statistics are transferred into a common generator. The student is then fine-tuned with an activity constraint, enabling conditional generation at a target activity level. To further model pairwise interactions between the two partners, we incorporate pair-level fusion and latent distribution alignment, which couple the two sequences both in feature space and in the joint latent geometry. As a result, PEAC produces promoter–enhancer pairs that are jointly coherent and activity-controllable.

This paper mainly makes three contributions:

A comprehensive benchmark of regulatory sequence design. We evaluate existing models across function, controllability, realism, and novelty, revealing trade-offs that single-metric evaluations overlook and providing a foundation for assessing both individual-element and paired design.

A novel framework for activity-conditioned promoter–enhancer co-design. We introduce PEAC, which extends controllable generation from individual regulatory elements to promoter–enhancer pairs.

Empirical evidence for controllable paired generation. Exhaustive experiments show that PEAC steers generated pairs into the requested activity regimes more accurately than a concatenation baseline, while preserving reference-like GC content and motif enrichment. This indicates that activity control and sequence fidelity can be achieved together in paired regulatory design.

2 Related Work

Existing regulatory DNA generation methods can be broadly divided into regulatory language models, diffusion-based generators, and general-purpose DNA language models. Among regulatory language models, regLM fine-tunes HyenaDNA-based autoregressive models on cis-regulatory elements and combines them with sequence-to-function predictors (Lal et al., 2024; Nguyen et al., 2023). Working on the same dataset, TACO further applies reinforcement learning to the pretrained autoregressive models, incorporating activity rewards and TFBS (Yang et al., 2025). Ctrl-DNA formulates regulatory sequence generation as a constrained reinforcement-learning problem, maximizing activity in a target cell type while explicitly restricting off-target activity in other cell types (Chen et al., 2025). Leonine formulates enhancer design as conditional generation given promoter sequence, cell type, and gene-expression context(Liu et al., 2025).

Diffusion and flow-based approaches provide an alternative to left-to-right autoregressive generation. DDSM models DNA within a Dirichlet probability simplex and was among the first diffusion frameworks applied to human promoter generation (Avdeyev et al., 2023). DFM uses Dirichlet flow matching to transport samples from a simple distribution to discrete promoter or enhancer sequences and supports activity-guided generation with efficient sampling (Stark et al., 2024). D3 introduces score-entropy discrete diffusion for conditional regulatory DNA generation, improving functional control and regulatory realism over earlier diffusion and flow-matching approaches (Sarkar et al., 2024).

In parallel, general-purpose DNA language models provide reusable genomic representations. Models like HyenaDNA and Evo2 have the ability to understand and generate long contexts (Nguyen et al., 2023; Brixi et al., 2026). NTv3 provides a strong long-context genomic foundation model with generative extension for controllable enhancer design (Boshar et al., 2025).

Although these models contribute to regulatory sequence generation in different ways, they are not designed to learn pair-level activity of promoter and enhancer sequences. Predictive models such as EPInformer have started to model promoter-enhancer information for gene-expression prediction (Lin et al., 2026), but generative models for promoter-enhancer pairs remain underexplored. Our work addresses this gap by introducing a generative model that can co-design promoter-enhancer pairs with desired activity levels, using STARR-seq-derived initialization together with CAGE-derived pair-level activity supervision (Bergman et al., 2022).

3 Method

3.1 Problem formulation

We formulate the regulatory sequence design as conditional promoter–enhancer co-design: given a desired pair-level activity condition c, the model generates a promoter sequence p and an enhancer sequence e as one functional pair. Each training example is a triple (p, e, c), where $p = (p_1, \ldots, p_{L_p})$ and $e = (e_1, \ldots, e_{L_e})$, and c specifies the requested regulatory regime. For pair-level fine-tuning, the continuous CAGE-derived pair activity is discretized into ordered bins (low, medium, high), and c indexes the target bin.

To enable joint autoregressive generation, a pair is serialized as a single DNA token sequence

$$x = \left[\texttt{<bos>}, p_1, \ldots, p_{L_p}, \texttt{<sep>}, e_1, \ldots, e_{L_e}, \texttt{<eos>}\right]. \tag{1}$$

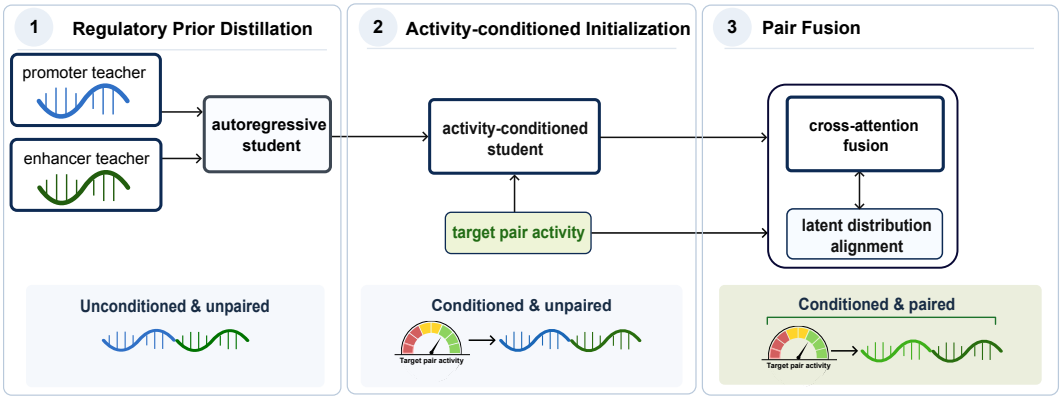

Figure 2: Overview of the three-stage PEAC framework for activity-conditioned promoter–enhancer co-design.

Promoter and enhancer tokens share the same DNA vocabulary and are distinguished by segment-type embeddings. The input embedding at each position is the sum of the DNA-token, positional, segment-type, and projected activity-condition embeddings.

We place the promoter before the enhancer so that generation follows the factorization:

$$p_\theta(p, e \mid c) = p_\theta(p \mid c)\, p_\theta(e \mid p, c). \tag{2}$$

This ordering treats the promoter as the local transcriptional context that sets the initiation site and basal regulatory state, and treats the enhancer as a distal partner generated under both c and the already sampled promoter. In a causal Transformer, every enhancer token then attends to the full promoter as left context, so the pair co-design task can be modeled during decoding without access to future tokens.

3.2 Method Overview

PEAC comprises three sequential components (Figure 2).

Regulatory prior distillation a type-conditioned causal Transformer backbone learns regulatory DNA grammar by distilling promoter and enhancer priors from selected teacher models.

Activity-conditioned initialization activity-conditioned initialization adapts the backbone with element-level activity supervision, so that sequence likelihood is tied to the requested activity condition before pair-level training.

Cross-attention pair fusion a cross-attention pair fusion module exchanges information between promoter and enhancer token representations and produces promoter, enhancer, and pair-level latent representations. Within this module, we apply latent distribution alignment and activity prediction losses to fine-tune the generator against CAGE-derived promoter-enhancer pairs, encouraging generated pairs to match the reference distribution within each activity regime.

3.3 Regulatory Prior Distillation

The first component of PEAC is a distilled regulatory sequence prior. This stage transfers general promoter and enhancer sequence grammar from upstream teacher-generated regulatory sequences into a compact autoregressive DNA student model. The teacher models are selected from a benchmark of existing regulatory sequence generation tools, so the distilled prior inherits sequence patterns from generators that performed well under the benchmark criteria. It provides a sequence model that has learned regulatory motif composition and broad promoter/enhancer structure before exposure to the final pair-level supervision.

The distilled prior is trained with standard teacher forcing on regulatory DNA sequences. For a concatenated promoter-enhancer sequence x, the language-modeling loss is

$$\mathcal{L}_{\mathrm{LM}} = -\sum_i \log p_\theta(x_i \mid x_{<i}, t_{<=i}, c), \tag{3}$$

where $t_{\leq i}$ indicates whether position i lies in the promoter or the enhancer segment. This objective yields an autoregressive backbone that generates valid regulatory DNA while keeping the two element types distinct.

3.4 Activity-conditioned initialization

The second component adapts the distilled student into an activity-conditioned initialization model. This stage establishes the first part of the conditional design pipeline: it teaches the autoregressive generator how individual regulatory sequences should respond to activity conditions before explicit promoter-enhancer interaction modeling is introduced. STARR-seq-derived data only provide gene expression results from one promoter with multiple enhancers, which lack accurate promoter-enhancer pair activity. So we use STARR-seq-derived labels and prepare activity-aware sequence representations before CAGE-derived pair-level activity learning.

The initialization objective combines sequence likelihood with activity supervision. The likelihood term preserves the regulatory DNA grammar learned during distillation, while promoter and enhancer activity losses encourage the hidden representations to encode element-level activity. We also include a condition-discrimination term that compares the likelihood of each sequence under its true activity condition with its likelihood under a mismatched condition from the same batch. This discourages the model from ignoring the condition vector and encourages activity-specific sequence likelihood.

3.5 Pair Fusion

The third component builds on this activity-conditioned initialization by introducing explicit promoter-enhancer pair fusion. The two stages are complementary: initialization provides activity-aware promoter and enhancer representations, while cross-attention fusion further models pair-level interactions between the two conditioned elements. The initialized model is therefore fine-tuned with FANTOM5 CAGE-derived expression labels assigned to promoter-enhancer candidate rows and a pair-representation module wrapped around the autoregressive backbone.

3.5.1 Cross-attention pair fusion

The pair-fusion module operates on promoter and enhancer token representations produced by the shared conditional backbone. Each fusion block first applies within-element self-attention to refine promoter and enhancer contexts separately. It then performs bidirectional cross-attention: promoter tokens and enhancer tokens query each other. This token-level exchange allows the representation of each element to depend on the sequence features of its paired partner.

Activity information is injected again during pair fusion. The target condition, activity bin, and activity quantile are projected into condition tokens, and both promoter and enhancer representations attend to these tokens. A learned pair token then attends over the fused promoter tokens, fused enhancer tokens, and condition tokens, producing a compact representation of the whole regulatory pair. After the fusion blocks, masked mean pooling gives promoter and enhancer summary vectors, while the learned pair token provides a pair-level summary. These vectors are projected into z_{prom}, z_{enh}, and z_{pair}, which are used for activity prediction, bin classification, and latent distribution alignment.

3.5.2 Latent distribution alignment

Latent distribution alignment is applied to fine-tune the model. It aligns generated and reference candidate rows in the conditioned latent space. A regression head predicts the CAGE-expression target from the fused representation z, and a classification head predicts the activity bin. During training, differentiable soft promoter and enhancer sequences are constructed from the teacher-forced generator logits using temperature-scaled Gumbel-softmax sampling. These soft sequences are encoded by

the same pair module, allowing auxiliary pair-level losses to back-propagate into the autoregressive generator.

The final objective is designed to avoid a common failure mode in regulatory sequence design: optimizing predicted activity can easily move samples away from the sequence grammar and pair structure of real regulatory elements. We therefore combine sequence likelihood, activity supervision, latent alignment, and condition consistency:

$$\mathcal{L}_{\text{final}} = \lambda_{\text{lm}}\mathcal{L}_{\text{lm}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}} + \lambda_{\text{mmd}}\mathcal{L}_{\text{mmd}} + \lambda_{\text{bin-cond}}\mathcal{L}_{\text{bin-cond}}. \tag{4}$$

Each term targets a distinct requirement of conditional pair generation.

Sequence likelihood loss. The likelihood term anchors fine-tuning to the autoregressive DNA grammar learned during distillation and initialization.

Activity supervision loss. Activity supervision gives the model a direct signal for what the requested regulatory regime means in the measured CAGE-derived expression space. We use both pointwise activity prediction and bin-level distribution matching so that the model learns more than a global high-versus-low ordering: generated pairs should also occupy the appropriate range within each target regime.

Latent MMD loss. Activity labels alone are too low dimensional to specify realistic promoter-enhancer pairs. The conditional maximum mean discrepancy (MMD) term addresses this by aligning generated and reference pair representations within each activity bin:

$$\mathcal{L}_{\text{mmd}} = \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}} \text{MMD}\left(\text{std}_b(z_{\text{gen}}), \text{std}_b(z_{\text{real}})\right). \tag{5}$$

where $\mathcal{B}$ stands for activity bins. Within each bin, both generated and reference latents are standardized using the reference-batch mean and standard deviation; these normalization statistics are detached from the computation graph. The MMD itself remains differentiable with respect to both generated and reference encoded representations during joint training. We therefore interpret this term as an auxiliary condition-aware distributional constraint, complementary to token likelihood and external sequence-realism metrics (Ouyang & Key, 2021).

Condition consistency loss. The condition-consistency term keeps the activity condition identifiable throughout generation. Without this pressure, adjacent activity bins can collapse into similar latent regions, especially because measured regulatory activity is continuous and noisy.

4 Experiments

4.1 Datasets

PEAC is trained with two public K562 resources that serve different stages of the pipeline. (Appendix A.1).

STARR-seq-derived element labels are used only for activity-conditioned initialization. We retain promoter and enhancer sequences with usable element-level activity annotations and map continuous scores to coarse regimes. These data come from reporter assays that test many enhancers against a shared promoter, so they provide element-level supervision rather than measured pair activity.

FANTOM5 CAGE-derived candidate pairs are used for pair fusion and conditional fine-tuning. Each row joins a promoter sequence, an enhancer sequence, and a processed K562 expression score assigned to that candidate pair. After filtering incomplete rows, the table contains 545,453 pairs, split into 436,363 / 54,545 / 54,545 for train / validation / test. The training target is the processed row-level score from this table, not raw CAGE tag counts and not STARR-seq $\log_2(\text{RNA}/\text{DNA})$ enrichment. Using training-split thresholds -5.3, -2.75, and -0.35, we discretize the score into low, medium, and high bins for conditioning, balanced sampling, and evaluation. The same bins define the activity condition c at generation time.

Teacher sequences used in prior distillation are generated by the selected upstream models ; preprocessing details appear in Appendix A.1.

4.2 Evaluation Metrics

We evaluate generated sequences along four axes: function, control, realism, and novelty/diversity. Function is measured with predictive activity oracles: single-element oracles for promoter and enhancer benchmarks, and a self-trained CAGE-expression oracle for promoter-enhancer designs. Control is measured by target-bin error. Sequence-quality diagnostics include GC content, motif enrichment, motif co-occurrence, motif spacing, and Fréchet Biological Distance (FBD). Novelty and diversity are measured with nearest-reference identity, generated-set diversity, and 4-mer discriminability. Discriminability is the AUROC of a generated-versus-real classifier on k-mer frequency features; values closer to 0.5 indicate that generated and reference sequences are harder to distinguish under this test. Metric definitions, oracle provenance, and baseline sampling details are reported in Appendix A.2.

For pair-level function validation, we construct a CAGE-oracle-assisted analysis and use EPInformer only as an external grouped plausibility diagnostic. CAGE-oracle was trained to predict pair activity. The held-out CAGE-expression oracle achieves test Spearman correlation of 0.868, three-regime bin accuracy of 0.773, and adjacent-bin accuracy of 0.993, so we use oracle scores for relative activity ordering and target-bin enrichment (Appendix A.3). To reduce leakage concerns, we also retrained three-bin oracles under pair-disjoint, promoter-disjoint, and enhancer-disjoint splits (Appendix Table 6, 7). Unless otherwise stated, the low, medium, and high activity regimes used in generation and evaluation are defined by training-distribution bin edges. EPInformer is applied by comparing the paired enhancer against 10 enhancers from a specified source under the same promoter, which is validated in Appendix A.4.

5 Results

In 5.1, PEAC is compared with existing regulatory sequence generators in single-element metrics. Then, we evaluate the generation of the promoter-enhancer pair conditioned on activity control in 5.2. Finally, we present ablation studies of the model components in 5.3.

5.1 Benchmark comparison

Table 1: Benchmarking of regulatory language models, diffusion models, and DNA language models. Implementation details and caveats are given in Appendix A.2. Activity-control metrics are reported only for methods that support conditional generation. A dash means the metric was not available. The top 2 results of each metrics are **bolded** and underlined.

Model family		Regulatory LM		Diffusion model				DNA LM		Joint generation
Dimension	Metric	regLM	TACO	Ctrl-DNA	D3	DFM	DDSM	HyenaDNA	Evo2	**PEAC**
Function	Pair activity (log(RNA/DNA))↑	–	–	–	–	–	–	–	–	**-4.35**
	Promoter activity↑	1.491	1.512	**3.871**	1.268	0.826	–	0.863	1.216	2.694
	Enhancer activity↑	0.361	1.026	**2.595**	0.974	0.701	0.435	0.267	0.520	0.517
Control	Single-element target error↓	–	0.685	0.085	0.245	1	**0.002**	–	–	–
	Pair-level target error↓	–	–	–	–	–	–	–	–	**0.301**
Realism	Promoter GC content diff↓	0.004	0.022	0.116	0.195	0.074	0.102	0.157	0.107	**0.001**
	Promoter motif enrichment↑	**0.950**	0.546	0.821	0.033	0.728	0.514	0.606	0.889	0.926
	Promoter motif co-occurrence↓	**36.177**	2607.211	433.668	73.245	98.552	415.821	68.300	59.655	40.817
	Promoter motif spacing↓	0.0035	**0.0028**	0.0066	0.0721	0.0130	0.0067	0.0227	0.0288	0.0061
	Promoter FBD↓	**0.0010**	0.0164	0.0199	0.0076	0.0039	0.1119	0.0047	0.0031	0.0019
	Enhancer GC content diff↓	0.046	0.346	0.138	0.052	0.004	0.072	0.062	0.029	**0.00**
	Enhancer motif enrichment↑	0.653	0.584	0.633	0.230	**0.954**	0.733	0.702	0.782	0.813
	Enhancer motif co-occurrence↓	24.122	574.739	160.991	31.527	38.359	**18.739**	24.588	29.186	29.316
	Enhancer motif spacing↓	0.0157	0.0168	0.0158	0.0574	0.0102	**0.0063**	0.0200	0.0262	0.0184
	Enhancer FBD↓	0.0016	0.0901	0.0274	0.0031	**0.0010**	0.0031	0.0017	0.0016	0.0017
Novelty/diversity	Percent identity↓	0.268	0.325	0.298	**0.084**	0.090	0.091	0.287	0.295	0.274
	Discriminability↓	**0.855**	1.000	1.000	0.996	1.000	0.993	0.856	0.862	0.908
	Diversity↑	0.372	**0.789**	0.513	0.353	0.305	0.776	0.364	0.362	0.369

We systematically benchmark representative regulatory DNA generators, including regulatory language models (regLM, TACO, and Ctrl-DNA), diffusion- and flow-based models (D3, DFM, and DDSM), and general-purpose DNA language models (HyenaDNA and Evo2). This benchmark places PEAC in the broader single-element generation landscape and tests whether adding joint-

generation capability preserves standard sequence-quality properties. Performance was assessed across four dimensions: function, control, sequence realism, and novelty (Table 1).

Table 1 indicates that existing methods exhibit different strengths across evaluation dimensions. Regulatory language models generally achieve strong single-element predicted activity and controllability, while diffusion- and flow-based models show competitive sequence realism and diversity. DNA language models are more balanced but have weaker function scores. Based on these results and the target activity-control objective, we selected Ctrl-DNA as the upstream promoter teacher and D3 as the enhancer teacher for PEAC distillation.

PEAC maintains competitive single-element quality while adding a joint-generation capability absent from prior generators. At the individual-element level, PEAC closely reproduces promoter motif enrichment ($r = 0.926$), co-occurrence (40.817), and spacing (0.0061), while also achieving high enhancer motif enrichment ($r = 0.813$). At the global sequence-distribution level, its low promoter and enhancer FBD scores (0.0019 and 0.0017) indicate close agreement with real regulatory sequences in the embedding space used for FBD, while a reference identity of 0.274 suggests that the generated sequences are not simple copies of training examples. However, the discriminability of AUROC remains 0.908 (Table 1), which means that generated sequences are still easily separable from reference sequences under this diagnostic. We checked whether PEAC retains single-element function signals under external predictors (Appendix Table 4).

5.2 Activity-guided promoter-enhancer generation

Table 2: Direct paired-generation baseline trained on the same processed CAGE-derived promoter-enhancer pair rows. Target-bin fraction is the fraction assigned by the oracle to the requested activity bin.

Method	Condition	Mean	Low	Medium	High	Target frac.
PEAC	Low	-5.926	0.791	0.205	0.004	0.791
	Medium	-4.444	0.073	0.917	0.010	0.917
	High	-2.074	0.001	0.321	0.678	0.678
Direct concat	Low	-4.827	0.318	0.637	0.045	0.318
	Medium	-4.559	0.255	0.686	0.059	0.686
	High	-4.191	0.186	0.698	0.116	0.116

Previous generators always focus on individual design of promoter or enhancer, while not both together as one pair. Since no existing method performs this co-design, we construct the baseline by concatenating promoter and enhancer outputs from the two teacher models, and train this concatenated model on the same processed CAGE-derived pair rows as PEAC, with identical sequence lengths, activity-bin edges, and sampling budget. For each conditioning setting (low, medium, high), we generated 1,000 pairs and scored them with the trained CAGE-expression oracle to predict pair activity (examples are available in Appendix A.5).

Table 2 shows that joint generation is feasible and the resulting pairs can be steered toward the requested regime. PEAC places 79.1%, 91.7%, and 67.8% of low-, medium-, and high-conditioned pairs in the target bin, compared with 31.8%, 68.6%, and 11.6% for the direct-concatenation baseline.

PEAC's improvements arise from its explicit joint modeling: regulatory prior distillation provides realistic element-level sequence grammar; activity-conditioned initialization ties sequence likelihood to activity regimes before pair training; and cross-attention pair fusion aligns generated pairs to the reference distribution within each activity bin. Together, these components allow PEAC to shift generated

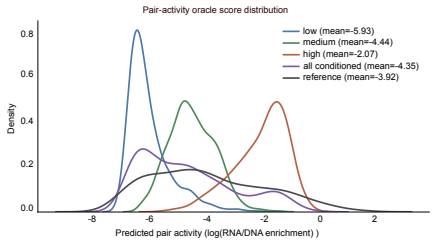

Figure 3: CAGE-derived oracle-predicted activity distributions for low-, medium-, high-, and all-conditionally generated promoter-enhancer pairs.

pairs into the requested activity regimes while preserving reference-like sequence statistics (Figure 3; Appendix A.5). Complementary post-hoc pairing controls and leakage-aware oracle checks further support this interpretation (Appendix A.3).

5.3 Ablation study

Table 3: Ablation results for the four architectural and training components. The full model row corresponds to the all-conditional PEAC model, and the reference row reports 1000 held-out pairs from the CAGE-derived pair-activity dataset. Each case generates all-conditional pairs, which are scored by the pair-activity oracle. FBD and motif enrichment correlations (r) are reported separately for promoters(P) and enhancers (E).

Model	Mean	High	P-GC	E-GC	P-FBD	E-FBD	P-r	E-r
Reference	-3.92	0.196	0.623	0.492	0.0000	0.0000	1.000	1.000
Full PEAC	-4.35	0.164	0.624	0.492	0.0019	0.0017	0.926	0.813
No distillation	-4.66	0.137	0.686	0.526	0.0074	0.0019	0.771	0.823
No initialization	-5.13	0.103	0.576	0.481	0.1020	0.0715	0.484	0.407
No pair fusion	-4.64	0.142	0.685	0.491	0.0106	0.0019	0.781	0.836
No latent MMD	-4.58	0.134	0.644	0.445	0.0027	0.0043	0.885	0.662

Table 3 examines the contributions of distillation, pretrained initialization, pair fusion, and latent-distribution alignment for all-conditional generation. The desired behavior is to preserve a realistic activity mixture while maintaining sequence realism. Full PEAC is closest to the reference high-activity fraction among the ablations and maintains strong promoter and enhancer realism.

Removing pretrained initialization causes the largest overall degradation, shifting the generated distribution toward lower oracle activity and causing a large drop in motif enrichment. This result indicates that the knowledge transferred from the pretrained promoter and enhancer generators provides an essential foundation for joint generation. Removing the student generators shifts the high-activity fraction farther from the reference and substantially degrades promoter realism. This suggests that the student representations are particularly important for preserving promoter features and adapting the pretrained generators to the pair-level objective. Removing pair fusion also shifts the high-activity fraction away from the reference and substantially worsens promoter realism. Similarly, removing the latent MMD objective shifts the high-activity fraction away from the reference and particularly decreases enhancer motif correlation from 0.813 to 0.662.

Overall, these ablations show that each of the four components contributes to the balance between controllability and realism. The distillation and initialization components provide a strong foundation for generating realistic sequences, while the pair fusion and latent MMD components help maintain the reference-like activity mixture and the realism of generated sequences.

Multi-seed ablations further support the same design rationale: full PEAC keeps the activity-bin proportions and realism stable across training and generation seeds, whereas removing individual components exposes complementary activity-control and realism failure modes (Appendix A.9).

6 Conclusion

PEAC addresses coarse activity-conditioned joint generation of promoter-enhancer pairs by combining STARR-seq-derived activity-conditioned initialization, FANTOM5 CAGE-derived expression conditioning, cross-attentive pair fusion with latent MMD alignment. The model shifts generated pairs toward requested CAGE-oracle-predicted activity regimes while maintaining selected reference-like sequence statistics.

The evidence supports oracle-predicted paired-activity regime control. Constructed-pair controls, ablations, and EPInformer diagnostics support this distributional interpretation and clarify the role of external model-based checks. The current model is still limited to single promoter-enhancer pairs and is trained primarily in the K562 setting. Real genomic regulation depends on multi-enhancer context, distance, 3D chromatin contacts, accessibility, and cell-type-specific TF state. Moreover, PEAC is a computational candidate-generation framework whose generated sequences require wet-lab validation before biological or therapeutic interpretation. Future work should extend PEAC to additional assays and cell types and close the loop with experimental validation.

AI USE STATEMENT

In preparing this submission, we used generative AI tools to assist with drafting and editing manuscript text for readability, improving LaTeX formatting, and checking that the paper includes the required disclosure language. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for or write mathematical proofs, propose or refine hypotheses, design the research methodology or experiments, implement the proposed method, translate the manuscript, clean or reformat datasets, support qualitative or thematic data analysis, or interpret experimental results. Tasks involving survey or interview question design and transcription of research recordings are not applicable to this work.

All AI-assisted text and formatting changes were reviewed by the authors. The authors take responsibility for the final content of this work, including text, claims, results, and artifacts produced with the aid of generative AI.

ACKNOWLEDGMENTS

The authors thank colleagues and lab members for helpful discussions and feedback throughout this project. We also acknowledge the creators and maintainers of the public regulatory genomics datasets and open-source software tools used in this work.

REPRODUCIBILITY STATEMENT

We document the complete implementation details in Appendix A.6. The loss weights and hyperparameters for the PEAC run is in Appendix Table 13 and 14. Code is available at `https://anonymous.4open.science/r/PEAC-review-B374`

REFERENCES

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation, 2023. URL https://arxiv.org/abs/2305.10699.

Lucía Barbadilla-Martínez, Noud Klaassen, Vinícius H. Franceschini-Santos, Jérémie Breda, Hatice Yücel, Miguel Hernández-Quiles, Tijs van Lieshout, Carlos G. Urzua Traslaviña, Minh Chau Luong Boi, Maryam Akbarzadeh, Celia Hermana-Garcia-Agullo, Sebastian Gregoricchio, Marcel de Haas, Roy Straver, Sarah Derks, Wilbert Zwart, Emile Voest, Lude Franke, Michiel Vermeulen, Jeroen de Ridder, and Bas van Steensel. Regulatory grammar in human promoters uncovered by mpra-based deep learning. *Nature*, 651(8107):1107–1116, February 2026. ISSN 1476-4687. doi: 10.1038/s41586-025-10093-z. URL http://dx.doi.org/10.1038/s41586-025-10093-z.

Drew T. Bergman, Thouis R. Jones, Vincent Liu, Judhajeet Ray, Evelyn Jagoda, Layla Siraj, Helen Y. Kang, Joseph Nasser, Michael Kane, Antonio Rios, Tung H. Nguyen, Sharon R. Grossman, Charles P. Fulco, Eric S. Lander, and Jesse M. Engreitz. Compatibility rules of human enhancer and promoter sequences. *Nature*, 607(7917):176–184, May 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-04877-w. URL http://dx.doi.org/10.1038/s41586-022-04877-w.

Sam Boshar, Benjamin Evans, Ziqi Tang, Armand Picard, Yanis Adel, Franziska K Lorbeer, Chandana Rajesh, Tristan Karch, Shawn Sidbon, David Emms, Javier Mendoza-Revilla, Fatimah Al-Ani, Evan Seitz, Yair Schiff, Yohan Bornachot, Ariana Hernandez, Marie Lopez, Alexandre Laterre, Karim Beguir, Peter Koo, Volodymyr Kuleshov, Alex Stark, Bernardo P de Almeida, and Thomas Pierrot. A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction. December 2025. doi: 10.64898/2025.12.22.695963. URL http://dx.doi.org/10.64898/2025.12.22.695963.

Garyk Brixi, Matthew G Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, Eric Nguyen, Chiara Ricci-Tam, David W Romero, Jonathan C Schmok, Ali Taghibakhshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K Wang, Michael T Pearce, Elana Simon, Etowah Adams, Zachary J Amador, Euan A Ashley, Stephen A Baccus, Haoyu Dai, Steven Dillmann, Stefano Ermon, Daniel Guo, Michael H Herschl, Rajesh Ilango, Ken Janik, Amy X Lu, Reshma Mehta, Mohammad R K Mofrad, Madelena Y Ng, Jaspreet Pannu, Christopher Ré, John St John, Jeremy Sullivan, Joseph Tey, Ben Viggiano, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Thomas McGrath, Kimberly Powell, Sudarshan Pinglay, Dave P Burke, Hani Goodarzi, Patrick D Hsu, and Brian L Hie. Genome modelling and design across all domains of life with evo 2. *Nature*, 652(8112):1349–1361, April 2026.

Xingyu Chen, Shihao Ma, Runsheng Lin, Jiecong Lin, and Bo Wang. Ctrl-dna: Controllable cell-type-specific regulatory dna design via constrained rl, 2025. URL https://arxiv.org/abs/2505.20578.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, February 2021. ISSN 1367-4811. doi: 10.1093/bioinformatics/btab083. URL http://dx.doi.org/10.1093/bioinformatics/btab083.

Seungsoo Kim and Joanna Wysocka. Deciphering the multi-scale, quantitative cis-regulatory code. *Molecular Cell*, 83(3):373–392, February 2023. ISSN 1097-2765. doi: 10.1016/j.molcel.2022.12.032. URL http://dx.doi.org/10.1016/j.molcel.2022.12.032.

Avantika Lal, David Garfield, Tommaso Biancalani, and Gokcen Eraslan. Designing realistic regulatory dna with autoregressive language models. *Genome Research*, 34(9):1411–1420, 2024. ISSN 1549-5469. doi: 10.1101/gr.279142.124. URL http://dx.doi.org/10.1101/gr.279142.124.

Jiecong Lin, Zhijian Li, Yajie Zhao, Ruibang Luo, and Luca Pinello. Epinformer: scalable and integrative prediction of gene expression from promoter-enhancer sequences with multimodal epigenomic profiles. *Nature Communications*, 17(1), March 2026. ISSN 2041-1723. doi: 10.1038/s41467-026-70535-8. URL http://dx.doi.org/10.1038/s41467-026-70535-8.

Junhao Liu, Pengpeng Zhang, Siwei Xu, Shushrruth Sai Srinivasan, Yongxian Wu, and Jing Zhang. Multimodal cell context instruction tuning for conditional dna regulatory sequence generation with large language models. In *2025 IEEE International Conference on Image Processing (ICIP)*, pp. 989–994. IEEE, 2025. doi: 10.1109/icip55913.2025.11084625. URL http://dx.doi.org/10.1109/icip55913.2025.11084625.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, Stefano Ermon, Stephen A. Baccus, and Chris Ré. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution, 2023. URL https://arxiv.org/abs/2306.15794.

Liwen Ouyang and Aaron Key. Maximum mean discrepancy for generalization in the presence of distribution and missingness shift. November 2021.

Anirban Sarkar, Alejandra Duran, Yiyang Yu, Da-Wei Lin, Yijie Kang, Nirali Somia, Pablo Mantilla, Jessica Zhou, Masayuki Nagai, Ziqi Tang, Kaarina Hanington, Kenneth Chang, and Peter K. Koo. Designing dna with tunable regulatory activity using discrete diffusion. May 2024. doi: 10.1101/2024.05.23.595630. URL http://dx.doi.org/10.1101/2024.05.23.595630.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design, 2024. URL https://arxiv.org/abs/2402.05841.

Patricia J. Wittkopp and Gizem Kalay. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. *Nature Reviews Genetics*, 13(1):59–69, December 2011. ISSN 1471-0064. doi: 10.1038/nrg3095. URL http://dx.doi.org/10.1038/nrg3095.

Zhao Yang, Bing Su, Chuan Cao, and Ji-Rong Wen. Regulatory dna sequence design with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.07981.

Pengcheng Zhang, Haochen Wang, Hanwen Xu, Lei Wei, Liyang Liu, Zhirui Hu, and Xiaowo Wang. Deep flanking sequence engineering for efficient promoter design using deepseed. *Nature Communications*, 14(1), October 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-41899-y. URL http://dx.doi.org/10.1038/s41467-023-41899-y.

A Additional Details

A.1 Dataset preprocessing details

For activity-conditioned initialization, we use the K562 STARR-seq-derived reporter resource as an element-level supervision source. From this resource we retain rows with usable promoter and enhancer DNA sequences together with the available `promoter_activity` and `enhancer_activity` annotations. These labels are used to make the distilled autoregressive prior activity-aware before final CAGE-conditioned pair-level training. We do not treat this stage as providing the final PEAC promoter-enhancer target, because the STARR-seq reporter output and the processed CAGE-derived candidate-row label used below are distinct measurements.

All sequence fields are first normalized to a common DNA alphabet by uppercasing and removing non-DNA characters; remaining non-`ACGT` bases are encoded as `N`. The generator uses fixed windows matching the upstream teacher interfaces: 250 bp for promoters and 230 bp for enhancers. When rebuilding tables from longer source sequences, we center-truncate each sequence to the target length, preserving the middle of the promoter or enhancer context:

$$s_{\text{trunc}} = s_{\lfloor (|s|-L)/2 \rfloor : \lfloor (|s|-L)/2 \rfloor + L}.$$

Sequences shorter than the target window are kept and padded by the tokenizer during fixed-length encoding. Thus all downstream models see promoter and enhancer tensors with the same lengths, while the source sequence context is kept as centered as possible.

For final CAGE-conditioned pair-level training and for the direct-concat baseline, we use the processed K562 FANTOM5 candidate-pair table. Rows without usable promoter sequence, enhancer sequence, or CAGE-derived expression target are removed. The resulting non-control candidate-pair table contains 545,453 rows, split into 436,363 training rows, 54,545 validation rows, and 54,545 held-out test rows. The generator is fit on the training split, while the validation and held-out test rows are used for oracle selection and reliability checks.

The supervised target used in this stage is copied from the processed candidate-pair table rather than recomputed from raw FANTOM5 CAGE tag counts. Specifically, the training loader reads the processed row-level scalar as `pair_activity`, falling back to `gene_expression_level` when `pair_activity` is absent. Runs that condition on `pair_activity_log1p` use the explicit processed `pair_activity_log1p` value when it is present; the fallback formula $\log(1+\max(y,0))$, where y is `pair_activity`, is used only for processed tables that do not already provide this column. In the experiments reported here, the values should therefore be read as processed CAGE-derived candidate-row expression scores on the table's native scale. They are not interchangeable with the $\log_2(\mathrm{RNA}/\mathrm{DNA})$ reporter enrichment used in STARR-seq assays. Negative values are expected because the processed expression-proxy scale is centered below zero for many candidate rows, not because activity measurements are invalid. Coarse activity regimes for balanced training, conditional sampling, and oracle-bin evaluation use the shared processed CAGE-derived thresholds -5.3, -2.75, and -0.35, corresponding to low, medium, and high bins. The main generation experiments sample low, medium, high, and all-condition settings.

A.2 Metric and baseline details

The metrics in Table 1 are produced by the evaluation scripts under `RegSeq/eval`. Unless otherwise stated, each row summarizes 1,000 generated sequences or pairs from the corresponding method.

Pair activity. Pair activity is the mean pair-level processed CAGE-expression proxy score returned by the promoter-enhancer oracle for methods that generate paired promoter-enhancer designs. The oracle takes the promoter and enhancer jointly as input and reports a scalar pair-expression prediction for each pair; the table reports the arithmetic mean across scored generated pairs.

Promoter activity. Promoter activity is the mean scalar score returned by the PARM promoter activity evaluator over generated promoter sequences. The main benchmark table reports the mean over available PARM cell-type models so that PEAC is compared to the other promoter generators on the same PARM scope. K562-specific PARM scores are reported separately in Table 4 and should not be directly ranked against multi-cell PARM means.

Enhancer activity. Enhancer activity is the mean scalar score returned by the regLM human enhancer activity evaluator over generated enhancer sequences. Unless a K562-only value is explicitly reported, the score is the mean of the HEPG2, K562, and SKNSH regLM enhancer predictions, followed by an arithmetic mean across generated enhancers.

Single-element target error. Single-element target error is the mean absolute error between a requested single-element activity target and the corresponding oracle-predicted activity. The control evaluator reads target activity columns from the generated file, scores each promoter or enhancer with the appropriate function oracle, computes $|\hat{y}_i - y_i|$ for each scored sequence, and reports the mean absolute error.

Pair-level target error. Pair-level target error is the mean absolute error between the requested pair-expression-proxy target and the CAGE-expression-proxy oracle prediction for generated promoter-enhancer pairs. The metric is computed as the average of $|\hat{y}_i - y_i|$ over scored pairs with a specified target activity.

Promoter GC content. Promoter GC content is the mean per-sequence GC fraction over generated promoter sequences. For each sequence, the evaluator counts `G` and `C` bases and divides by the number of canonical `A,C,G,T` bases; noncanonical bases are not counted in the denominator. The table reports the mean GC fraction, and values closer to the reference promoter GC mean (0.624) are preferable.

Enhancer GC content. Enhancer GC content is computed in the same way as promoter GC content, but on generated enhancer sequences. The table reports the mean enhancer GC fraction, and values closer to the reference enhancer GC mean (0.492) are preferable.

Promoter motif enrichment. Promoter motif enrichment measures the agreement between generated and reference promoter motif composition. Motifs are scanned on both strands with FIMO using the prepared JASPAR 2026 CORE vertebrate non-redundant MEME database and the FIMO threshold $\texttt{--thresh} = 10^{-4}$ unless precomputed FIMO tables are supplied. For each motif, counts are summed across generated promoters and across reference promoters, and the metric is the Pearson correlation between the two total motif-count vectors over the shared motif universe.

Enhancer motif enrichment. Enhancer motif enrichment uses the same FIMO scanning and Pearson-correlation calculation as promoter motif enrichment, but the generated and reference sequence sets are enhancers.

Promoter motif co-occurrence. Promoter motif co-occurrence is the Frobenius norm between generated and reference promoter motif-count covariance matrices. Each promoter is represented by a vector of motif counts over the shared motif universe; the evaluator computes a covariance matrix across generated promoters and a covariance matrix across reference promoters, then reports the Frobenius norm of their difference.

Enhancer motif co-occurrence. Enhancer motif co-occurrence is computed identically to promoter motif co-occurrence, but motif-count covariance matrices are formed from generated and reference enhancer sequences.

Promoter motif spacing. Promoter motif spacing compares the distribution of adjacent motif-hit distances between generated and reference promoters. Within each sequence, FIMO hits are sorted by start coordinate and adjacent gaps are computed as $\max(0, start_{i+1} - stop_i - 1)$. Gaps are binned in 10 bp bins up to 500 bp plus an overflow bin, histograms are normalized, and the metric is one-half of the L1 distance between generated and reference spacing histograms.

Enhancer motif spacing. Enhancer motif spacing uses the same adjacent FIMO-hit gap calculation and histogram distance as promoter motif spacing, but on generated and reference enhancer sequences.

Promoter FBD. Promoter FBD is the Fréchet Biological Distance between generated and reference promoter feature distributions. In this implementation, each promoter is represented by a normalized 4-mer frequency vector over `A,C,G,T`; 4-mers containing `N` are dropped and the vector is divided by the number of valid 4-mer windows. The Fréchet distance is computed from the empirical means and covariance matrices of generated and reference 4-mer feature vectors. No deep feature extractor, learned embedding, or additional whitening is used. Therefore, low FBD supports only the claim

that generated and reference sets have similar normalized 4-mer mean and covariance statistics; it does not by itself establish overall biological realism.

Enhancer FBD. Enhancer FBD is computed with the same normalized 4-mer Fréchet-distance procedure as promoter FBD, but using generated and reference enhancer sequences.

Percent identity. Percent identity is the mean nearest-reference normalized Hamming similarity from each generated sequence to the reference set. For each generated sequence, the evaluator computes position-wise similarity to every reference sequence in the same orientation, using the longer sequence length as the denominator, and records the maximum similarity. This denominator choice can make longer or differently formatted outputs appear more novel by construction. We therefore treat the main-table percent-identity values as diagnostic rather than as a cross-method ranking. A length-controlled novelty analysis should compare fixed-format sequences and report promoter and enhancer results separately. For PEAC all-condition generation, the separated diagnostics are promoter percent identity 0.347 and enhancer percent identity 0.302, computed against the corresponding fixed-format reference sets. Paired promoter-enhancer records can also be represented as promoter sequence, 20 `N` separator bases, and enhancer sequence, but this concatenated format is not used as a definitive cross-method novelty ranking. Indel alignment and reverse-complement matching are not used.

Discriminability. Discriminability is the cross-validated AUROC of a logistic-regression classifier trained to distinguish generated sequences from reference sequences using normalized 4-mer frequency features. The evaluator uses stratified folds, standardizes features within each training fold, trains the classifier, and reports the generated-versus-real AUROC. Values closer to 0.5 indicate lower separability, whereas values near 1.0 indicate that generated and reference sequences remain easily distinguishable by 4-mer composition.

Within-set nearest-neighbor similarity. The diversity diagnostic is reported from the novelty evaluator as the mean nearest-neighbor normalized Hamming similarity within the generated set. For each generated sequence, the evaluator finds the most similar other generated sequence under the same position-wise normalized Hamming definition used for percent identity, then averages these nearest-neighbor similarities across the generated set. Because the reported statistic is a similarity, lower values indicate greater within-set diversity under this diagnostic. The main table therefore reports it as nearest-neighbor similarity, not as a higher-is-better diversity score.

The single-element function oracles in Table 1 and Table 4 are PARM for promoter activity and the regLM enhancer activity model for enhancer activity. PARM values use the K562 model or the mean over available PARM cell-type models as stated in Table 4; the enhancer oracle averages HEPG2, K562, and SKNSH regLM enhancer predictions unless a K562-only value is explicitly reported. These predictors evaluate single regulatory elements, while the pair-level processed CAGE-expression proxy is scored with the oracle described in Appendix A.3.

Baseline sequences in Table 1 use the available method checkpoints and generation settings for each original model rather than a single shared architecture. Unless otherwise stated, each method contributes 1,000 generated sequences for the relevant element type. The PEAC rows use the K562 pair checkpoint for the reported all-condition run, sampled with promoter length 250, enhancer length 230, temperature 0.95, top-$k = 4$, batch size 8, seed 42, and condition bins low, medium, high, or all as indicated. The direct concatenated baseline uses the same FANTOM5 CAGE-derived pair rows, the same sequence lengths, the same activity-bin edges, and the same per-condition sampling budget. The Ctrl-DNA/D3 constructed-pair controls use cached K562 Ctrl-DNA promoter and D3 enhancer teacher outputs when available; rebuilding uses the K562 Ctrl-DNA promoter corpus builder and D3 transformer enhancer generator with activity grid 0, 1,000 samples, D3 transformer architecture, K562 cell-type id 0, and seed 42. The random control pairs independently sampled promoters and enhancers. The activity-matched control pairs elements after single-element activity matching. The oracle-reranked control samples 10 D3 enhancers per Ctrl-DNA promoter and selects the pair with the largest CAGE-oracle activity score.

Table 4: External single-element function checks for generated and reference regulatory elements. Promoter activity is predicted with PARM (Barbadilla-Martínez et al., 2026); K562 scores use the K562 PARM model, and mean scores average the available PARM cell-type models. Enhancer activity is predicted with the Enformer enhancer activity model. These are single-element diagnostics and are not used as pair-activity or pair-function oracles.

Set	n	Element / predictor	Mean	Median	Min	Max
Real K562 reference	1000	Promoter / PARM K562	6.107	6.372	1.167	14.856
Ctrl-DNA teacher	1000	Promoter / PARM mean	3.871	2.994	0.833	7.994
PEAC all-condition	1000	Promoter / PARM K562	2.694	2.333	1.032	13.042
PEAC all-condition	1000	Promoter / PARM mean	1.445	1.236	0.620	7.179
PEAC promoter student	1000	Promoter / PARM mean	1.728	1.326	0.581	7.025
regLM	1000	Promoter / PARM mean	1.491	1.172	0.627	10.467
Ctrl-DNA baseline	1000	Enhancer / Enformer K562	2.595	2.460	0.817	4.866
D3 teacher	1000	Enhancer / Enformer mean	0.974	0.623	-1.528	6.312
PEAC enhancer student	1000	Enhancer / Enformer mean	0.877	0.548	-0.759	6.299
Real K562 reference	2000	Enhancer / Enformer mean	0.660	0.368	-1.020	7.525
PEAC all-condition	1000	Enhancer / Enformer K562	0.517	0.232	-0.729	5.875
PEAC all-condition	1000	Enhancer / Enformer mean	0.462	0.265	-0.801	4.832
regLM	1000	Enhancer / Enformer mean	0.361	0.261	-0.644	4.502

A.3 Oracle validation and reliability

The CAGE-expression oracle used for generation-time evaluation was trained only on the CAGE-derived training split and selected on the validation split. The oracle takes the promoter and enhancer jointly as input and predicts the CAGE expression label assigned to each candidate row. We evaluate oracle reliability on held-out reference rows before using it to score generated sequences. After preprocessing, the expression-labeled non-control candidate-pair dataset contains 545,453 candidate rows. The training split contains 436,363 rows, and the held-out validation and test splits each contain 54,545 rows. We additionally trained a stronger cross-fusion oracle with convolutional motif encoders, lightweight Transformer encoders, promoter-enhancer cross-attention, and regression/bin/ordinal supervision. The original CNN oracle remains the evaluation oracle used for the main tables; the stronger oracle and a CNN/cross-fusion ensemble are used here to estimate the robustness and practical ceiling of oracle-based bin assignment.

Table 5: Held-out performance of CAGE-expression oracles. Activity metrics are computed against CAGE-derived expression labels assigned to candidate rows. Bin accuracy uses the shared activity-bin edges -5.3, -2.75, and -0.35; adjacent accuracy counts predictions within one neighboring bin as correct.

Oracle	Split	n	MAE	RMSE	Pearson	Spearman	Bin / adjacent
CNN	Validation	54,545	0.828	1.145	0.863	0.866	– / –
CNN	Test	54,545	0.824	1.145	0.864	0.868	0.773 / 0.993
Cross-fusion	Validation	54,545	0.835	1.151	0.861	0.864	0.770 / 0.993
Cross-fusion	Test	54,545	0.831	1.151	0.862	0.866	0.771 / 0.993

The oracle preserves the rank ordering of the processed pair-expression proxy on held-out data (CNN test Spearman $= 0.868$) and gives similar validation and test performance, supporting generalization beyond the validation split (Table 5). The stronger cross-fusion oracle produced similar exact held-out bin accuracy, consistent with the discretization of a noisy continuous processed CAGE-derived expression-proxy label into hard bins. In contrast, adjacent-bin accuracy is consistently about 0.993, showing that the oracle almost never confuses distant activity regimes. We therefore use the oracle primarily for relative activity ordering and target-bin enrichment.

The PEAC evaluation setting uses short synthetic promoter-enhancer pairs scored for CAGE-derived expression. We therefore use the self-trained oracle for the primary target-regime analysis and add

EPInformer as an external, orthogonal CAGE-like sanity check (Lin et al., 2026). EPInformer is trained for endogenous gene-expression prediction from promoter-enhancer sequence and auxiliary features, providing a useful test of whether generated pairs remain plausible under a different promoter-enhancer model.

To assess whether the conditional hit rates depend on the original oracle, we rescored the same PEAC-generated low-, medium-, high-, and all-condition samples using three independently trained three-bin CAGE-expression proxy oracles with pair-disjoint, promoter-disjoint, and enhancer-disjoint splits. The generated sequences were fixed; only the evaluation oracle was changed. Table 7 reports target-bin hit rates for the three explicitly conditioned PEAC sets. We also rescored the direct-concat baseline samples with the same three oracles to test whether the PEAC-versus-baseline comparison transfers across oracle choices (Table 8).

Table 6: Leakage-controlled three-bin CAGE-expression oracle validation. All metrics are computed on held-out test pairs. Pair overlap, promoter overlap, and enhancer overlap report train-test overlap counts under the split definition. Adjacent accuracy counts predictions within one neighboring activity regime as correct.

Split	Test n	MAE	RMSE	Pearson	Spearman	Bin acc.	Adj. acc.	Train-test overlap
Pair-disjoint	43,636	0.831	1.153	0.862	0.865	0.771	0.996	pair 0; prom. 777; enh. 813
Promoter-disjoint	44,709	1.294	1.680	0.683	0.683	0.626	0.991	pair 0; prom. 0; enh. 813
Enhancer-disjoint	42,953	1.278	1.651	0.701	0.696	0.641	0.992	pair 0; prom. 777; enh. 0

Across the three disjoint oracles, PEAC medium- and high-conditioned samples retain high target-bin assignment rates, with majority-vote hit rates of 0.775 and 0.782, respectively. The low-conditioned PEAC set is much less stable: majority-vote low-bin assignment is 0.351, a 44 percentage-point decrease from the primary-oracle value of 0.791. Table 8 shows that independent-oracle rescoring supports a PEAC advantage over direct concat for the high condition, but not for every target condition: the direct-concat medium hit rate is higher under majority vote. Thus this analysis demonstrates partial transfer of the PEAC conditioning behavior to independent oracles, not a complete independent-oracle reproduction of the original PEAC-versus-baseline ranking. The low-condition boundary check (Table 9) suggests one reason for the sensitivity: many low-conditioned samples are scored close to the -5.3 low/medium boundary or just above it, so small differences in oracle calibration move them into the medium bin.

Table 7: Conditional target-bin hit rates after rescoring the same PEAC-generated samples with leakage-controlled oracles. "Majority" is the per-sample majority vote across the three disjoint oracles.

Generated condition	Pair-disjoint	Promoter-disjoint	Enhancer-disjoint	Majority
Low	0.355	0.348	0.355	0.351
Medium	0.779	0.742	0.770	0.775
High	0.804	0.722	0.791	0.782

Table 8: Primary-oracle and independent-oracle target-bin hit rates for PEAC and the direct-concat baseline. The same generated samples are rescored by the same pair-disjoint, promoter-disjoint, and enhancer-disjoint oracles. "Majority" is the per-sample majority vote across the three independent oracles.

Method	Condition	Primary oracle	Pair-disjoint	Promoter-disjoint	Enhancer-disjoint	Majority
PEAC	Low	0.791	0.355	0.348	0.355	0.351
Direct concat	Low	0.318	0.282	0.291	0.251	0.262
PEAC	Medium	0.917	0.779	0.742	0.770	0.775
Direct concat	Medium	0.686	0.824	0.826	0.860	0.851
PEAC	High	0.678	0.804	0.722	0.791	0.782
Direct concat	High	0.116	0.144	0.119	0.134	0.120

Table 9: Low-condition score distribution under independent oracles. Fractions are computed from continuous predicted scores relative to the shared low/medium boundary at -5.3. "Within 0.5" is the fraction with score in $[-5.8,-4.8]$, showing how much of the low-conditioned set lies near the decision boundary.

Method	Oracle	Mean score	Score < -5.3	Score ≥ -5.3 and < -2.75	Within 0.5 of -5.3
PEAC	Pair-disjoint	-5.067	0.463	0.505	0.327
PEAC	Promoter-disjoint	-4.863	0.385	0.583	0.429
PEAC	Enhancer-disjoint	-5.027	0.447	0.520	0.305
Direct concat	Pair-disjoint	-4.906	0.403	0.561	0.363
Direct concat	Promoter-disjoint	-4.704	0.317	0.650	0.408
Direct concat	Enhancer-disjoint	-4.814	0.361	0.608	0.339

A.4 External EPInformer sanity check

We used EPInformer as an external plausibility diagnostic under a model trained for endogenous gene-expression prediction. Holding each PEAC promoter fixed, we compared its paired PEAC enhancer against 10 enhancers from another source and ranked candidates by mean promoter-to-enhancer attention.

We firstly checked for the shuffled enhancers (i.e. 10 enhancers from other generated pairs). The EPInformer pair attention and pair activity outputs show that the balanced results between matched pairs and shuffled pairs. We confirm this result is not directly related to failure in pair generation by add reference pairs shuffle results (Table 10).

The PEAC enhancer ranked first for most comparison sources (Appendix Table 11). Top-1 retrieval was highest against HyenaDNA, DDSM, and regLM enhancers (0.996, 0.991, and 0.986) and remained high for Evo2, Ctrl-DNA, DFM, TACO, and D3 enhancers (0.876–0.959). Against reference enhancers, top-1 retrieval was 0.230, while PEAC still won 64.2% of pairwise comparisons. Matched-versus-shuffled controls showed no detectable attention difference for either PEAC or reference pairs, so we use this analysis as a cross-source plausibility check.

Table 10: EPInformer external sanity check for matched and shuffled pairs. Pair activity is the EPInformer K562 CAGE-like expression prediction, and pair attention is the mean promoter-to-enhancer attention score exposed by the EPInformer wrapper. Matched-minus-shuffled deltas are computed pairwise. EPInformer attention is used as a post-hoc diagnostic for promoter-enhancer pairing behavior and is interpreted together with the matched and shuffled reference rows.

Set	Metric	Matched mean	Shuffled mean	Mean delta	Matched $>$ shuffled
PEAC	Pair activity	1.082747	1.082763	-0.000016	0.501
PEAC	Pair attention	0.412321	0.412208	0.000113	0.503
Real K562	Pair activity	1.325910	1.326299	-0.000390	0.493
Real K562	Pair attention	0.484168	0.483945	0.000223	0.509

Table 11: EPInformer-based cross-source enhancer retrieval. Results are reported across 1,000 groups per comparison source. Each group contains one PEAC promoter, its PEAC enhancer, and 10 comparison enhancers from the listed source.

Comparison source	Paired top-1	Pairwise win frac.
Reference	230/1000	0.642
D3	876/1000	0.986
DDSM	991/1000	0.999
DFM	931/1000	0.993
Evo2	959/1000	0.996
regLM	986/1000	0.999
TACO	905/1000	0.989
Ctrl-DNA	957/1000	0.982
HyenaDNA	996/1000	1.000

A.5 Qualitative generated examples

Table 12 lists representative generated promoter-enhancer pairs from the low-, medium-, and high-condition runs. For compactness, each sequence is shown as a 5' and 3' excerpt; the full 250 bp promoter and 230 bp enhancer sequences are used for all oracle and realism evaluations. The examples illustrate the activity shift observed quantitatively: the high-condition example receives a substantially higher pair-activity score than the low-condition example, while GC content remains in the range observed for regulatory sequences.

Table 12: Representative generated promoter-enhancer examples. Sequence excerpts show the first and last 10 bp of each generated element.

Condition	Oracle score	Pred. bin	P/E GC	Promoter excerpt	Enhancer excerpt
Low	-6.870	Low	0.412 / 0.443	`AAAACATGAT...CCTTGGTAAT`	`GTTTATTTCC...TGAGTTGCAG`
Medium	-4.518	Medium	0.592 / 0.409	`AAAACGCCCT...CCTCGGTCCT`	`CTTTTTTTCC...TTAGTTACTG`
High	-2.019	High	0.616 / 0.543	`GAAACGCGCT...CGTTGGTCCT`	`CTTTGTTTGC...TGAGTCACAG`

A.6 Implementation details

PEAC uses fixed 250 bp promoter windows and 230 bp enhancer windows. The autoregressive input is the concatenation of a beginning-of-sequence token, the promoter tokens, a separator token, the enhancer tokens, and an end-of-sequence token, giving a maximum length of 483 tokens. The DNA vocabulary contains `A`, `C`, `G`, `T`, `N`, and special tokens. The base type-conditioned generator uses token, position, promoter/enhancer type, activity-condition, activity-bin, and activity-quantile embeddings. It is a 6-layer causal Transformer with hidden dimension 256, 8 attention heads, feed-forward dimension 1024, dropout 0.1, and per-bin residual adapters of dimension 64. The language-model head is tied to the token embedding, with additional zero-initialized activity-bin residual LM heads.

The initialization model wraps the base generator with promoter and enhancer motif encoders. Each motif encoder embeds tokens into 256 dimensions, applies parallel 1D convolutions with kernel widths 3, 7, and 15, projects the pooled features to a 128-dimensional motif representation, and concatenates this with the pooled Transformer representation for each element. A promoter-to-enhancer multi-head attention layer produces a pair context. The pair representation concatenates promoter, enhancer, pair-context, elementwise product, absolute context difference, and the 3-dimensional condition vector before the pair-level heads.

The final model starts from the initialization checkpoint and adds two PairFusion blocks. Each block applies promoter self-attention, enhancer self-attention, promoter-to-enhancer and enhancer-to-promoter cross-attention, condition attention, pair-token attention over promoter/enhancer/condition tokens, activity-bin FiLM modulation, and feed-forward updates. The fused promoter, enhancer, and pair-token states are projected to 16-dimensional latents. These three latents are concatenated for pair-activity regression and activity-bin classification; the latent vector used for distribution alignment also includes the predicted pair activity. Conditional MMD uses RBF bandwidths 0.2, 0.5, 1.0, 2.0, and 5.0.

In the main training runs, the `--freeze-base-student` option is not used. The autoregressive student, pair-fusion blocks, condition projections, FiLM/adaptor parameters, and activity/bin heads are therefore jointly updated. The model also supports freezing the base student for ablation-style runs, but that is not the default setting reported in the main results. Stop-gradient is used for supervision targets and for the reference-batch mean and standard deviation used to standardize MMD features. Gradients from the activity, bin-classification, activity-distribution, and MMD losses otherwise flow through the conditional pair encoder and, for soft generated sequences, back to the generator logits.

The soft generated sequences used during final-stage training are not self-rollout samples. They are obtained by applying Gumbel-softmax to the teacher-forced LM logits for the promoter and enhancer positions in the current training batch. Autoregressive sampling is used at generation time, so the training relaxation is an auxiliary gradient path that approximates generated sequence feedback under the observed prefix context.

A.7 Shortcut risks in condition-aware pair losses

The final-stage pair encoder receives both sequence-derived token states and explicit target-condition inputs, including the scalar condition, activity bin, and activity quantile. Because the activity-regression head, bin-classification head, and MMD latents are computed from this condition-aware representation, the auxiliary objectives have a potential shortcut: the heads may partially recover the requested target from the supplied condition rather than from the promoter-enhancer sequence. Similarly, because generated and reference latents are encoded by the same jointly updated pair encoder, the MMD term could shrink if both sides organize primarily by condition tokens. This is a possible degenerate path, not evidence that the reported model failed in this way.

The present manuscript partially probes this risk through no-pair-fusion, no-activity-bin-conditioning, and no-MMD ablations, plus external oracle rescoring. These checks are useful but not sufficient to prove that pair-fusion and MMD losses constrain sequence-dependent function. A stricter diagnostic suite should include: removing condition tokens from the auxiliary pair encoder while keeping them in the generator; freezing a separately trained condition-blind evaluator, or an evaluator whose target-condition input is fixed or masked, for activity and bin losses; perturbing or shuffling promoter/enhancer sequences under a fixed target condition and verifying that the auxiliary heads and latents change accordingly; and evaluating MMD with a fixed or sequence-only encoder on full autoregressive rollouts rather than only on Gumbel-softmax relaxations from teacher-forced logits. Until such controls are run, we treat the pair-fusion and MMD objectives as helpful regularizers whose sequence dependence is supported indirectly, not as independently validated functional constraints.

Table 13: Loss weights used for PEAC training. The initialization stage adapts the pretrained autoregressive prior with STARR-seq-derived single-element activity supervision before final pair-level training; CAGE-derived expression labels enter during the final pair-level stage. The final stage uses the checkpoint metadata from the model used for the main results.

Stage	Loss term	Weight
Initialization	Language modeling	1.0
	Promoter activity Huber	0.3
	Enhancer activity Huber	0.3
	Condition contrastive	0.8
	Condition contrastive margin	0.25
	Auxiliary pair-score / enhancer contribution terms	0.0
Final	Language modeling	0.15
	Pair activity Huber	2.0
	Conditional latent MMD	1.0
	Activity distribution matching	0.2
	Activity mean matching	0.05
	Condition contrastive	0.8
	Condition contrastive margin	0.25
	Activity-bin classification	0.5
	Enhancer contribution	0.0

Table 14: Training and sampling hyperparameters for the main PEAC runs.

Stage	Hyperparameter	Value
Initialization	Epochs	2
	Batch size	8
	Student learning rate	1×10^{-5}
	Head learning rate	5×10^{-5}
	Weight decay	1×10^{-4}
	Validation fraction	0.05
	Maximum train steps per epoch	2000
	Maximum validation steps	100
	Monitor sampling temperature / top-k	0.95 / 4
Final	Epochs	6
	Best checkpoint epoch	5
	Batch size	8
	Learning rate	5×10^{-5}
	Weight decay	1×10^{-4}
	Validation fraction	0.05
	Fusion layers / latent dimension	2 / 16
	Activity-bin edges	-5.3, -2.75, -0.35
	Soft-token temperature	0.8
	Gumbel-softmax	hard
	Balanced bin batches	yes
	Maximum steps per epoch	1200
	Deterministic training / seed	yes / 42
Sampling	Number of generated pairs per setting	1000
	Batch size	8
	Temperature	0.95
	Top-k	4
	Condition bins	low, medium, high, all
	Activity-bin edges	-5.3, -2.75, -0.35
	Seed	42

A.8 CONSTRUCTED PAIR BASELINES

To separate native joint generation from pair construction, we compared PEAC with Ctrl-DNA promoter and D3 enhancer baselines constructed by random pairing, activity matching, and CAGE-oracle-assisted reranking (Table 15). All methods were evaluated with the same CAGE-expression oracle. This analysis tests whether simple post-hoc pairing can explain PEAC's pair-level behavior. Since PEAC targets conditional activity distributions, the primary comparison uses distance to the reference activity mixture together with mean activity and high-activity fraction as diagnostics.

Table 15: Constructed promoter–enhancer pair baselines. Mean activity and high fraction are reported as diagnostics of the generated distribution alongside the target-distribution metrics. Bin L1 and JS measure distance between each generated three-regime activity distribution and the reference activity distribution; lower values are better. Random and activity-matched baselines pair independently generated Ctrl-DNA promoters and D3 enhancers without pair-level training. The oracle-reranked baseline samples 10 enhancers per promoter and selects the highest CAGE-derived oracle score, so it represents an oracle-assisted activity-maximizing control.

Method	Mean	High	Above real P75	Bin L1 ↓	Bin JS ↓	Note
PEAC	-4.354	0.164	0.157	0.148	0.0075	native
Ctrl-DNA + D3 random	-4.061	0.148	0.127	0.230	0.0121	random
Ctrl-DNA + D3 activity-matched	-3.289	0.272	0.217	0.320	0.0538	matched
Ctrl-DNA + D3 oracle-reranked	-2.742	0.467	0.437	0.458	0.0713	reranked

Post-hoc pairing provides a useful control for interpreting pair-level oracle scores. Random Ctrl-DNA/D3 pairing reaches a similar high-activity fraction to PEAC and has little matched-versus-shuffled separation under this diagnostic control (original-minus-shuffled mean, -0.008; original $>$ shuffled fraction, 0.493). Because real reference pairs also show limited matched-versus-shuffled separation under the same style of analysis, we treat this control as a guard against overinterpreting binary pair matching. Activity matching and oracle-assisted reranking deliberately push the constructed set toward higher oracle activity by selecting high-activity single elements or by using the evaluation oracle itself. Their higher mean activity and high-activity fractions therefore show that post-hoc construction can bias a generated set toward the high regime; under PEAC's objective, this corresponds to a different selection criterion from requested or reference-like activity-mixture recovery. Consistent with this interpretation, PEAC has the smallest distance to the reference three-regime activity distribution among these controls (Bin L1 $= 0.148$, JS $= 0.0075$), whereas activity matching and oracle reranking move farther from the reference mixture despite increasing mean activity. Activity matching likewise has negligible original-versus-shuffled separation (-0.006; 0.512), indicating that its shift is largely explained by single-element activity with limited promoter–enhancer matching signal. Oracle-assisted reranking represents an oracle-dependent activity-maximizing post-processing baseline. A complementary post-hoc baseline optimized directly for target-bin membership or target activity proportions would test a different selection objective. These results clarify that Appendix Table 1 serves as benchmark context; pair-level conclusions are based on explicit constructed-pair controls, target-distribution evaluation, and independent validation.

A.9 MULTI-SEED ABLATION ROBUSTNESS

We repeated the all-condition ablation analysis across three training seeds $(1, 2, 3)$ and three generation seeds $(42, 43, 44)$. Each training-seed and generation-seed combination produced 1,000 promoter-enhancer pairs, giving nine runs and 9,000 generated pairs per model variant. Table 16 reports the mean across the nine runs, the standard deviation across runs, and a bootstrap interval computed over generated pairs. The bootstrap interval summarizes sample variation within the pooled generated set for the sampled models; run-level standard deviations summarize variability across the sampled training and generation seeds. We therefore use the run-level standard deviation and per-run table to interpret training/generation-seed variability. Table 17 reports every seed combination for all-condition generation. Because these two tables summarize the all-condition generated mixture, they should not be used as substitutes for conditional target-bin hit rates.

Table 16: Training-seed and generation-seed uncertainty for all-condition generation. Each model variant is evaluated across three training seeds and three generation seeds using the shared activity-bin edges -5.3, -2.75, and -0.35. High fraction aggregates the high and very-high bins. Run SD is computed across the nine seed combinations; bootstrap intervals are computed over the pooled 9,000 generated pairs and summarize pooled-sample variation. This table characterizes the all-condition mixture and does not directly test low-, medium-, and high-condition controllability.

Model	Runs	Pairs	Mean activity	Activity CI	High frac.	High-frac. CI
Full PEAC	9	9000	-4.241 $\pm$ 0.234	[-4.275, -4.208]	0.178 $\pm$ 0.012	[0.170, 0.186]
No activity-bin cond.	9	9000	-4.667 $\pm$ 0.562	[-4.701, -4.632]	0.170 $\pm$ 0.104	[0.163, 0.178]
No initialization	9	9000	-4.208 $\pm$ 0.207	[-4.241, -4.176]	0.174 $\pm$ 0.055	[0.167, 0.182]
No latent MMD	9	9000	-4.265 $\pm$ 0.219	[-4.298, -4.231]	0.186 $\pm$ 0.021	[0.178, 0.194]
No pair fusion	9	9000	-4.239 $\pm$ 0.315	[-4.273, -4.206]	0.194 $\pm$ 0.044	[0.186, 0.202]

Across the nine runs, the high-fraction means differ from the single-run ordering in Table 3: No latent MMD and No pair fusion have higher mean high fractions than Full PEAC in this all-condition setting. The multi-seed experiment primarily serves as a variability, realism, and component-tradeoff check. Full PEAC has the smallest run-to-run standard deviation in all-condition high fraction (0.012), but Table 17 also shows that the all-condition low fraction changes substantially across training seeds, from about 0.32 for training seed 1 to about 0.15 for training seed 3. Thus high-fraction stability alone does not imply stability of the full activity distribution. Table 18 reports the direct conditional hit-rate evidence. Full PEAC gives strong medium-condition control and a higher high-condition hit rate than the no-initialization, no-pair-fusion, and no-activity-bin-conditioning variants, but it is not best on every conditional metric: No pair fusion has the highest low-condition hit rate, and No latent MMD has the highest high-condition hit rate. We therefore treat the conditional ablations as evidence for module-dependent tradeoffs rather than as proof that every module monotonically improves controllability. The realism metrics reveal complementary tradeoffs: without activity-bin conditioning, enhancer FBD increases; without initialization, promoter and enhancer co-occurrence distances increase; without pair fusion, promoter FBD and promoter co-occurrence worsen; and without latent MMD, promoter co-occurrence distance increases. These results support a realism-control tradeoff interpretation rather than a claim that each module monotonically raises high-activity output.

Table 17: Per-run training-seed and generation-seed results. Each row reports 1,000 all-condition generated promoter-enhancer pairs scored by the same CAGE-expression oracle. These all-condition fractions should not be interpreted as target-condition hit rates.

Model	Train seed	Gen. seed	Mean activity	High frac.	Low frac.	Medium frac.
Full PEAC	1	42	-4.445	0.167	0.317	0.516
	1	43	-4.469	0.158	0.331	0.511
	1	44	-4.333	0.185	0.309	0.506
	2	42	-4.389	0.168	0.259	0.573
	2	43	-4.433	0.176	0.286	0.538
	2	44	-4.283	0.195	0.265	0.540
	3	42	-3.948	0.187	0.148	0.665
	3	43	-3.976	0.175	0.152	0.673
	3	44	-3.897	0.192	0.147	0.661
No activity-bin cond.	1	42	-5.414	0.034	0.594	0.372
	1	43	-5.408	0.034	0.611	0.355
	1	44	-5.394	0.030	0.583	0.387
	2	42	-4.203	0.240	0.300	0.460
	2	43	-4.218	0.242	0.331	0.427
	2	44	-4.160	0.255	0.300	0.445
	3	42	-4.416	0.236	0.447	0.317
	3	43	-4.386	0.244	0.448	0.308
	3	44	-4.400	0.219	0.449	0.332
No initialization	1	42	-4.076	0.231	0.286	0.483
	1	43	-4.121	0.234	0.282	0.484
	1	44	-3.997	0.257	0.274	0.469
	2	42	-4.446	0.119	0.220	0.661
	2	43	-4.515	0.115	0.234	0.651
	2	44	-4.476	0.112	0.228	0.660
	3	42	-4.085	0.160	0.194	0.646
	3	43	-4.119	0.170	0.198	0.632
	3	44	-4.041	0.172	0.202	0.626
No latent MMD	1	42	-4.354	0.204	0.336	0.460
	1	43	-4.394	0.186	0.355	0.459
	1	44	-4.167	0.222	0.287	0.491
	2	42	-4.516	0.172	0.341	0.487
	2	43	-4.535	0.165	0.339	0.496
	2	44	-4.381	0.194	0.312	0.494
	3	42	-4.043	0.156	0.143	0.701
	3	43	-4.051	0.173	0.124	0.703
	3	44	-3.940	0.202	0.121	0.677
No pair fusion	1	42	-4.411	0.184	0.318	0.498
	1	43	-4.450	0.168	0.323	0.509
	1	44	-4.327	0.202	0.293	0.505
	2	42	-4.523	0.137	0.315	0.548
	2	43	-4.506	0.147	0.301	0.552
	2	44	-4.455	0.168	0.317	0.515
	3	42	-3.883	0.236	0.210	0.554
	3	43	-3.832	0.246	0.196	0.558
	3	44	-3.769	0.260	0.194	0.546

Table 18: Conditional target-bin hit rates across training and generation seeds. Each entry is the mean ± run-level standard deviation across nine training/generation-seed combinations. For low and medium conditions, the hit rate is the fraction assigned to the matching low or medium oracle bin. For the high condition, the hit rate uses the high bin, matching the main-text single-run reporting convention.

Model	Low hit	Medium hit	High hit
Full PEAC	0.571 ± 0.167	0.850 ± 0.050	0.640 ± 0.059
No activity-bin cond.	0.471 ± 0.131	0.360 ± 0.056	0.168 ± 0.110
No initialization	0.578 ± 0.206	0.845 ± 0.062	0.533 ± 0.239
No latent MMD	0.539 ± 0.217	0.763 ± 0.157	0.670 ± 0.015
No pair fusion	0.702 ± 0.115	0.781 ± 0.050	0.578 ± 0.063

A.10 SUPERVISION-SOURCE STRESS TEST

Other than ablation tests, we also test whether the method depends on a single dataset or teacher choice. We compare the full STARR/CAGE two-teacher model with CAGE-only, STARR-only, no-distillation, and independently retrained two-teacher variants. Each variant generates 1,000 all-condition promoter-enhancer pairs and is evaluated with the same oracle and realism metrics.

Table 19: Supervision-source stress test. The variants are generated from the same source-ablation suite and change which regulatory data sources or single-element teachers are available while using the same evaluation protocol. CAGE-derived oracle scores are reported as mean and 90th percentile; High is the fraction assigned to the high predicted-activity bin. Lower FBD and higher motif enrichment correlation are better.

Model	Mean	P90	High	P-FBD	E-FBD	P-r	E-r
CAGE-only, no STARR initialization	-4.60	-3.25	0.091	0.3878	0.2924	0.491	0.589
Enhancer-teacher-only STARR/CAGE	-4.69	-2.88	0.061	0.0072	0.0231	0.711	0.394
Full STARR/CAGE two-teacher (suite)	-4.57	-1.83	0.152	0.0157	0.0016	0.723	0.804
No-distill STARR/CAGE	-4.66	-1.84	0.137	0.0074	0.0019	0.771	0.823
Promoter-teacher-only STARR/CAGE	-4.70	-2.28	0.103	0.0030	0.0051	0.903	0.759
Retrained STARR/CAGE two-teacher	-4.54	-2.06	0.123	0.0149	0.0018	0.766	0.818
STARR-only, no CAGE expression conditioning	-5.00	-2.15	0.130	0.0236	0.0030	0.615	0.749

The supervision-source stress test shows tradeoffs among CAGE conditioning, STARR initialization, and teacher priors (Table 19). CAGE-only training has weaker realism and a smaller high-activity tail, while STARR-only training gives the lowest mean oracle score. The no-distillation variant improves promoter FBD and promoter/enhancer motif correlations relative to the full suite row but lowers the high-activity fraction, indicating a realism/activity tradeoff instead of a uniform degradation from removing distillation. The independently retrained two-teacher model remains close to the full model on several metrics, supporting reproducibility of the broad behavior rather than dominance on every metric. Together, these results argue against a single dataset-specific shortcut and motivate broader cross-assay testing.