Understanding Conditional Computation in Contrastive Phenomic Retrieval

Karush Suri[1], **Puria Azadi Moghadam**[1,2], **Frederik Wenkel**[1],
Maciej Sypetkowski[1], **Emmanuel Bengio**[1], **Emmanuel Noutahi**[1], **Dominique Beaini**[1,3]
[1] Valence Labs Montreal, [2] University of British Columbia, [3] Université de Montréal, Mila
karush@valencelabs.com

Abstract

Contrastive Phenomic Retrieval is the problem of learning a joint embedding space between molecules and cell phenomes using multimodal contrastive learning. Conditional computation strategies such as Mixtures of Experts (MoEs) have been widely adopted for contrastive learning. However, their efficacy and versatility arising from subnetwork computations remain an unanswered question. *How Do MoEs benefit contrastive phenomic retrieval?* We identify a key phenomenon in CLIP-based objectives wherein models overfit to the underlying cosine similarity metric when learning representations of molecules and cell phenomes. Embedding logits of multimodal encoders collapse in a small range and remain similar to each other, hence hindering retrieval performance. We empirically validate that MoEs prevent this overfitting by providing a higher level of diversity in encoder representations. Expert networks act as diversity-inducing bottlenecks which prevent encoder embeddings from becoming too similar. Our theoretical analysis shows that expert widths scale quadratically with the ratio of layernorm parameters. Our experiments demonstrate that embeddings learned by MoEs are diverse and continue to adapt, thereby resulting in $1.23\times$ improvement over previous CLIP methods and 7.07% improvement over previously performant contrastive phenomic models. We further study the effectiveness of MoEs in few-shot downstream tasks of concentration prediction and molecular activity recognition, as well as zero-shot tasks of activity cliff prediction and gene knockout identification. Finally, for the first time, we show that MoEs enable amortized inference of large multimodal phenomic models by learning conditional GFlowNet samplers.

Contents

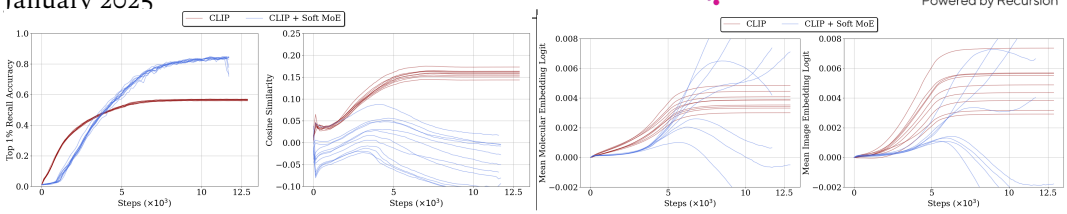

Figure 1: (**left**) Variation of Top-1% recall accuracy and cosine similarity. CLIP overfits to the underlying cosine similarity metric which hurts molecular retrieval performance. (**right**) Variation of encoder embedding logits during training. Similarity overfitting occurs as a direct consequence of embedding logits collapsing in a small value range and remaining similar to each other. MoEs prevent this logit collapse by inducing a higher level of diversity in encoder embeddings. Results are presented over 9 random runs.

1 Introduction

Learning effects of molecules on cellular compositions is a pivotal aspect of discovering novel therapeutic candidates (Akarapipad et al., 2021; Bohacek et al., 1996). Cellular compositions encode phenotypical effects which hold the key to reversing the effects of a particular disease (Vincent et al., 2022). These phenotypical effects are learned by leveraging contrastive learning over multiple molecular modalities (such as molecular structures, fingerprints, microsocopy lab experiment images, etc.) (Simm et al., 2018; Bray et al., 2016; Hofmarcher et al., 2019). Once learned, similar molecules may then be retrieved from the joint embedding space of molecules and cell phenomes either using similarity search or amortized inference (Fradkin et al., 2024). Identifying and retrieving such molecules from the joint embedding space of multimodal contrastive learning models is the problem of *contrastive phenomic retrieval* (Sanchez-Fernandez et al., 2023; Nguyen et al., 2023).

Akin to multimodal contrastive learning, recent advances in contrastive phenomic retrieval leverage conditional computation strategies to stabilize and improve retrieval performance. One such strategy is the use of Mixtures of Experts (MoEs) (Jacobs et al., 1991). MoEs serve as parameter-efficient bottlenecks wherein embeddings are distributed across different networks (or experts) using a router network. Each expert learns representations corresponding to a particular aspect of the embedding, hence expertizing across a set of features. Expert embeddings are then aggregated to yield network outputs. In the case of language and vision, MoEs aid in constructing distributed representations such that each token has access to multiple experts. This results in improved scalability over increasing number of data samples (Jiang et al., 2024).

While MoEs have demonstrated significant improvements in both performance and parameter efficiency, a concrete understanding of their utility and versatility remain an unanswered question. Models of today adopt MoEs as black-box architectures with little intuition about their operation. Theoretically, an understanding on the scalability of each expert remains unexplored. Empirically, it is unclear as to how MoEs benefit contrastive learning and what is the central factor aiding them to construct rich representations across a wide range of modalities.

We aim to answer the above questions by studying the utility of MoEs in contrastive phenomic retrieval. We begin by identifying a key phenomenon in CLIP-based objectives wherein models learned with CLIP overfit to the underlying cosine similarity metric. Embedding logits learned by CLIP encoders collapse in a small value range and remain similar to each other. We further show that this collapse of logit values hinders retrieval performance. Such a phenomenon leads us to explore the addition of MoEs in our encoders. Consequentially, MoEs prevent the collapse of embedding logits and serve as diversity-inducing bottlenecks which yield higher cosine similarities between both encoder embeddings. Additionally, we show that prevention of similarity overfitting by MoEs results in an absolute $1.23\times$ improvement over previous CLIP methods and 7.07% improvement over previously performant contrastive phenomic models. Our theoretical analysis shows that expert widths scale quadratically with the ratio of parameters of layernorm operator. Empirically, we evaluate the utility of diversity induced by MoEs on few-shot downstream tasks of concentration prediction and molecular activity recognition, as well as zero-shot tasks of activity cliff prediction and gene knockout identification. Finally, as a first, we show that MoEs enable amortized inference of large multimodal phenomic models by learning conditional GFlowNet samplers.

2 PRELIMINARIES

Contrastive Phenomic Retrieval: We study the problem of learning multimodal representations of molecules and phenomic experiments of treated cells. Our setting considers a set of lab experiments $\mathcal{E}$ defined as the tuple $(\mathbf{X}, \mathbf{M}, \mathbf{C}, \mathbf{\Psi})$ wherein each experiment $e \in \mathcal{E}$ consists of cell images $\mathbf{x} \in \mathbf{X}$ and molecules as perturbations $\mathbf{m} \in \mathbf{M}$ obtained at specific dosage concentrations $\mathbf{c} \in \mathbf{C}$ with $\psi \in \mathbf{\Psi}$ denoting the threshold of molecular activity. Contrastive learning with CLIP utilizes an image encoder $f_{\theta_\mathbf{x}}$ with parameters $\theta_\mathbf{x}$ to generate the image embedding $\mathbf{z}_\mathbf{x} = f_{\theta_\mathbf{x}}(\mathbf{x})$, and a molecular encoder $f_{\theta_\mathbf{m}}$ with parameters $\theta_\mathbf{m}$ to generate the molecular embedding $\mathbf{z}_\mathbf{m} = f_{\theta_\mathbf{m}}(\mathbf{m}, \mathbf{c})$. Prior methods in multimodal contrasive learning utilize the InfoNCE loss to maximize the joint likelihood of $\mathbf{x}$ and $\mathbf{m}$. Given a set of $N \times N$ random samples $(\mathbf{x}_1, \mathbf{m}_1, \mathbf{c}_1), (\mathbf{x}_2, \mathbf{m}_2, \mathbf{c}_2), \ldots (\mathbf{x}_{N^2}, \mathbf{m}_{N^2}, \mathbf{c}_{N^2})$ containing N positive samples at k^{th} index and $(N-1) \times N$ negative samples, optimizing Equation 1 maximizes the likelihood of positive pairs while minimizing the likelihood of negative pairs.

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\log \frac{\exp(\langle \mathbf{z}_{\mathbf{x}_i}, \mathbf{z}_{\mathbf{m}_i} \rangle / \tau)}{\sum_{k=1}^{N} \exp(\langle \mathbf{z}_{\mathbf{x}_i}, \mathbf{z}_{\mathbf{m}_k} \rangle / \tau)} + \log \frac{\exp(\langle \mathbf{z}_{\mathbf{x}_i}, \mathbf{z}_{\mathbf{m}_i} \rangle / \tau)}{\sum_{k=1}^{N} \exp(\langle \mathbf{z}_{\mathbf{m}_i}, \mathbf{z}_{\mathbf{x}_k} \rangle / \tau)}\right]. \quad (1)$$

Here, τ denotes the softmax temperature and $\langle \cdot \rangle$ denotes the cosine similarity.

Conditional Computation with MoEs: MoE architectures distribute embeddings across expert networks. Our study focusses on Soft MoEs (Puigcerver et al., 2023) which utilize a series of *dispatch* and *aggregate* operations. We denote the input embedding $\mathbf{Z} \in \mathbb{R}^{b \times d}$, where b is the number of tokens and d is their dimension. Each MoE layer uses a set of s expert networks $(g_1, g_2, \ldots, g_s)$ applied on individual tokens. Each expert processes p slots, each consisting of a d-dimensional set of parameters, $\mathbf{\Phi} \in \mathbb{R}^{d \times (s.p)}$. The input slots $\tilde{\mathbf{Z}} \in \mathbb{R}^{(s.p) \times d}$ arise from a weighted sum of all b tokens as presented in Equation 2.

$$\mathbf{D}_{ij} = \frac{\exp((\mathbf{Z}\mathbf{\Phi})_{ij})}{\sum_{i'=1}^{b} \exp((\mathbf{Z}\mathbf{\Phi})_{i'j})} \quad ; \quad \tilde{\mathbf{Z}} = \mathbf{D}^{\mathrm{T}}\mathbf{Z} \quad (2)$$

Here, $\mathbf{D}$ are the dispatch weights. Upon application of expert functions on each slot, we obtain the output slots $\tilde{\mathbf{Y}}_i = g_i(\tilde{\mathbf{Z}}_i)$. Finally, a weighted sum across all $(s.p)$ slots results in the output tokens $\mathbf{Y}$ with $\mathbf{A}$ denoting the aggregate weights in Equation 3.

$$\mathbf{A}_{ij} = \frac{\exp((\mathbf{Z}\mathbf{\Phi})_{ij})}{\sum_{j'=1}^{s.p} \exp((\mathbf{Z}\mathbf{\Phi})_{ij'})} \quad ; \quad \mathbf{Y} = \mathbf{A}\tilde{\mathbf{Y}} \quad (3)$$

3 RELATED WORK

Multimodal Contrastive Learning: Prior multimodal models build on CLIP (Radford et al., 2021) and combine samples from two or more domains to learn expressive representations (Alayrac et al., 2022; Huang et al., 2023). Contrastive learning methods maximize similarity between paired samples traditionally in language-vision domains (Oord et al., 2018a). Recent works in multimodal contrastive learning aim at reducing compute and parameter budgets for data-efficient training (Henaff, 2020). (Zhai et al., 2022) utilize unimodal pretrained models for one or both modalities improving zero-shot performance with an order of magnitude fewer paired samples. (Zhai et al., 2023) further demonstrate that utilization of an elementwise sigmoid loss allows contrastive learners to scale under label noise. By using a unimodal pretrained model to learn similarities, (Srinivasa et al., 2023) demonstrate improved performance on zero-shot retrieval. (Yang et al., 2023) extend the multimodal framework to molecular multi-omics for cancer subtyping. Our treatment of multimodal contrastive learning models is parallel to the aforesaid directions.

Pretraining Molecule-Phenome Representations: Self-supervised methods have demonstrated pronounced success in vision, language and molecule representations (Balestriero et al., 2023; Radford et al., 2019; Zaidi et al., 2022). In vision, these methods are used to minimize distance in the latent space of models between two views of the same sample (Chen et al., 2020; Sohn, 2016; Grill et al., 2020; He et al., 2020). On the other hand, reconstruction-based objectives have also permeated representation learning such as masked autoencoders

(MAE) (He et al., 2022). MAEs allow for a higher degree of versatility by simulatenously masking and reconstructing segments of modality inputs (Seo et al., 2023). These architectures typically utilize transformer architectures to partition the image into learnable tokens and reconstruct masked patches (Feichtenhofer et al., 2022; Cao et al., 2022; Dosovitskiy et al., 2020). These methods have been extended to microscopy experimental data designed to capture cell morphology (Xie et al., 2023). (Kraus et al., 2024) utilize a masked autoencoder with a ViT-L/8+ architecture and a custom Fourier domain reconstruction loss, yielding informative representations of phenomic experiments (Dehghani et al., 2023). Our work leverages analogous pretraining schemes.

Conditional Computation with MoEs: Recent advances in large model pretraining and contrastive learning employ MoEs (Jacobs et al., 1991). These architectures serve as parameter-efficient alternatives which construct sets of distributed representations (Cai et al., 2024). MoEs, being sparse gated units, only activate a portion of their parameters during training (Shazeer et al., 2017). Recently, (Puigcerver et al., 2023) make an effort to relax the sparsity constraints by proposing a soft MoE layer which adopts a weighted aggregation operation (eg- softmax) to yield output tokens. While contemporary models utilized MoEs for gaussian process regression and inference (Ng & Deisenroth, 2014), recent methods have adopted these architectures for pretraining large foundational priors such as vision and language models (Lo et al., 2024). (Zhu et al., 2024) bootstrap continual MoE training from pretrained language embeddings to accelerate downstream language generalization. (Bao et al., 2021), on the other hand, incorporate MoEs in the multimodal setting by embedding vision and language via MoE encoders. (Li et al., 2024) extend this idea further and embed audio, speech, vision and text in a joint latent space utilizing parallelized experts in MoEs. Finally, (Yu et al., 2023) utilize MoEs to learn interactions across different input modalities under constraints of partial labels. While the above works demonstrate scalable and performant use of MoEs, they fail to shed light on why these architectures benefit training. Current methods utilize experts as black-box architectures and provide little insights towards their operation. Among recent works, (Chen et al., 2022) attempt to explain MoEs by assessing their theoretical properties and non-linear expressivity of experts. MoE router learns cluster-center features which the experts can leverage to divide the problem into smaller sub-problems. Our empirical analysis in contrastive learning follows this line of work.

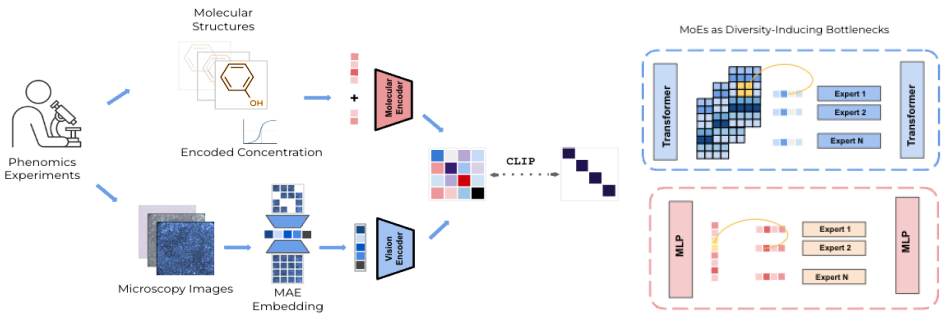

Figure 2: (**left**) Illustration of contrastive phenomic retrieval wherein we use microsocopy images, molecule structure fingerprints and one-hot encoded concentrations as input modalities. CLIP learns a joint embedding space of phenome lab experiments using a vision encoder and a molecular encoder. (**right**) Implementation of Soft MoEs in different encoder architectures. We combine MoEs with MLPs and Transformers where molecule features act as tokens.

4 The Many Shades of CLIP

We begin by studying the behavior of CLIP during contrastive pretraining of phenome lab experiments. Figure 2 (left) presents our setting wherein we consider three input modalities, (1) microscopy images, (2) molecule structure fingerprints and (3) concentrations as one-hot encoded context. Our empirical analysis utilizes biological phenome maps consisting of raw cellular microscopy images (Fay et al., 2023). Since raw microscopy images have an order of magnitude more dimensions than molecule fingerprints, we additionally pretrain a unimodal MAE to extract compact latent representations of visual inputs. Both encoders

are trained simultaneously using the CLIP objective (Radford et al., 2021). Additional details of our experimental setup can be found in Appendices C and D.

CLIP overfits to the underlying similarity metric: Figure 1 (left) presents our main result in understanding CLIP for molecular retrieval. We observe that during training, cosine similarities between the vision encoder embedding and molecular encoder embedding converge in a fixed value range. While this convergence in itself is not problematic as similarity values do not overshoot or explode. On the other hand, cosine similarity values collapse and fail to decrease over the course of pretraining. Since both encoders shape the embedding space to separate positive phenotypic traits from the negative ones, one would expect a consistent decrease in similarity. This is due to the fact that encoder embeddings represent different molecular features which correspond to different cluster groups in the embedding space. Thus, a performant encoder must continue to cluster molecules.

Similar representations hinder performance: As we observed, both the vision and molecular encoders overfit to the similarity metric. *But why are similar embeddings detrimental?* We empirically observe that overly similar embeddings prevent CLIP from further improving on molecular retrieval. In Figure 1 (left), CLIP demonstrates significantly lower retrieval rates. When compared to Soft MoEs (which we explain in the next section), the naive CLIP model is $1.34\times$ behind in Top-1% retrieval accuracy while having $7.5\times$ higher similarity values. Our detailed results (in Appendix F) further show that compared to Sparse MoEs, naive CLIP has an order of magnitude higher cosine similarities while trailing by $1.21\times$ in recall performance.

Logits collapse and remain similar: Given the above phenomenon and the reduced performance of CLIP, one may ask *what is the main cause of similarity overfitting?* We empirically answer this question by monitoring expected values of encoder embedding logits. Figure 1 (right) presents this variation wherein we observe that logits of CLIP saturate early on during training. Furthermore, this saturation is consistently reported to collapse in the same value range for both encoders. Upon comparing to Soft MoEs, this variation is in stark contrast. The addition of MoEs to CLIP leads to a wider spread of logit values. Embedding logits continue to grow (or shrink) during training which prevents them from collapsing in a single direction. Additionally, as we noted before in Figure 1 (left), this addition of MoEs results in performant CLIP models for molecular retrieval.

Method	Without Soft MoEs					With Soft MoEs				
	Recall@1 (↑)	Recall@5 (↑)	Recall@10 (↑)	Cosine Similarity (↓)	Test Error (↓)	Recall@1 (↑)	Recall@5 (↑)	Recall@10 (↑)	Cosine Similarity (↓)	Test Error (↓)
CLIP (Radford et al., 2021)	0.73±0.0003	0.80±0.0007	0.89±0.0004	0.15±0.00	5.62±0.001	**0.90±0.004**	**0.94±0.003**	**0.97±0.001**	**-0.02±0.03**	**4.73±0.007**
Hopfield-CLIP (Ramsauer et al., 2020)	0.180±0.01	0.18±0.01	0.33±0.016	0.07±0.003	8.48±0.02	**0.24±0.013**	**0.25±0.015**	**0.43±0.02**	**0.014±0.009**	**8.45±0.02**
InfoLOOB (Poole et al., 2019)	0.492±0.009	**0.58±0.01**	0.740±0.009	0.56±0.03	7.46±0.06	**0.499±0.016**	0.57±0.02	**0.743±0.018**	**0.45±0.04**	**7.31±0.09**
NtXent (Sohn, 2016)	0.27±0.01	0.30±0.02	0.46±0.02	0.51±0.02	8.76±0.15	**0.41±0.01**	**0.46±0.01**	**0.64±0.01**	**0.40±0.02**	**8.49±0.10**

Table 1: Molecular retrieval performance of different contrastive learning objectives. Similarity overfitting affects objectives which utilize CLIP as their loss function. These methods require the use of diversity-inducing strategies such as MoEs. Bold entries denote highest values across corresponding columns while shaded entries denote highest values across the table.

Does similarity overfitting affect other CLIP-based objectives? Given that we observe similarity overfitting in CLIP, we now assess whether this phenomenon plagues other methods that build on top of CLIP. We compare retrieval performance of recent CLIP-based objectives. In Table 1 we observe that CLIP-based objectives, in the absence of MoEs, have high cosine similarities. Specifically, InfoLOOB (Poole et al., 2019) and NtXent (Sohn, 2016) have approximately $3\times$ higher similarities compared to CLIP. Furthermore, all the presented objectives explicitly require the use of MoEs to reduce similarity overfitting and boost performance. Thus, high cosine similarities of CLIP-based methods combined with their reliance on MoE architectures indicate that similarity overfitting plagues recent methods as well that aim to adopt or improve over CLIP.

5 MoEs as Diversity-Inducing Bottlenecks

MoEs prevent similarity overfitting and benefit molecular retrieval: In the previous section we showed that CLIP-based models are plagued by similarity overfitting. How can we induce diversity in the embeddings of CLIP encoders? We now show that MoEs serve as a suitable alternative for constructing diverse molecular representations. Figure 1 (left)

demonstrates that the addition of distributed experts via the soft MoE layer in encoder blocks prevents the embeddings from collapsing to similar values. Additionally, prevention of this collapse further benefits molecular retrieval performance. Compared to CLIP, CLIP with soft MoE encoders has $7.5\times$ lower cosine similarity values resulting in an absolute $1.23\times$ increase in Top-1% recall accuracy. Each expert network in the mixture abstracts molecule features in a different subspace, hence resulting in diverse embedding logits. We validate this functionality of MoEs empirically in Figure 1 (right). Compared to CLIP, soft MoE encoder embeddings present a wider spread of logits during the course of training. Contrary to CLIP logits, MoE feature logits continue to grow (or shrink) as new molecules are processed by experts. This trend is consistent for both image and molecular encoders, indicating that MoEs mitigate the collapse and induce diversity irrespective of the input modality.

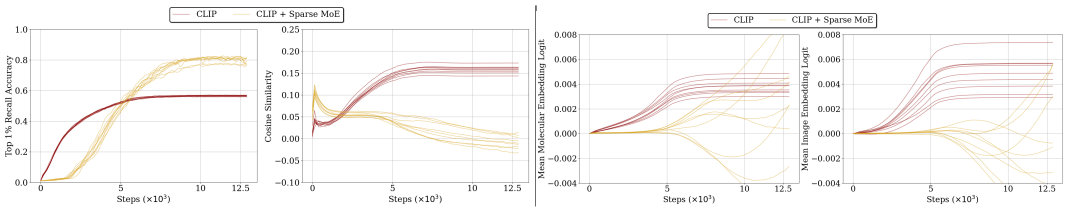

Figure 3: (**left**) Variation of Top-1% recall accuracy and cosine similarity for CLIP with Sparse MoEs. In addition to soft MoEs, traditional sparse MoEs are also found to mitigate similarity overfitting. (**right**) Variation of encoder embedding logits during training. Compared to CLIP, the addition of sparse MoEs have a wider spread of embedding logits. Results are presented over 9 random runs.

Which MoEs induce diversity? While we noted the efficacy of soft MoEs in inducing diversity, a natural question to ask is *do other MoEs also demonstrate similar properties?* Figure 3 (left) validates our hypothesis wherein we observe that sparse MoEs (Shazeer et al., 2017) also present a similar trend. Compared to CLIP, sparse MoE encoders provide an absolute $1.21\times$ improvement in retrieval performance boosting recall performane from 73% to 88% across 9 random runs. Furthermore, this improvement arises as a direct consequence of lower similarity between encoder embeddings wherein sparse MoEs have cosine similarities values of 0.003, an order of magnitude lower than 0.15 similarity values of CLIP. As seen for soft MoEs, Figure 3 (right) shows that both encoders learn embedding logits which continue to adapt in comparison to the collapsed embeddings of naive CLIP.

Theoretical Analysis (**MoEs enable scaling of expert widths**)**:** We now study the theoretical properties of MoE layers with the intent of understanding their ability to induce diversity. Our analysis aims to uncover as to what kind of experts are required for different molecular input features to the MoE layer. Considering the soft MoE layer block consisting of linearly parameterized experts with layer normalization and the softmax activation, we obtain the below result.

Theorem 1. *Given a linear expert g_i with parameters $\mathbf{\Phi}_i$, input features $\mathbf{Z}$, layernorm operator LN(·) with parameters α and β and softmax(·) operator , width of expert network g_i, denoted by d, scales inversely with the squared ratio of layernorm parameters, $d \propto \left(\frac{\beta}{\alpha}\right)^2$.*

Intuitively, Theorem 1 indicates a bias-variance trade-off between layernorm parameters of the expert which is captured in its width. Wider expert networks present a high bias across their layer-normalized features whereas smaller experts have a high variance across layer-normalized features. Furthermore, the variation of width is exponential for parameter ratios. A small increment in expert bias may indicate the requirement of a quadratically wider expert. Similarly, quadratically smaller experts may be suitable for input features having a low variance.

6 Diagnosing Contrastive Phenomic Retrieval with MoEs

In this section, we demonstrate the efficacy of MoEs as diversity-inducing bottlenecks for contrastive phenomic retrieval. We first assess their ability to accelerate pretraining, followed by positive transfer on few-shot tasks of concentration prediction and activity recognition, zero-shot task of activity cliff prediction, and amortizing inference with GFlowNets. Additional results for zero-shot gene knockout identification and pretraining can be found in Appendices E and F respectively.

6.1 Pretraining with Representation Diversity

Method	Recall@1 (↑)	Recall@5 (↑)	Recall@10 (↑)	Cosine Similarity (↓)	Test Error (↓)
CLIP	0.73±0.0003	0.80±0.0007	0.89±0.0004	0.15±0.00	5.62±0.001
CLIP+Dropout	0.67±0.00	0.73±0.0003	0.84±0.00	0.11±0.00	5.94±0.00
CLIP+Weight Decay	0.73±0.00	0.80±0.00	0.90±0.00	0.14±0.00	5.60±0.001
CLIP+Resets	0.38±0.11	0.41±0.12	0.56±0.13	0.03±0.003	7.76±0.58
CLIP+Pruning	0.069±0.007	0.006±0.006	0.12±0.011	0.07±0.06	25.14±11.46
CLIP+Soft Pruning	0.61±0.08	0.67±0.09	0.80±0.08	0.13±0.0008	6.56±0.87
Sparse CLIP	0.70±0.00	0.77±0.00	0.87±0.00	0.14±0.00	5.76±0.001
CLIP+Sparse MoE	0.88±0.006	0.93±0.004	0.96±0.003	0.003±0.003	4.95±0.08
CLIP+Soft MoE	**0.90±0.004**	**0.94±0.003**	**0.97±0.001**	-0.02±0.03	**4.73±0.007**

Slots	Recall@1 (↑)	Recall@5 (↑)	Recall@10 (↑)	Cosine Similarity (↓)	Test Error (↓)
2	0.83±0.10	0.87±0.11	0.90±0.11	-0.03±0.02	4.80±0.66
4	0.90±0.02	0.94±0.003	0.97±0.001	-0.01±0.001	4.62±0.07
8	0.72±0.12	0.76±0.13	0.80±0.13	-0.03±**0.019**	4.80±0.35
16	0.89±0.23	0.92±0.25	0.93±0.26	0.004±0.04	5.84±1.41
32	**0.906±0.004**	**0.946±0.003**	**0.97±0.001**	-0.03±0.001	**4.52±0.007**

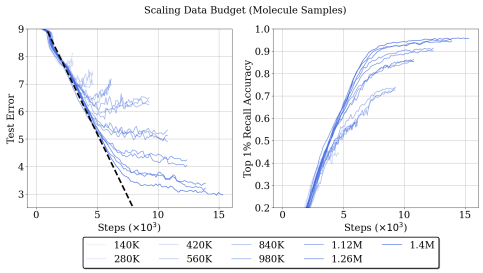

Figure 4: (**top-left**) Comparison of molecular retrieval performance for CLIP-based models. Across different design choices, MoEs consistently demonstrate improved recall performance. (**bottom-left**) Retrieval performance for varying number of slots per expert. More slots provide expressivity to experts. (**right**) Variation of test error and retrieval performance with data budget. A larger and diverse pretraining data distribution boosts retrieval performance.

Our pretraining analysis aims to answer three main questions; (1) Does the addition of representation diversity benefit performance?, (2) Does the addition of MoEs obey a scaling law fit?, and (3) Which components of MoEs benefit diversity and performance? Figure 4 (top-left) answers our first question by comparing CLIP variants with the addition of Soft MoEs. In our comparison, we include common design decisions which are required to stabilize contrastive learning (such as dropout, weight decay and sparsity). Additionally, we include methods which benefit unsupervised learning models in general (such as periodic resetting of parameters and pruning network weights). Across the board, we note that CLIP with the addition of soft MoEs outperforms prior design decisions leading to $2-17\%$ improvements over baselines. Furthermore, MoE-based methods induce the most diversity as indicated by lowest cosine similarity values between the encoder embeddings.

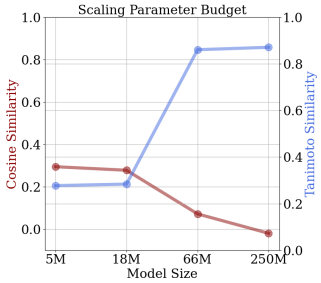

Figure 5: Variation of Tanimoto and cosine similarities with parameter budget. Larger models prevent cosine similarity overfitting, hence presenting higher chemical similarity between encoder embeddings.

We now answer our second question by assessing the scalability of models trained using soft MoEs. Figure 4 (right) presents scaling law for data budget when varying the number of molecule-phenomic samples in our RXRX3 pretraining data distribution. We note that increasing number of data samples leads to an asymptotic increase in recall performance. Model performance scales along the line $y = -0.00091 \times x + 9.5$ validating the power law fit. Figure 5 studies the scaling law for model parameter budget. Larger models (upto the order of 250M parameters) present a consistent increment in diversity by decreasing cosine similarity values. This increase in dviersity leads to chemically similar embeddings as validated by growing tanimoto similarity values. Intuitively, additio of MoEs at larger parameter budgets leads to embeddings that encode chemically-rich information by mitigating similarity overfitting.

While MoEs demonstrate improvements in performance and scalability, it is unclear as to which components of these architectures contribute towards their benefits. We answer our third question by ablating between different MoE architectures. We begin by varying the number of input slots for each expert while keeping the number of experts fixed. Figure 4 (bottom-left) presents the variation of retrieval performance for varying number of input slots. While MoEs with smaller slots per experts induce diversity, these fall short of boosting molecular retrieval performance. This arises as a lack of expressivity within the model architecture. MoEs with larger number of slots better induce diversity and boost performance with 32 slots per expert providing a 7.6% improvement over the smallest architecture. Figure 6 (top-left) extends our analysis to varying number of experts while keeping the slots per expert fixed. More experts per encoder benefit diversity the most. Empirically, both encoders consisting of 8-16 experts are found to be stable with the most representation diversity arising from distributed representations of each expert. In Figure 6 (bottom-right) we answer the key question of *where should the Soft MoE layer be placed?* Traditional language architectures place MoEs in the penultimate layers of the model. We see that phenomic models benefit the most in performance and diversity when MoEs are placed within the encoder blocks. Placing MoEs in the input layer leads to instability wherein encoders fail to capture representations propagating from different expert heads. On the other hand, placing MoEs in the projection head leads to coarser embeddings wherein outputs are too similar. Encoder blocks serve as the right balance between stability and diversity by accumulating and distributing embeddings between layers.

Experts	Recall@1 (↑)	Recall@5 (↑)	Recall@10 (↑)	Cosine Similarity (↓)	Test Error (↓)
2	0.90±0.002	0.94±0.002	0.97±0.001	0.01±0.001	4.59±0.04
4	0.90±0.002	0.94±0.002	0.97±0.001	-0.008±0.001	4.57±0.05
8	**0.906±0.002**	**0.946±0.002**	**0.976±0.001**	-0.03±0.001	**4.56±0.08**
16	0.904±0.08	**0.946±0.002**	**0.976±0.001**	-0.01±0.00	**4.56±0.03**

Placement	Recall@1 (↑)	Recall@5 (↑)	Recall@10 (↑)	Cosine Similarity (↓)	Test Error (↓)
Input Layer	0.05±0.00	0.04±0.00	0.10±0.001	0.01±0.001	9.016±0.003
Projection Head	0.06±0.03	0.05±0.002	0.10±0.006	0.13±0.005	9.66±0.18
Encoder Block	**0.90±0.004**	**0.94±0.003**	**0.97±0.001**	-0.02±0.03	**4.73±0.007**

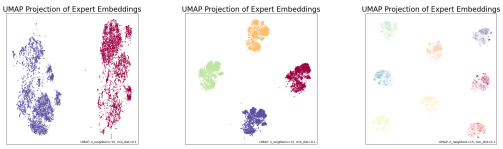

Figure 6: Variation of molecular retrieval performance with (**top-left**) varying number of experts and (**bottom-left**) different placements of MoE layer. Larger experts placed in encoder blocks are found to benefit retrieval performance. (**right**) UMAP projection of learned expert embeddings from the molecular encoder for 2, 4 and 8 experts respectively. Expert representations are well clustered indicating that each expert learns different aspects of molecule features (enlarged figures presented in F).

We qualitatively validate the utility of MoE experts in inducing representation diversity. Figure 6 (right) presents the UMAP projection of expert output embeddings for Soft MoE layers with 2, 4 and 8 experts respectively. Irrespective of the number of experts, the soft MoE layer assigns different clusters to different features of the same molecule. Each expert learns different yet specific aspects of molecular features which are captured to mitigate similarity overfitting.

We now compare the addition of Soft MoEs with current state-of-the-art contrastive phenomic models. In addition to the performant MolPhenix model (Fradkin et al., 2024), we compare with InfoLOOB (Poole et al., 2019), CLOOME (Sanchez-Fernandez et al., 2023) and SigLIP (Zhai et al., 2023). Our implementation adds the Soft MoE layer in the molecular encoder of MolPhenix. Table 2 presents top-1% and top-5% recall accuracies of contrative phenomic retrieval models. Compared to MolPhenix, the addition of MoEs leads to an absolute 7.07% improvement in top-1% accuracy and 4.21% improvement in top-5% accuracy, hence setting a new *state-of-the-art in contrastive phenomic retrieval.*

Method	Top-1% recall accuracy (↑)	Top-5% recall accuracy (↑)
InoLOOB (Poole et al., 2019)	0.3878±0.0018	0.5565±0.0012
CLOOME (Sanchez-Fernandez et al., 2023)	0.4176±0.0033	0.5870±0.0017
SigLIP (Zhai et al., 2023)	0.5343±0.0032	0.6819±0.0026
MolPhenix (our implementation) (Fradkin et al., 2024)	0.6745±0.0031	0.8662±0.0017
MolPhenix + Sot MoE	**0.7452**±0.0003	**0.9083**±0.0041

Table 2: Comparison with state-of-the-art phenomic retrieval models. Compared to MolPhenix, addition of Soft MoEs improves Top-1% recall accuracy by 7.07%.

6.2 Few-Shot Concentration Prediction

Predicting the concentration of molecular perturbations is a key ingredient in evaluating the chemical validity of CLIP embeddings. We consider the task of concentration prediction,

wherein we train a logistic binary classifier in few shots to classify whether a molecule belongs to a concentration set or not. Our classifiers are trained from frozen embeddings of both molecular and vision encoders. In order to prevent passing ground-truth labels, we mask out true concentrations as inputs to the molecular encoder. Since concentrations exhibit an inherent discrete sparse structure, an ideal embedding would be the one that captures correlations in as less as a few bits.

Table 3 presents our results for few-shot concentration prediction using embeddings of CLIP-like models. While MoE models present improved diversity and downstream classification performance, sparse MoEs are found to be more performant. Compared to CLIP and sparse MoEs, addition of soft MoEs presents the most diverse embeddings. Compared to CLIP and soft MoEs, addition of sparse MoEs presents diverse as well as performant embeddings.

Method	Concentration=1.0							Concentration=0.1						
	AUROC (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	RBF Similarity (↑)	Cosine Similarity (↓)	Training Error (↓)	AUROC (↑)	Precision (↑)	Recall (↑)	Accuracy (↑)	RBF Similarity (↑)	Cosine Similarity (↓)	Training Error (↓)
CLIP	0.62±0.19	0.59±0.16	0.58±0.15	0.58±0.15	0.2815	0.1579	0.71±0.18	0.97±0.04	0.96±0.09	0.96±0.09	0.95±0.08	0.2872	0.1552	0.19±0.16
CLIP+Soft MoEs	0.79±0.12	0.74±0.10	0.74±0.10	0.74±0.10	**0.6367**	**-0.0269**	0.58±0.20	0.98±0.25	0.97±0.03	0.97±0.03	0.97±0.03	**0.6366**	**-0.0265**	**0.13±0.13**
CLIP+Sparse MoEs	**0.81±0.11**	**0.76±0.09**	**0.76±0.09**	**0.76±0.09**	0.6094	-0.0243	**0.56±0.20**	**0.98±0.24**	0.97±0.03	0.97±0.03	0.97±0.03	0.6103	-0.0247	0.14±0.13

Table 3: Few-shot concentration prediction performance of downstream classifiers trained with embeddings of CLIP, Soft MoEs and Sparse MoEs.

6.3 Few-Shot Molecule Activity Recognition

Our second task focusses on predicting molecular activity with a handful of unseen molecules. We train a downstream linear regressor to predict p value cutoff thresholds ψ which denote the boundary of molecular activity. Since, novel molecules are unseen we do not have access to their phenomic images. Thus, we only utilize the embedding of frozen molecular encoder as input to the regressor.

Method	MSE (↓)	MAE (↓)	R^2 (↑)
CLIP	0.07±0.01	0.21±0.02	0.35±0.14
CLIP+Soft MoEs	**0.05±0.01**	**0.20±0.02**	**0.45±0.09**
CLIP+Sparse MoEs	0.06±0.01	0.21±0.02	0.36±0.13

Table 4: Few-shot activity recognition performance of downstream linear regressors trained with embeddings of CLIP, Soft MoEs and Sparse MoEs.

Table 4 presents our results for few-shot molecular activity prediction using CLIP models. While the CLIP embedding in itself captures activity sufficiently, addition of MoEs leads to further improvements in recognition performance. Notably, cutoff thresholds are best predicted using soft MoEs of molecular encoders. This arises as a direct consequence of cutoff thresholds possessing a more complex continuous structure which is best captured by weighted embeddings of soft MoEs.

6.4 Zero-Shot Activity Cliff Prediction

Detection of *activity cliffs*, defined as differences in biological activity between pairs of molecules, is an important aspect in identifying likely drug candidates (Husby et al., 2015; Jiménez-Luna et al., 2022). While domain-specific methods such as connectivity fingerprinting and molecular property scores have been the pinnacle of cliff predictions, deep learning methods have for long struggled on this challenging task (Xia et al., 2023). Detection of activity cliffs requires diverse molecular embeddings which spread over a range of cosine similarity values and simultaneously minimize the SALI score as presented in Equation 4. Here $\mathbf{m}_i$ is the candidate molecule, $\mathbf{m}_T$ is the target molecule and $\triangle$pIC50 is the difference in activities.

$$\text{SALI} = \frac{|\triangle \text{pIC50}|}{1 - \texttt{cosine\_sim}(\mathbf{m}_i, \mathbf{m}_T)} \tag{4}$$

A high SALI score denotes sharper cliffs whereas a lower SALI score spread over cosine similarity values denotes lower gaps in molecular activity and hence, a smoother activity landscape.

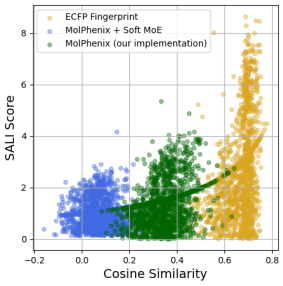

Figure 7: Variation of SALI score with cosine similarity for zero-shot activity cliff prediction methods (↓ is better). Compared to state-of-the-art MolPhenix, addition of Soft MoEs better flattens the SALI landscape.

With the aim of evaluating diversity in MoE embeddings and assessing their utility for real-world biological tasks, we conduct activity cliff prediction experiments on the open-source MPRO activity dataset[1] consisting of 2000 screened molecules corresponding to COVID mutations (Consortium et al., 2020). All compounds contained in the dataset demonstrate high biological activity with a risk of severe toxicity including cardiac impairment, liver or renal damage, carcinogenicity, mutagenicity, teratogenicity and severe allergic responses.

Figure 10 presents the variation of SALI scores with cosine similarity values for different pretrained methods used in the detection of activity cliffs. We compare the addition of Soft MoEs with state-of-the-art MolPhenix and domain-specific ECFP fingerprints (Landrum et al., 2013). Compared to ECFP fingerprints, MolPhenix lowers SALI scores resulting in an improved activity cliff landscape. Furthermore, the addition of Soft MoEs best flattens the landscape resulting in 1.2 $\times$ lower SALI scores. MoE representations demonstrate zero-shot postivie transfer to active MPRO molecules, hence pushing the state-of-the-art for cliff detection.

6.5 Amortizing Inference with GFlowNets

Our final task focusses on the design of molecules. Inferring novel candidates with similar chemical properties is at the pinnacle of learning-driven drug discovery. Furthermore, it is non-trivial to sample such candidates from the embedding space of a large pretrained prior. We tackle this challenging problem by amortizing inference over a pretrained CLIP prior. Our setting leverages GFlowNets (Bengio et al., 2021) to approximate the posterior distribution by learning a stochastic sampling policy from the embeddings of CLIP as states.

Formally, the GFlowNet utilizes the embedding of molecular encoder $\mathbf{z}_{\mathbf{m}}^t$ as state, samples a compound fragment using its forward policy $\mathbf{a} \sim \pi(\mathbf{a}|\mathbf{z}_{\mathbf{m}})$, and transitions to the next embedding state $\mathbf{z}_{\mathbf{m}}^{t+1}$. We utilize the challenging setting of distractor actions wherein one of the actions is duplicated to inject a noisy policy distribution. Upon completion of the trajectory $\tau = (\mathbf{z}_{\mathbf{m}}^1 \rightarrow \mathbf{z}_{\mathbf{m}}^2 \rightarrow \dots \mathbf{z}_{\mathbf{m}}^T)$, the total forward probability *flow* $P_F = \prod_{t=1}^{T-1} P_F(\mathbf{z}_{\mathbf{m}}^{t+1}|\mathbf{z}_{\mathbf{m}}^t)$ observes a reward $\mathbf{r}$. Since our objective is to sample chemically similar molecules, we utilize the cosine similarity (scaled by temperature α) between the sampled molecule $\tilde{\mathbf{x}}$ and a set of heldout *anchor molecules* $\mathbf{m}_{\text{anch}} \in \mathfrak{M}$ as our reward function (Equation 5). GFlowNet samplers are trained using the Trajectory Balance objective (Madan et al., 2023).

$$\mathbf{r}(\tilde{\mathbf{x}}) = \alpha \times (1 + \texttt{cosine\_sim}(\tilde{\mathbf{x}}, \mathbf{m}_{\text{anch}})) \quad ; \quad \mathbf{m}_{\text{anch}} \in \mathfrak{M} \tag{5}$$

Figure 8 (left) presents the variation of laplacian similarity with average episode return when sampling molecules from different CLIP priors. Addition of Soft MoE in the prior leads to generated candidates that are both structurally similar and rewarding against heldout anchor molecules. Figure 8 (center-left) assesses the variation of QED score and laplacian similarity. Soft MoE embeddings guide the GFlowNet sampler in generating compounds that have a higher drug likelihood. Peak QED scores reach a value of 0.42, a 2.3$\times$ gain over 0.18 peak QED score of standard CLIP prior. Figure 8 (center-right) compares the variation of QED score with average return indicating that samples generated from the MoE prior are chemically sound (higher QED) and structurally similar (higher return). Finally, we study the distribution of action preferences induced by both CLIP priors in the forward policy of GFlowNet. Figure 8 (right) presents action counts of fragments selected by the forward policy during GFlowNet training over 20 random runs. Traditional CLIP prior induces a policy which frequently selects chemically irrelevant halogen fragments and distractor duplicate benzene rings. These actions result in large molecules with chemically unsound compositions and dissimilar structures. Addition of Soft MoE in the prior, on the other hand, leads to a policy that prefers smaller organic and hydroxide groups. These make up the majority of molecules in our pretraining distribution with high drug likelihoods and chemically realizable structures.

[1] https://covid.postera.ai/covid/activity_data

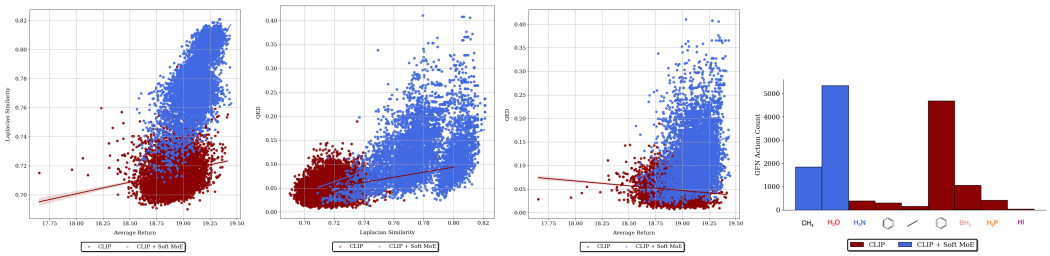

Figure 8: (**left**) Variation of laplacian similarity with average return (↗ is better). (**center-left**) Variation of QED score with laplacian similarity (↗ is better). (**center-right**) Variation of QED score with average return (↗ is better). All variations are over 2000 randomly sampled molecules (enlarged figures in Appendix F). (**right**) Action preferences learned by GFlowNet forward policy.

7 Conclusion

We studied the utility of MoEs through the lens of contrastive phenomic retrieval. By utilizing lab experiment images and molecular structure fingerprints as modalities, we learned a joint embedding space over molecules and cell phenomes. We observed that CLIP-based objectives overfit to the underlying cosine similarity metric. Embedding logits of both encoders collapse in a small value range and remain similar to each other. MoEs address this key phenomenon in training of CLIP by providing a higher-level of diversity in encoder representations. Our experiments empirically validated the efficacy of MoEs in preventing similarity overfitting and improving retrieval performance by $1.23\times$ over previous CLIP methods, and by 7.07% over previously performant contrastive phenomic models. Additionally, we showed that MoE representations demonstrate a positive transfer on few-shot and zero-shot downstream tasks of concentration prediction, molecule activity recognition, activity cliff prediction, gene knockout identification and amortized inference with GFlowNets.

Limitations and Future Work: While our setting studies molecule-cellular interactions, we reserve two aspects for future work. Firstly, our setting can be extended to accommodate additional modalities such as molecule structures. These would inform the model of the geometry of each molecule. Lastly, MoE representations can be transferred to additional downstream tasks which require a higher degree of versatility such as protein binding and target rescue.

Author Contributions

KS discovered similarity overfitting with CLIP, developed the idea of using MoEs, conducted experiments and wrote sections 1, 2, 4, 5 and 6 of the paper. **PAM** implemented the initial version of the code base, ran baseline experiments with CLIP and wrote section 3 of the paper. **FW** helped write sections of the paper and edited the initial draft. **MS** implemented the data pipeline, extracted datasets and preprocessed modalities. **EB** helped setup the GFlowNet experiments, provided feedback on inference and metrics and helped position the paper. **EN** and **DB** provided guidance on CLIP architecture, experimental setup and overall direction of the project. All authors contributed equally to the reviewing and editing of the final version.

Acknowledgments

We thank the broader Valence Labs community for providing feedback on various versions of the manuscript. We thank Ihab Bendidi, Philip Fradkin, Miruna Cretu and Yassir El Mesbahi for helpful discussions. We acknowledge Berton Earnshaw for providing the computational resources. This work is supported by Recursion Pharmaceuticals.

References

Patarajarin Akarapipad, Kattika Kaarj, Yan Liang, and Jeong-Yeol Yoon. Environmental toxicology assays using organ-on-chip. *Annual Review of Analytical Chemistry*, 14:155–183, 2021.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

Saleh Albelwi. Survey on self-supervised learning: auxiliary pretext tasks and contrastive learning methods in imaging. *Entropy*, 24(4):551, 2022.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023.

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv preprint arXiv:2111.02358*, 2021.

Dominique Beaini, Shenyang Huang, Joao Alex Cunha, Gabriela Moisescu-Pareja, Oleksandr Dymov, Samuel Maddrell-Mander, Callum McLean, Frederik Wenkel, Luis Müller, Jama Hussein Mohamud, et al. Towards foundational models for molecular learning on large-scale multi-task datasets. 2024.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *The Journal of Machine Learning Research*, 24(1):10006–10060, 2023.

Regine S. Bohacek, Colin McMartin, and Wayne C. Guida. The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal Research Reviews*, 16(1): 3–50, January 1996. ISSN 1098-1128.

Mark-Anthony Bray, Shantanu Singh, Han Han, Chadwick T Davis, Blake Borgeson, Cathy Hartland, Maria Kost-Alimova, Sigrun M Gustafsdottir, Christopher C Gibson, and Anne E Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature protocols*, 11(9):1757–1774, 2016.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*, 2024.

Shuhao Cao, Peng Xu, and David A Clifton. How to understand masked autoencoders. *arXiv preprint arXiv:2202.03670*, 2022.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding mixture of experts in deep learning. *arXiv preprint arXiv:2208.02813*, 2022.

MinGyu Choi, Wonseok Shin, Yijingxiu Lu, and Sun Kim. Triangular contrastive learning on molecular graphs. *arXiv preprint arXiv:2205.13279*, 2022.

COVID Moonshot Consortium, Hagit Achdout, Anthony Aimon, Dominic S Alonzi, Robert Arbon, Elad Bar-David, Haim Barr, Amir Ben-Shmuel, James Bennett, Vitaliy A Bilenko, et al. Open science discovery of potent non-covalent sars-cov-2 main protease inhibitors. *BioRxiv*, pp. 2020–10, 2020.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512. PMLR, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yin Fang, Qiang Zhang, Ningyu Zhang, Zhuo Chen, Xiang Zhuang, Xin Shao, Xiaohui Fan, and Huajun Chen. Knowledge graph-enhanced molecular contrastive learning with functional prompt. *Nature Machine Intelligence*, 5(5):542–553, 2023.

Marta M Fay, Oren Kraus, Mason Victors, Lakshmanan Arumugam, Kamal Vuggumudi, John Urbanik, Kyle Hansen, Safiye Celik, Nico Cernek, Ganesh Jagannathan, et al. Rxrx3: Phenomics map of biology. *Biorxiv*, pp. 2023–02, 2023.

Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022.

Philip Fradkin, Puria Azadi, Karush Suri, Frederik Wenkel, Ali Bashashati, Maciej Sypetkowski, and Dominique Beaini. How molecules impact cells: Unlocking contrastive phenomolecular retrieval. *Advances in Neural Information Processing Systems*, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pp. 4182–4192. PMLR, 2020.

Markus Hofmarcher, Elisabeth Rumetshofer, Djork-Arne Clevert, Sepp Hochreiter, and Gunter Klambauer. Accurate prediction of biological assays with high-throughput microscopy images and convolutional networks. *Journal of chemical information and modeling*, 59(3):1163–1171, 2019.

Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*, 2023.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models, 2023.

Jarmila Husby, Giovanni Bottegoni, Irina Kufareva, Ruben Abagyan, and Andrea Cavalli. Structure-based predictions of activity cliffs. *Journal of chemical information and modeling*, 55(5):1062–1076, 2015.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Moksh Jain, Tristan Deleu, Jason Hartford, Cheng-Hao Liu, Alex Hernandez-Garcia, and Yoshua Bengio. GFlowNets for AI-Driven Scientific Discovery. *Digital Discovery*, 2023a.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcıa, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-Objective GFlowNets. *International Conference on Machine Learning*, 2023b.

Hyosoon Jang, Minsu Kim, and Sungsoo Ahn. Learning Energy Decompositions for Partial Inference of GFlowNets. *International Conference on Learning Representations*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

José Jiménez-Luna, Miha Skalic, and Nils Weskamp. Benchmarking molecular feature attribution methods with activity cliffs. *Journal of Chemical Information and Modeling*, 62(2): 274–283, 2022.

Oren Kraus, Kian Kenyon-Dean, Saber Saberian, Maryam Fallah, Peter McLean, Jess Leung, Vasudev Sharma, Ayla Khan, Jia Balakrishnan, Safiye Celik, Dominique Beaini, Maciej Sypetkowski, Chi Vicky Cheng, Kristen Morse, Maureen Makes, Ben Mabey, and Berton Earnshaw. Masked autoencoders for microscopy are scalable learners of cellular biology, 2024.

Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8(31.10):5281, 2013.

Elaine Lau, Stephen Zhewen Lu, Ling Pan, Doina Precup, and Emmanuel Bengio. QGFN: Controllable Greediness with Action Values. 2024.

Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. Uni-moe: Scaling unified multimodal llms with mixture of experts. *arXiv preprint arXiv:2405.11273*, 2024.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models. *arXiv preprint arXiv:2406.18219*, 2024.

Kanika Madan, Jarrid Rector-Brooks, Maksym Korablyov, Emmanuel Bengio, Moksh Jain, Andrei Nica, Tom Bosc, Yoshua Bengio, and Nikolay Malkin. Learning GFlowNets from partial episodes for improved convergence and stability. *International Conference on Machine Learning*, 2023.

Sobhan Mohammadpour, Emmanuel Bengio, Emma Frejinger, and Pierre-Luc Bacon. Maximum entropy GFlowNets with soft Q-learning. *International Conference on Artificial Intelligence and Statistics*, 2024.

Oscar Méndez-Lucio, Christos Nicolaou, and Berton Earnshaw. Mole: a molecular foundation model for drug discovery, 2022.

Jun Wei Ng and Marc Peter Deisenroth. Hierarchical mixture-of-experts model for large-scale gaussian process regression. *arXiv preprint arXiv:1412.3078*, 2014.

Cuong Q Nguyen, Dante Pertusi, and Kim M Branson. Molecule-morphology contrastive pretraining for transferable molecular representation. *bioRxiv*, pp. 2023–05, 2023.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018a.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018b.

Ling Pan, Moksh Jain, Kanika Madan, and Yoshua Bengio. Pre-training and fine-tuning generative flow networks. *arXiv preprint arXiv:2310.03419*, 2023.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*, 2023.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data, 2020.

Ana Sanchez-Fernandez, Elisabeth Rumetshofer, Sepp Hochreiter, and Günter Klambauer. Cloome: contrastive learning unlocks bioimaging databases for queries with chemical structures. *Nature*, 2023.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pp. 1332–1344. PMLR, 2023.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Max W Shen, Emmanuel Bengio, Ehsan Hajiramezanali, Andreas Loukas, Kyunghyun Cho, and Tommaso Biancalani. Towards Understanding and Improving GFlowNet Training. *International Conference on Machine Learning*, 2023.

Jaak Simm, Günter Klambauer, Adam Arany, Marvin Steijaert, Jörg Kurt Wegner, Emmanuel Gustin, Vladimir Chupakhin, Yolanda T Chong, Jorge Vialard, Peter Buijnsters, et al. Repurposing high-throughput image assays enables biological activity prediction for drug discovery. *Cell chemical biology*, 25(5):611–618, 2018.

Rohit Singh, Samuel Sledzieski, Bryan Bryson, Lenore Cowen, and Bonnie Berger. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proceedings of the National Academy of Sciences*, 120(24):e2220778120, 2023.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 1857–1865, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Rakshith Sharma Srinivasa, Jaejin Cho, Chouchang Yang, Yashas Malur Saidutta, Ching-Hua Lee, Yilin Shen, and Hongxia Jin. Cwcl: Cross-modal transfer with continuously weighted contrastive loss. *Advances in Neural Information Processing Systems*, 36, 2023.

Maciej Sypetkowski, Frederik Wenkel, , Farimah Poursafaei, Nia Dickson, Karush Suri, Philip Fradkin, and Dominique Beaini. On the scalability of foundational models for molecular graphs. *arxiv*, 2024.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *arXiv preprint arXiv:2405.20971*, 2024.

Fabien Vincent, Arsenio Nueda, Jonathan Lee, Monica Schenone, Marco Prunotto, and Mark Mercola. Phenotypic drug discovery: recent successes, lessons learned and new directions. *Nature Reviews Drug Discovery*, 21(12):899–914, 2022.

Jun Xia, Lecheng Zhang, Xiao Zhu, Yue Liu, Zhangyang Gao, Bozhen Hu, Cheng Tan, Jiangbin Zheng, Siyuan Li, and Stan Z. Li. Understanding the limitations of deep models for molecular property prediction: Insights and solutions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=NLFqlDeuzt.

Ronald Xie, Kuan Pang, Gary D. Bader, and Bo Wang. Maester: Masked autoencoder guided segmentation at pixel resolution for accurate, self-supervised subcellular structure recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023.

Ziwei Yang, Zheng Chen, Yasuko Matsubara, and Yasushi Sakurai. Moclim: Towards accurate cancer subtyping via multi-omics contrastive learning with omics-inference modeling. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 2895–2905, 2023.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

Haofei Yu, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. Mmoe: Mixture of multimodal interaction experts. *arXiv preprint arXiv:2311.09580*, 2023.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction, 2022.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023.

Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. Llama-moe: Building mixture-of-experts from llama with continual pre-training. *arXiv preprint arXiv:2406.16554*, 2024.

Yiheng Zhu, Jialu Wu, Chaowen Hu, Jiahuan Yan, Tingjun Hou, Jian Wu, et al. Sample-efficient Multi-objective Molecular Optimization with GFlowNets. *International Conference on Neural Information Processing Systems*, 2023.

A Additional Related Work

Molecular Representation Learning: Contrastive phenomic models of today utilize the InfoNCE objective as a successful pretraining strategy (Oord et al., 2018b; Albelwi, 2022). (Nguyen et al., 2023) propose a multimodal model with modalities as hand-engineered visual features and a molecular Graph Neural Network (GNN) encoder. (Poole et al., 2019) build on the InfoNCE paradigm to enable molecular retrieval across CLIP-based encoders. Specifically, CLOOME (Sanchez-Fernandez et al., 2023) utilizes the InfoLOOB objective (Poole et al., 2019) with hopfield networks for zero-shot retrieval of phenotype embeddings. (Singh et al., 2023) leverage contrastive learning across molecules and text to predict and screen drug-target candidate pairs. (You et al., 2020) attempt a similar model by encoding geometry of molecular structures. From a structural perspective, GNNs have been used to predict molecular properties by reasoning over graph structures. A combination of reconstruction and supervised objectives have led to models generalizing to a diverse range of prediction tasks (Méndez-Lucio et al., 2022; Zhou et al., 2023; Sypetkowski et al., 2024; Rong et al., 2020). (Fang et al., 2023) present an orthogonal approach by augmenting contrastive pretraining with knowledge graphs to preserve molecular semantics. During the finetuning stage, functional prompts (Liu et al., 2023) are used to invoke task-specific knowledge from learned embeddings. Finally, (Choi et al., 2022) accomodate the angular geometry of embedding space by proposing a novel triangular area loss over molecular graph representations. Our experimental setup and empirical analysis is in line with the above directions.

Generative Flow Networks for Drug Design: GFlowNets (Bengio et al., 2021) are a class of generative models which learn to sample from a data distribution by learning a stochastic policy over preferences. Traditionally, GFlowNets are trained using the Trajectory Balance (TB) loss which, for each node in the sampling trajectory, matches the inflow probability to the outflow probability (Bengio et al., 2023). Recent advances in GFlowNets aim to improve and stabilize training for molecule generation and drug design (Jain et al., 2023a). (Jang et al., 2024) learn partial energy decompositions of GFlowNet with the intent of providing a stable learning objective when compared to TB loss. (Madan et al., 2023) enable data-efficient GFlowNet training using partial incomplete episodes. Such a training strategy results in improved convergence and stable stochastic sampling. (Shen et al., 2023) attempt to understand training bottlenecks in GFlowNet training and propose stability measures for forward and backward policies. (Jain et al., 2023b) propose to learn GFlowNet samplers across a wide array of objective functions. This aids in learning of sampling diverse candidates in an active learning setting. (Zhu et al., 2023) extend the multi-objective GFlowNet training approach towards sampling of molecular sub-structures by utilizing off-policy data aggregation and improvement.

Recent methods have borrowed advances from Reinforcement Learning (RL) to control the diversity in sampled candidates. On one hand, (Lau et al., 2024) tailor the sampler to controllably exploit sampling of high-rewarding molecules. On the other hand, (Mohammadpour et al., 2024) aim to study the diversity of GFlowNets by drawing a connection to maximum-entropy Q learning. Finally, even recent methods study the behavior of GFlowNets for downstream training and inference. (Pan et al., 2023) study transfer learning capabilities of learned policies. (Hu et al., 2023) amortize inference in large language models using the TB objective. (Venkatraman et al., 2024) improve the TB loss to strike a balance between the prior and posterior for amortizing inference in diffusion models. Our evaluation scheme leverages GFlowNets for downstream inference.

B PROOF

Theorem 1. *Given a linear expert g_i with parameters $\mathbf{\Phi}_i$, input features $\mathbf{Z}$, layernorm operator LN(·) with parameters α and β and softmax(·) operator , width of expert network g_i, denoted by d, scales inversely with the squared ratio of layernorm parameters, $d \propto \left(\frac{\beta}{\alpha}\right)^2$.*

Proof. We closely follow the approach of (Puigcerver et al., 2023) by first expanding the layernorm of the unit-normalized input and then computing the softmax of the logits.

$$LN(\mathbf{Z}_i) = \alpha \frac{\mathbf{Z}_i - \mu(\mathbf{Z}_i)}{\sigma(\mathbf{Z}_i)} + \beta \tag{6}$$

Rewriting the layernorm with respect to the centered vector $\tilde{\mathbf{Z}} = \mathbf{Z} - \mu(\mathbf{Z})$, and the vector scaled to have unit norm $\hat{\mathbf{Z}} = \frac{\tilde{\mathbf{Z}}_i}{\|\tilde{\mathbf{Z}}\|}$,

$$LN(\tilde{\mathbf{Z}}_i) = \alpha \frac{\tilde{\mathbf{Z}}_i}{\sqrt{\frac{1}{d}\sum_{j=1}^{d} \tilde{\mathbf{Z}}_j^2}} + \beta = \sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta \tag{7}$$

We now compute $h_{\mathbf{\Phi}}(\mathbf{Z}_i) = softmax(\mathbf{\Phi} LN(\mathbf{Z}_i))$.

$$h_{\mathbf{\Phi}}(\mathbf{Z}_i) = softmax(\mathbf{\Phi}(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta)) \tag{8}$$

$$h_{\mathbf{\Phi}}(\mathbf{Z}_i) = \frac{\exp\left(\sum_{k=1}^{d} \mathbf{\Phi}_{ik}(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta)\right)}{\sum_{j=1}^{n} \exp\left(\sum_{k=1}^{d} \mathbf{\Phi}_{jk}(\sqrt{d}\alpha \hat{\mathbf{Z}}_j + \beta)\right)} \tag{9}$$

Computing the gradient of $h_{\mathbf{\Phi}}(\mathbf{Z}_i)$ with respect to $\mathbf{\Phi}$ and setting it to zero,

$$\nabla_{\mathbf{\Phi}_i} h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) = \nabla_{\mathbf{\Phi}_i} \left(\frac{\exp\left(\sum_{k=1}^{d} \mathbf{\Phi}_{ik}(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta)\right)}{\sum_{j=1}^{n} \exp\left(\sum_{k=1}^{d} \mathbf{\Phi}_{jk}(\sqrt{d}\alpha \hat{\mathbf{Z}}_j + \beta)\right)} \right) \tag{10}$$

$$= \nabla_{\mathbf{\Phi}_i} \left(\frac{\exp\left(\sum_{k=1}^{d} \mathbf{\Phi}_{ik}(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta)\right)}{\sum_{j=1}^{n} \exp\left(\sum_{k=1}^{d} \mathbf{\Phi}_{jk}(\sqrt{d}\alpha \hat{\mathbf{Z}}_j + \beta)\right)} \right) \left[\sum_{k=1}^{d} \underbrace{\nabla_{\mathbf{\Phi}_i} \mathbf{\Phi}_{ik}}_{=1} \left(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta\right) \right] \tag{11}$$

$$= \nabla_{\mathbf{\Phi}_i} h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) \left[\sum_{k=1}^{d} \left(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta\right) \right] \tag{12}$$

Since $h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) = softmax(\cdot)$,

$$\nabla_{\mathbf{\Phi}_i} h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) = \begin{cases} h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i)(1 - h_{\mathbf{\Phi}_j}(\hat{\mathbf{Z}}_j)), & \text{if } i = j \\ -h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) h_{\mathbf{\Phi}_j}(\hat{\mathbf{Z}}_j), & \text{otherwise} \end{cases} \tag{13}$$

Since we consider $\mathbf{\Phi}_i$, we are interested in the case where $i = j$,

$$h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) = h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i)(1 - h_{\mathbf{\Phi}_j}(\hat{\mathbf{Z}}_j)) \left[\sum_{k=1}^{d} \underbrace{\nabla_{\mathbf{\Phi}_i} \mathbf{\Phi}_{ik}}_{=1} \left(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta\right) \right] \tag{14}$$

Setting the gradient to zero, we get,

$$h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i)(1 - h_{\mathbf{\Phi}_j}(\hat{\mathbf{Z}}_j)) \left[\sum_{k=1}^{d} \underbrace{\nabla_{\mathbf{\Phi}_i} \mathbf{\Phi}_{ik}}_{=1} \left(\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta \right) \right] = 0 \tag{15}$$

Since $h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) \in (0, 1)$ by definition, $h_{\mathbf{\Phi}_i}(\hat{\mathbf{Z}}_i) \neq 0$ and $1 - h_{\mathbf{\Phi}_j}(\hat{\mathbf{Z}}_j) \neq 0$.

Thus, $\sqrt{d}\alpha \hat{\mathbf{Z}}_i + \beta = 0$.

In vector form,

$$\sqrt{d}\alpha \hat{\mathbf{Z}} + \beta = 0 \tag{16}$$

$$\sqrt{d} = -\frac{\beta}{\alpha} \frac{1}{\hat{\mathbf{Z}}} \tag{17}$$

$$d = \frac{\beta^2}{\alpha^2} \frac{1}{\|\hat{\mathbf{Z}}\|} \tag{18}$$

As $\|\hat{\mathbf{Z}}\| = 1$ (by definition of unit normalized vector), and α and β are bounded as $\alpha < \alpha_c$ and $\beta < \beta_c$, $d \propto \left(\frac{\beta}{\alpha}\right)^2$. □

C Implementation Details

C.1 Pretraining with Representation Diversity

Our pretraining strategy closely follows the setup of standard CLIP training. In the case of smaller models, we only utilized 1 encoder block with tuned learning rates. Explicit learning rate schedules were not used. For larger models exceeding 250M parameters, we use custom learning rate schedules to kickstart training. A common strategy we utilize for 250M and 1B parameter models is the use of warm starts learning rate. Learning rate is set to 1e-6 and increased by a factor of 1.01 every 100th step for the first 1000 steps.

We experimented with two warms starts strategies. In the first case, which we denote as fast increment, we increase the learning rate every 100th step. In the second case, which we call slow increment, we increase the learning rate every 500th step. We empirically found fast increment to work well and stabilize CLIP training.

Similar to warm starts learning, we decay learning rates for larger model sizes. Following 30,000 steps of training, learning rates for 250M and 1B parameter models are decayed by a factor of 0.99 at intervals of every 100 steps.

In the case of encoder architectures, we found wider encoders to work best. Increasing the depth of MLP and Transformer encoders also yielded favourable results. In the case of MLP encoders, we tuned the number of blocks between 2, 4, 6 and 8 with 8 being the best configuration. For Transformer models, we tuned the number of encoder blocks between 2, 4, 8, 12, 16 and 24 with 24 being the best configuration. As for width, we tuned between 256, 512, 1024, 2048 and 4096. In the case of MLP models, 4096 was found to work the best whereas for transformer models we were only able to accomodate 2048 due to the large memory requirements posed by attention layers.

For all architectures, weight decay was found to benefit training while dropout had little to no effect. In the case of MoEs, we kept the number of experts fixed to 8 and the number of slots per expert fixed to 32. In general, larger number of slots per expert were found to benefit training.

C.2 Few-Shot Concentration Prediction

Our few-shot concentration prediction experiment utilized the 250M parameter model for probing. We freeze both the encoders and add a linear layer along with sigmoid nonlinearity to predict the threshold of concentration. Since we are predicting the concentration modalities themselves, we blocked the one-hot encoded context input and only provided an empty mask in its place. This allowed us to fairly evaluate our model's latent space of cell concentration information. Thus, we only provided the image and molecule fingerprint embedding to our downstream classification head.

The task, thus being of binary prediction, only required a handful of samples. In the case of concentration = 1.0, we utilized a total of 300 shots. For concentration = 0.1, only 20 shots were utilized. We specifically chose these concentrations as these possessed the most number of noisy embeddings, noisy molecule samples and active molecule perturbations. While both CLIP and CLIP+Soft MoE models were stable, the addition of MoEs accelerated downstream learning across both concentrations.

C.3 Few-Shot Activity Recognition

Our few-shot activity recognition is formulated as a regression problem wherein we train a downstream linear classifier to predict perturbation values corresponding to each molecule. We do not employ a nonlinearity and purely train the regressor in an online setting. Since the perturbation value is specific to each molecule and does not exist for molecule-image paired samples, we only utilized the molecular embedding as input to our linear regressor.

Similar to our concentration prediction experiments, only a handful of samples were required. We restricted the number of shots to 300. We trained the linear regressor with a

smaller learning rate of 0.0001 than the concentration prediction task since a higher value demonstrated instability. Regressor convergence was stable in both CLIP models.

Contrary to concentration prediction, we observed an implicit regularization effect when training the regressor from embeddings of CLIP. In the case of CLIP+Soft MoEs, this effect does not arise and the parameter norm grows at a sublinear rate. However, in the case of CLIP, parameter norm continued to decrease at a constant rate. While this phenomenon does not lead to a drop in performance, we leave this as an investigation for future work.

C.4 Amortizing Inference with GFlowNets

Our downstream GFlowNet experiments utilize GFlowNet as a decoder to generate high-fidelity diverse samples. Contrary to prior downstream tasks, GFlowNets required significant tuning and a systematic assessment of their reward function. We thus, experimented with different GFlowNet architectures, reward functions and action fragments. Empirically, we found that the choice of reward function and action fragment set play the most important roles in the generation of diverse molecular candidates.

In the case of reward function, we initially tried the cosine similarity between the generated molecule and a set of heldout molecules as the terminal reward. The formulation is expressed below.

$$r(\tilde{\mathbf{x}}) = \texttt{cosine\_sim}(\tilde{\mathbf{x}}, \mathfrak{M}) \quad ; \quad \mathfrak{M} : \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N\} \tag{19}$$

The above reward function did not converge well for larger molecules and a diverse heldout set. Additionally, GFlowNets often underwent mode collapse wherein the sampler continued to sample similar molecule fragments. With the aim of increasing diversity in our sampling process, we then inverted the above reward function as presented below.

$$r(\tilde{\mathbf{x}}) = \min_{\mathfrak{M}} \frac{1}{1 + \texttt{cosine\_sim}(\tilde{\mathbf{x}}, \mathfrak{M})} \quad ; \quad \mathfrak{M} : \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N\} \tag{20}$$

The above reward function aims to a maximize a lower bound on diversity. While effective, the function did not yield chemically plausible molecules. Additionally, these molecules were both structurally and chemically different from the our pretraining set, a property not desirable for screening drug candidates.

We thus switched to our original reward function by making a small modification to increase stability. Instead of utilizing a large heldout set, we selected the best molecules which we wanted to optimize against. These molecules were used as our anchor set in the latent space in order to guide the sampler. Additionally, we switched from an unconditional generation setup to a conditional generation framework wherein the embedding of anchor molecules was concatenated with the environment state of GFlowNet. Intuitively, we restrict our generation process to a subset of modes which present favourable chemical properties and a smooth optimization landscape. Since the choice of anchor molecules has the most effect on sampling quality and diversity of generated candidates, we iteratively experimented with multiple molecule subsets until we reached a set with high QED scores. Below is our final reward function accomodating the anchor molecule denoted as $\mathbf{m}_{\text{anch}}$.

$$r(\tilde{\mathbf{x}}) = \texttt{cosine\_sim}(\tilde{\mathbf{x}}, \mathbf{m}_{\text{anch}}) \quad ; \quad \mathbf{m}_{\text{anch}} \in \mathfrak{M} \tag{21}$$

Hyperparameter tuning of GFlowNet was carried out over a larger set of values. In the case of learning rate, we found smaller values to work well. We tuned for the following values- 1e-3, 7e-4, 5e-4, 3e-4, 1e-4. While larger values demonstrated instability, the smallest 1e-4 value was found to be stable. Additionally, we increased the number of training episodes from 1000 to 1500 to 2000 as the sampler continued to generate diverse molecules with increased gradient steps. Our choice of action fragments was motivated by the objective of sampling organic molecules with structures interleaving rings and halogen atoms. Since majority of such molecules formed our pretraining distribution, GFlowNets were able to

discover these modes and generate highly similar samples within 200 steps of training. We thus did not experiment further with our action fragment set.

C.5 Dataset Details

We note that our pretraining data distribution consists of 1.6M paired samples containing 205456 unique molecules spanning a total of 14 unique concentrations. While the image encoder utilizes embeddings of raw microsocopy images as input, the molecular encoder utilizes fingerprints generated from valid molecule SMILES as input. We utilize the ECFP fingerprinting technique (Landrum et al., 2013) in order to generate and cache molecule fingerprints. Our input to the molecular encoder consists of MACCS keys and the hashed MORGAN fingerprinting with an atomic radius of 2. When combined together, the input of size 2215 dimensions is passed to the molecular encoder.

Our pretrained MAE encoder downscales each image into an embedding of 768 dimensions which are fed as input to the CLIP image encoders. In addition to the image embedding, we encode the concentration of each phenome experiment and concatenate it with the input embedding. Since we have 13 unique concentrations out of which 8 are the most frequent ones, we utilize one-hot encoding as our encoding strategy to generate the concentration embedding. Other encoding choices, such as log and sigmoid, were also tried but one-hot was empirically found to be more informative.

Both image and molecule encoders utilize the same architecture consisting of 2-4 blocks of MLP/Transformer layers with GeLU activations and skip connections. Each encoder is followed by a LayerNorm projection which is passed to the projection head. Similar to the original CLIP architecture, the projection head acts as a dimensionality-reduction bottleneck wherein the final output embedding is projected and normalized before computation of similarity matrix. Output embeddings of encoders are of size 1024-2048 dimensions wherein larger dimensions are found to be more performant.

To make our experiments reproducible, we setup our code base on the open-source RXRX3 dataset[2], the largest open-sourced dataset consisting of biological phenome maps (Fay et al., 2023). The dataset consists of 2.2M images from the HUVEC cell line. The dataset additionally consists of 1674 known chemical entities across 8 different unique concentrations each. Each image is of dimension $2048 \times 2048 \times 6$ consisting of 6 channels corresponding to experiment stains Hoechst, ConA, Phalloidin, Syto14, MitoTracker and WGA. Each phenome experiment consists of 1536-well plate density containing 1 imaging site per well.

[2] https://rxrx3.rxrx.ai/downloads

D Hyperparameters

D.1 Pretraining with Representation Diversity

Hyperparameter	Value
Embedding size	2048
Batch size	8192
Train split	0.85
Train epochs	100
Dropout rate	0.1
Weight decay 3e-3	
Encoder blocks	8
MoE layers	1
Warm start learning rate	False
Learning rate	0.0001
Decay learning rate	False
Experts	8
Slots per expert	32
Checkpoint interval	25
Validation interval	5000

Table 5: Hyperparameter values for pretraining CLIP with Soft MoEs

D.2 Few-Shot Concentration Prediction

Hyperparameter	Value
Embedding size	2048
Batch size	32
Train split	0.90
Train epochs	1
Dropout rate	0
Weight decay	3e-5
Few shot learning rate	0.01
Few shots	300
Encoder blocks	8
MoE layers	1
Warm start learning rate	False
Learning rate	0.0001
Decay learning rate	False
Experts	8
Slots per expert	32
Checkpoint interval	25
Validation interval	5000

Table 6: Hyperparameter values for concentration prediction with Soft MoEs

D.3 Few-Shot Activity Recognition

Hyperparameter	Value
Embedding size	2048
Batch size	32
Train split	0.90
Train epochs	1
Dropout rate	0
Weight decay	3e-3
Few shot learning rate	0.0001
Few shots	300
Encoder blocks	8
MoE layers	1
Warm start learning rate	False
Learning rate	0.0001
Decay learning rate	False
Experts	8
Slots per expert	32
Checkpoint interval	25
Validation interval	5000

Table 7: Hyperparameter values for activity recognition with Soft MoEs

D.4 Amortizing Inference with GFlowNets

Hyperparameter	Value
Embedding size	2048
Batch size	32
Train split	0.90
Train epochs	1
Dropout rate	0
Weight decay	3e-5
GFlowNet learning rate	0.0001
GFlowNet episodes	2000
Encoder blocks	8
MoE layers	1
Warm start learning rate	False
Learning rate	0.0001
Decay learning rate	False
Experts	8
Slots per expert	32
Checkpoint interval	25
Validation interval	5000

Table 8: Hyperparameter values for concentration prediction with Soft MoEs

E Additional Experiments

E.1 Downstream Implicit Regularization in CLIP

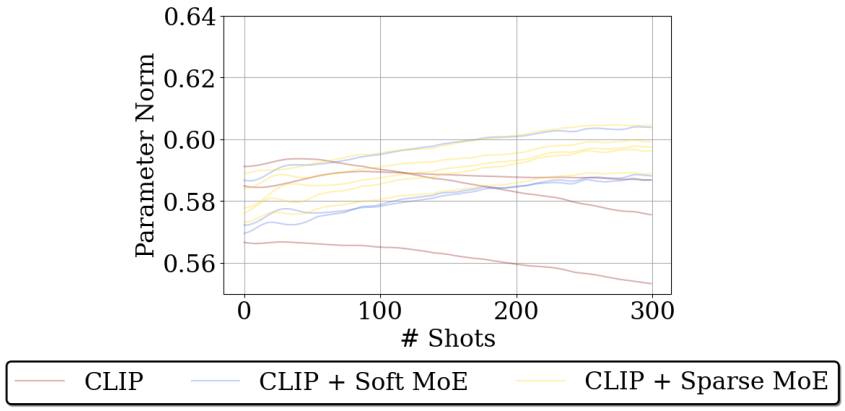

Figure 9: Implicit regularization in downstream linear regressor trained with CLIP embeddings for few-shot activity recognition. While regressors trained with MoE architectures have a steady growth in their parameter norm, parameter norm of regressors trained with CLIP decreases at a constant rate.

Contrary to concentration prediction, we observed an implicit regularization effect when training the regressor from embeddings of CLIP. In the case of CLIP+Soft MoEs and CLIP+Sparse MoEs, this effect does not arise and the parameter norm grows at a sublinear rate. However, in the case of CLIP, parameter norm continued to decrease at a constant rate. While this phenomenon does not lead to a drop in performance, we leave this as an investigation for future work.

E.2 Zero-Shot Activity Cliff Prediction

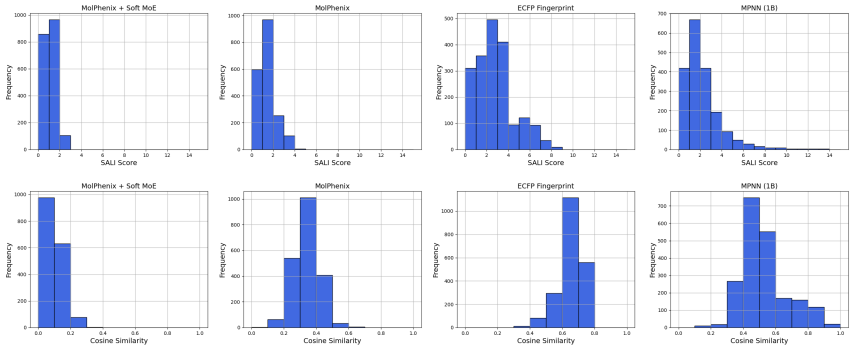

Figure 10: (**left**) Distributions of SALI scores for different pretrained methods on the task of zero-shot activity cliff prediction. MolPhenix with the addition of MoEs achieves lowest SALI scores resulting in a smoother activity cliff landscape. (**right**) Distribution of cosine similarities between molecular embeddings of different pretrained methods. Across all structure-based and domain-specific methods, MolPhenix with MoEs presents highest diversity in embeddings.

Figure 10 presents the distribution of SALI scores and cosine similarity values for different pretrained methods used in the detection of activity cliffs. In addition to MolPhenix, we also compare MoEs with a 1B parameter Message Passing Neural Network (MPNN) (Sypetkowski

et al., 2024) pretrained on open-source molecular graphs (PCBA_1328 dataset) (Beaini et al., 2024). In accordance with the current trend in GNN training literature, we scale the MPNN model for width and depth and utilize a total of 1B parameters to create rich general and meaningful molecular embeddings.

Compared to MPNN, MoEs achieve lower SALI scores demonstrating a smoother activity cliff landscape. Additionally, MolPhenix with MoEs presents diverse embeddings which are marked by lower cosine similarity values. Among all pretrained methods, addition of MoEs results in the most diverse set of embeddings without utilizing structural information of molecules.

E.3 Zero-Shot Gene Knockout Identification

Identifying genetic associations corresponding to phenotypic traits is a key component in the search of new drug candidates. We evaluate the utility of MoEs in identifying unseen genetic correlations corresponding to encoded phenomics. Similar to the setting of MolPhenix (Fradkin et al., 2024), we study the problem of gene knockout identification which aims to identify which genes are activated/deactivated corresponding to a given set of molecules. Our dataset utilizes a set of genes and molecules from the ChemBL gene database. Corresponding to each gene, we extract the embedding using the pretrained unimodal MAE. We repeat the same process for microscopy images and obtain their phenomic embeddings. Gene and phenomic embeddings are passed through the frozen vision encoder to obtain their similarity matrix. Intuitively, the similarity matrix encodes gene-phenomic relationships corresponding to a given molecular perturbation. We utilize this similarity matrix to compute recall using cosine similarity percentiles.

Figure 11 presents the variation of top-1% recall accuracy across different cosine similarity percentiles for molecule perturbations with a p value cutoff threshold of 0.01. Corresponding to different concentrations, MolPhenix closes the gap with ground truth MAE performance recalling approximately 35% of unseen genetic associations. Figure 12 presents the same variation for a p value cutoff threshold of 0.1. We note that the addition of Soft MoE benefits zero-shot recall performance at high concentrations, when the number of gene-molecule data pairs are scarce. Furthermore, we see that compared to MolPhenix, addition of MoEs improves zero-shot recall performance on 6 out of 8 concentrations at a high p value cutoff threshold.

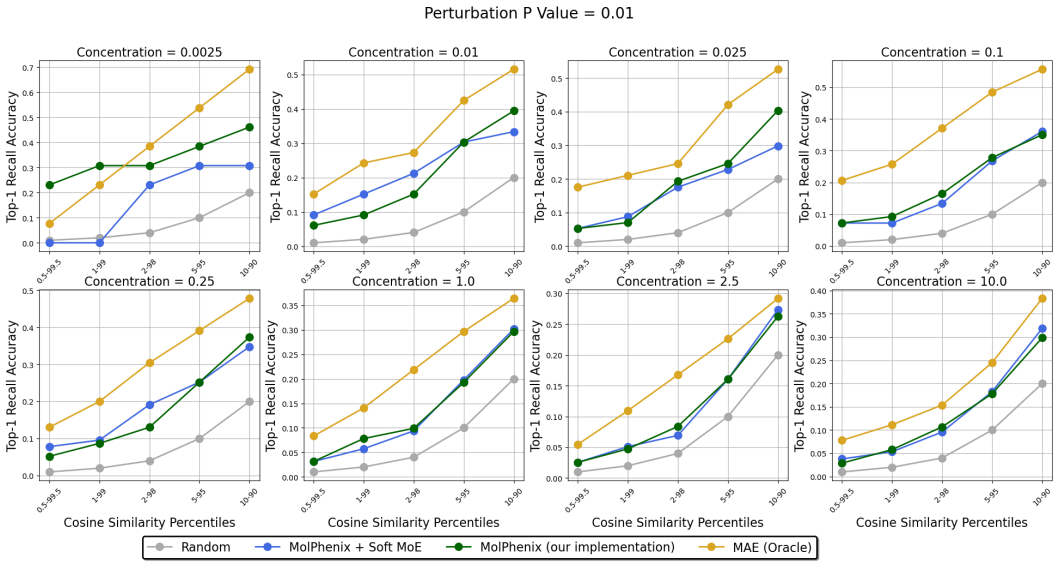

Figure 11: Variation of Top-1 % recall accuracy across different cosine similarity percentiles for perturbation p-value cutoff threshold of 0.01. Addition of Soft MoE to state-of-the-art MolPhenix improves zero-shot recall performance at higher concentrations.

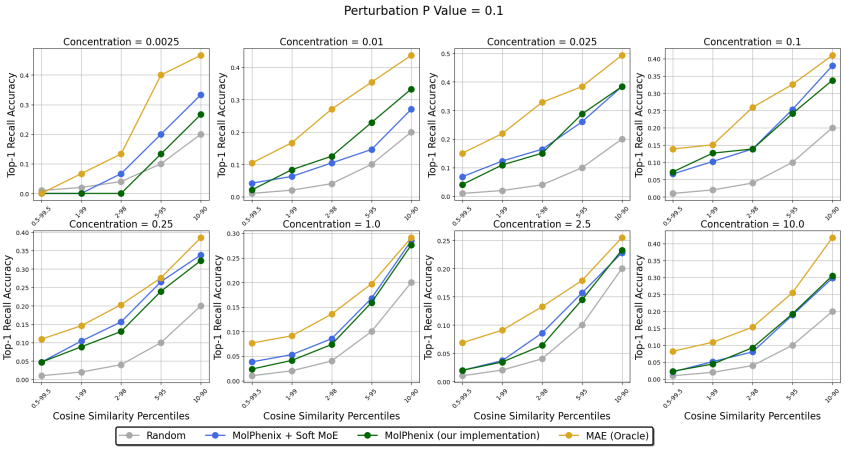

Figure 12: Variation of Top-1 % recall accuracy across different cosine similarity percentiles for perturbation p-value cutoff threshold of 0.1. Addition of Soft MoE to state-of-the-art MolPhenix improves zero-shot recall performance at higher concentrations. MolPhenix with Soft Moe outperforms the MolPhenix baseline on 6 out of 8 molecule concentrations.

E.4 Robustness Ablations

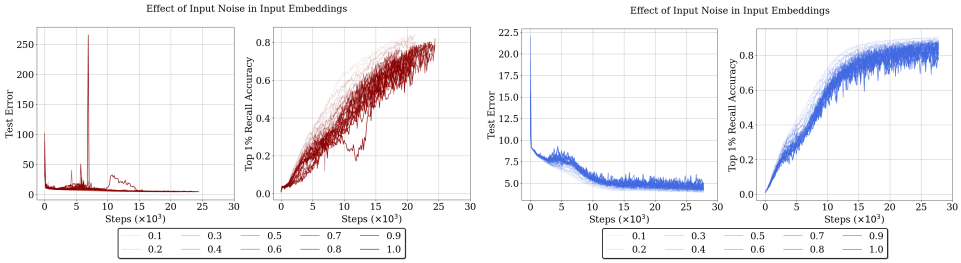

Figure 13: (**left**) Variation of test error and Top-1% recall accuracy with variance of input noise of vision encoder for CLIP. (**right**) Variation of test error and Top-1% recall accuracy with variance of input noise of vision encoder for CLIP with Soft MoEs.

A key component affecting the robustness of molecular retrieval models is the presence of noise in input modalities. We assess how CLIP models behave when combined with MoE encoders in the presence of input noise. We specifically focus on the vision encoder making use of MAE embeddings as this encoder captures phenomic representations from a high-dimensional space susceptible to experiment/instrumentation noise. At training time, random noise is sampled from a gaussian distribution with zero mean and variable variance $\epsilon \sim \mathcal{N}(0; \Sigma)$. Noise is added to the encoder embeddings of each data sample, $\mathbf{x}_i + \epsilon_i$, before training the encoder. At test time, both encoders are evaluated on ground truth uncorrupted embeddings. We conduct an ablation study for different levels of noise by varying the variance Σ.

Figure 13 presents the variation of test error and top-1% recall accuracy for increasing values of Σ, i.e- increasing noise in the MAE embeddings. Gradually corrupting the phenomic embedding hurts molecular retrieval performance, as one would expect. Lower noise levels have little impact on performance, while higher noise levels lead to a steady decline. In the case of CLIP, noise leads to a severe degradation in performance and stability during training. MoE-based encoders, on the other hand, are found robust to significant decreases in performance. CLIP with Soft MoEs converges smoothly with minimal drops in recall accuracies.

F Additional Results

F.1 Pretraining with Representation Diversity

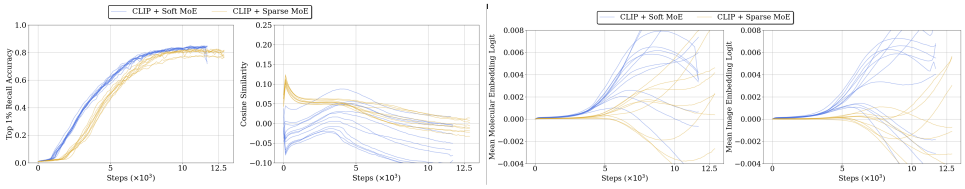

Figure 14: (**left**) Variation of Top-1% molecular retrieval performance and cosine similarity for different MoE architectures. While both MoEs outperform naive CLIP, soft MoE models are more performant due to a higher degree of diversity. (**right**) Variation of encoder embedding logits during training. While both MoEs circumvent similarity overfitting, soft MoEs have a wider spread of logit values. Results are presented across 9 random runs.

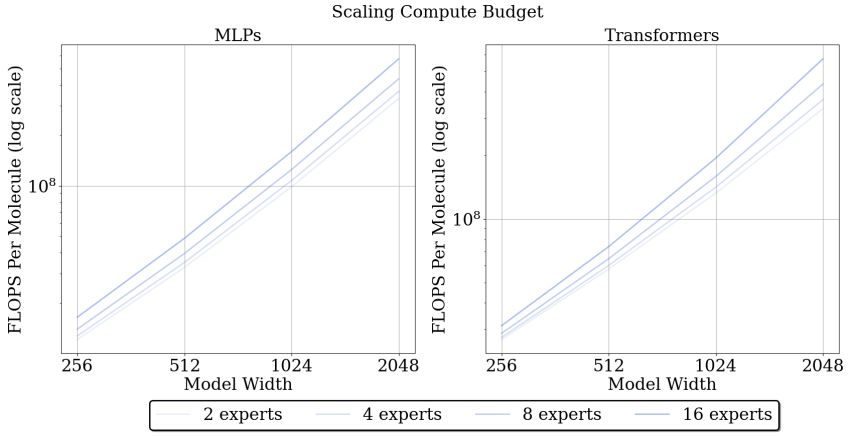

Figure 15: Variation of FLOPS per molecule (log scale) for different model widths across different number of experts. Compute scales linearly for both MLP and Transformer architectures as the size of model and number of experts grow.

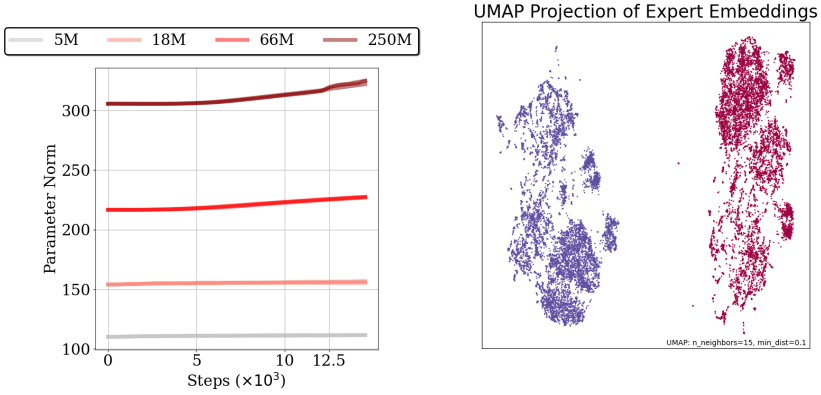

Figure 16: (**left**) Variation of parameter norm for different MoE model sizes. Larger performant models see a faster growth in their paramter norm over training. (**right**) (2 experts) UMAP projection of expert embeddings obtained from the image encoder.

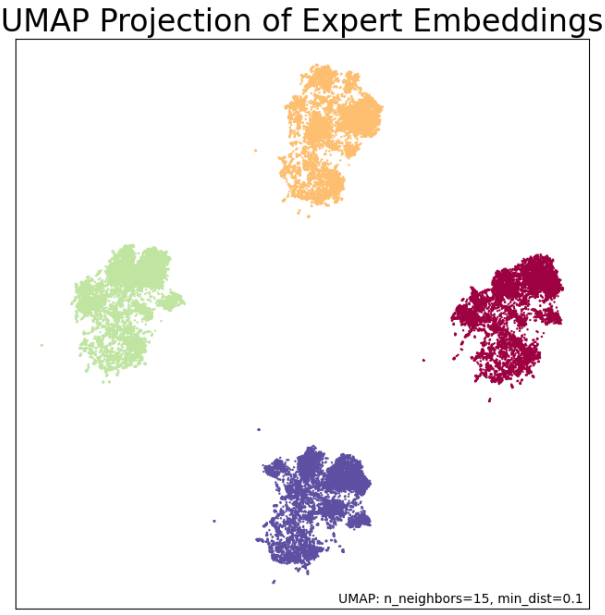

Figure 17: (4 experts) UMAP projection of expert embeddings obtained from the image encoder.

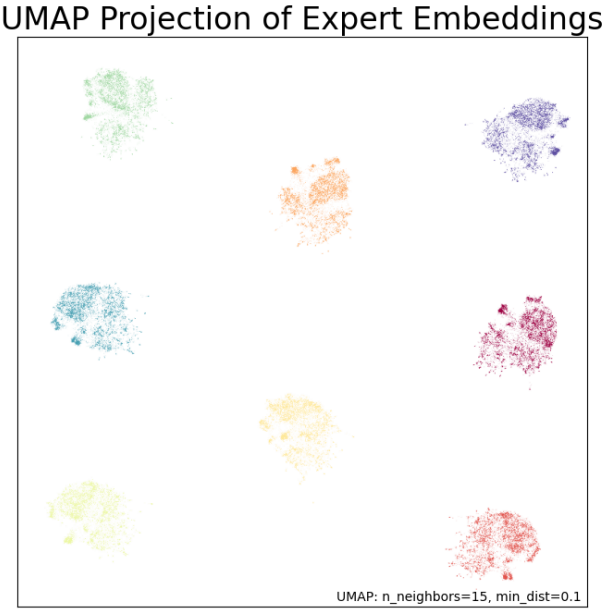

Figure 18: (8 experts) UMAP projection of expert embeddings obtained from the image encoder.

F.2 Few-Shot Concentration Prediction

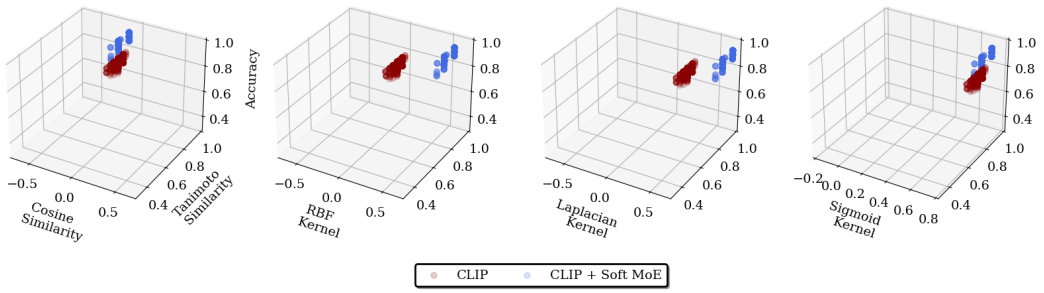

Figure 19: (**concentration**=0.1) Variation of similarity with prediction accuracy. X axis denotes different similarity metrics, Y axis denotes tanimoto similarity and Z axis denotes prediction accuracy. Top-right-back is better. Each point denotes the model embedding obtained corresponding to a molecule. CLIP+Soft MoE embeddings have a higher correlation between similarity and accuracy when compared to CLIP. CLIP+Soft MoEs have high similarity values for RBF, Laplacian and Sigmoid kernels but lower values for cosine similarity. CLIP, on the other hand, has high values only for cosine similarity as a result of similarity overfitting.

F.3 Amortizing Inference with GFlowNets

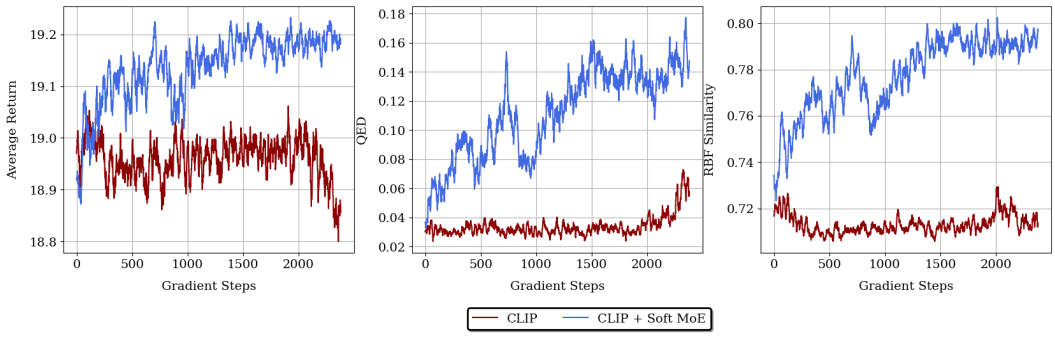

Figure 20: Comparison of average return, QED score and RBF kernel similarity between GFlowNets trained with CLIP and Soft MoE embeddings as states.

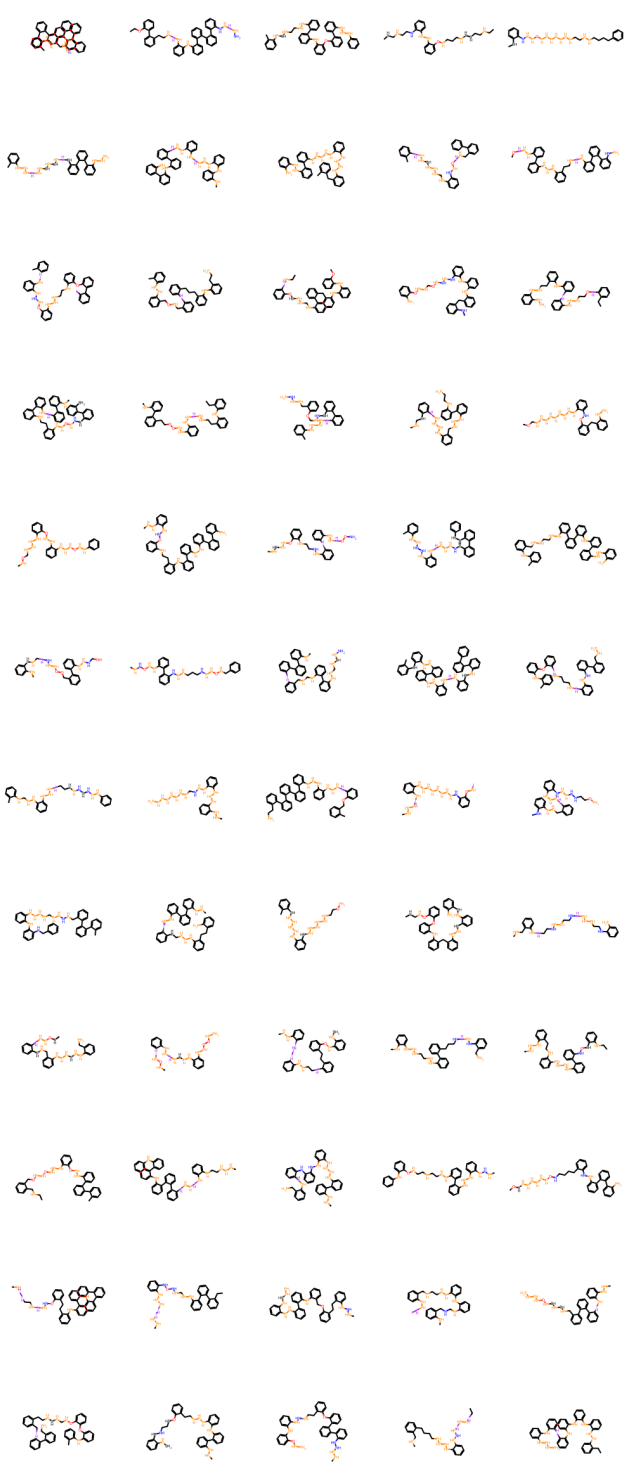

Figure 21: Conditionally generated molecules from GFlowNet samplers utilizing Soft MoE embeddings as their states.