Synergistic Benefits of Joint Molecule Generation and Property Prediction

Adam Izdebski *adam.izdebski@helmholtz-munich.de*
Institute of AI for Health, Helmholtz Zentrum Munchen
Technical University of Munich, TUM School of Computation, Information and Technology
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw

Jan Olszewski
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw

Pankhil Gawade
Institute of AI for Health, Helmholtz Zentrum Munchen

Krzysztof Koras
Ardigen SA

Serra Korkmaz
Institute of AI for Health, Helmholtz Zentrum Munchen

Valentin Rauscher
Technical University of Munich

Jakub M. Tomczak
Eindhoven University of Technology

Ewa Szczurek *ewa.szczurek@helmholtz-munich.de*
Institute of AI for Health, Helmholtz Zentrum Munchen
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw

Reviewed on OpenReview: *https://openreview.net/forum?id=jnzCOLyGOA*

Abstract

Modeling the joint distribution of data samples and their properties allows to construct a single model for both data generation and property prediction, with synergistic benefits reaching beyond purely generative or predictive models. However, training joint models presents daunting architectural and optimization challenges. Here, we propose Hyformer, a transformer-based joint model that successfully blends the generative and predictive functionalities, using an alternating attention mechanism and a joint pre-training scheme. We show that Hyformer is simultaneously optimized for molecule generation and property prediction, while exhibiting synergistic benefits in conditional sampling, out-of-distribution property prediction and representation learning. Finally, we demonstrate the benefits of joint learning in a drug design use case of discovering novel antimicrobial peptides.

Coda available at: `https://github.com/szczurek-lab/hyformer`

1 Introduction

Developing models that simultaneously excel in both generative and predictive tasks is a long-standing challenge in machine learning (Bishop, 1994; Jaakkola & Haussler, 1998; Lasserre et al., 2006). Joint models, which unify these tasks, offer synergistic benefits, including improved control over the generative process of the model, improved predictive robustness towards unseen, e.g., newly generated or out-of-distribution (OOD) data, and learning representations predictive of high-level molecular features (Nalisnick et al., 2019; Grathwohl et al., 2020; Cao & Zhang, 2022; Tomczak, 2022). These benefits are crucial for applications such as drug design, where success depends on balancing the generation of novel molecules from unexplored regions of the chemical space coupled with robust property prediction extrapolating towards the newly generated molecules (Grisoni, 2023; Steshin, 2023; van Tilborg et al., 2025).

However, molecule generation and property prediction are predominantly approached in separation. This division persists even though transformer-based models are state-of-the-art across both tasks (Bagal et al., 2022; Gao et al., 2024b; Irwin et al., 2022; Xia et al., 2023; Zhou et al., 2023). A likely reason is that joint training poses daunting challenges, as combining a generative and a predictive part into a single model may over-regularize both parts (Lasserre et al., 2006) or cause gradient interference between the generative and predictive objectives (Nalisnick et al., 2019). As a result, molecular models continue to forgo the potential benefits of joint learning. This raises a natural question, whether one can *develop a transformer-based joint model optimized for both generative and predictive performance, at the same time offering the synergistic benefits of joint learning?*

To address this challenge, we introduce Hyformer, a joint model that combines an autoregressive transformer decoder with a bidirectional transformer encoder in a single model with shared parameters. Upon training, we alternate between using the model as a decoder and as an encoder, with either a causal or bidirectional self-attention mechanism, alleviating problems typical for joint models. We evaluate the generative and predictive performance, as well as synergistic benefits of joint learning using Hyformer across a variety of molecular tasks (Wu et al., 2018; Brown et al., 2019; Steshin, 2023; Chen et al., 2023). Our contributions are:

1. We propose a novel joint model, Hyformer, that unifies the generative and the predictive task in a single set of parameters.

2. We demonstrate the synergistic benefits of joint modeling, where Hyformer outperforms baselines on (i) conditional molecule generation, (ii) out-of-distribution property prediction and (iii) molecular representation learning via probing.

3. We show that Hyformer rivals the generative and predictive performance of state-of-the-art purely generative and predictive models.

4. We showcase the applicability of joint modeling in a real-world drug design use case of discovering novel antimicrobial peptides.

2 Related Work

Molecule Generation Existing generative approaches can be categorized into sequence- and graph-based models. Sequence-based methods represent molecules as SMILES (Weininger, 1988) or SELFIES (Krenn et al., 2020) and process tokenized strings using recurrent or transformer-based language models (Segler et al., 2018; Flam-Shepherd et al., 2022; Bagal et al., 2022). In contrast, graph-based models treat molecules as graphs and have been implemented using variational autoencoders (Liu et al., 2018; Jin et al., 2019; Maziarz et al., 2022; Hetzel et al., 2023), normalizing flows (Luo et al., 2021), energy-based models (Liu et al., 2021a), and graph transformers (Gao et al., 2024b). More recently, 3D-based generative models have been proposed to capture the spatial geometry of molecules (Hoogeboom et al., 2022; Guan et al., 2023; Gao et al., 2024a), however real world drug discovery pipelines continue to rely predominantly on 2D-molecular representations (Xiang et al., 2024).

Molecular Property Prediction Analogously, prediction models leverage distinct molecular representations. Methods based on pre-trained language models predominantly work with SMILES (Wang et al., 2019; Fabian et al., 2020; Irwin et al., 2022; Sultan et al., 2024), while other approaches represent molecules as graphs (Li et al., 2021; Wang et al., 2022). Recent methods leverage the three-dimensional spatial structure of a molecule, either using graph neural networks (Fang et al., 2022) or transformers (Zhou et al., 2023). Finally, Yang et al. (2019); Fabian et al. (2020); Stokes et al. (2020) incorporate pre-computed physicochemical descriptors of molecules into training.

Joint Models for Molecules Early joint models combine variational autoencoders with latent-space predictors (Gómez-Bombarelli et al., 2018; Maziarz et al., 2022). Regression Transformer (Born & Manica, 2023) frames property prediction as conditional sequence generation, but lacks unconditional generative capability. Graph2Seq (Gao et al., 2024b) is a graph-based encoder-decoder transformer, trained separately as a generative or as a predictive model, but evaluated on both molecule generation and property prediction. UniGEM (Feng et al., 2024) is a diffusion-based model for unified generation and prediction, however specializing in 3D molecular modeling and not directly applicable to standard SMILES-based benchmarks.

Therefore, the question of whether the transformer architecture can be used to implement a joint model for both SMILES-based generation and prediction, while enjoying synergistic benefits, remains open.

3 Background

Problem Formulation The aim of *joint modeling* is to learn the joint distribution of the data and its properties $p(\mathbf{x}, y)$, i.e., to identify a model that at the same time generates new data and predicts its properties. We assume access to a *labeled dataset* $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, sampled from the joint data distribution $p(\mathbf{x}, y)$, often accompanied with an *unlabeled dataset* $\mathcal{D}_U = \{\mathbf{x}_n\}_{n=1}^{N_U}$, sampled from $p(\mathbf{x})$. Here, examples $\mathbf{x}$ can be thought of as molecules and labels y as molecular properties.

In the general formulation of Lasserre et al. (2006), joint modeling aims to learn the joint distribution $p(\mathbf{x}, y)$ by defining a *joint model* $p_{\theta,\phi}(\mathbf{x}, y)$ that factorizes into a *generative model* $p_\theta(\mathbf{x})$ and a *predictive model* $p_\phi(y \mid \mathbf{x})$ such that

$$p_{\theta,\phi}(\mathbf{x}, y) = p_\phi(y \mid \mathbf{x}) p_\theta(\mathbf{x}), \tag{1}$$

where θ denotes the parameters of the generative model, and ϕ the parameters of the predictive model. Training of the joint model is equivalent to minimizing the negative log-likelihood, i.e., the *joint loss*

$$\ell_\lambda(\theta, \phi) = -\mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\ln p_\theta(\mathbf{x}) + \lambda \ln p_\phi(y \mid \mathbf{x})], \tag{2}$$

where $\lambda \in \mathbb{R}$ weights the predictive and the generative parts.

Choosing the extent to which parameters θ and ϕ are shared and the way the joint loss is optimized, is crucial for obtaining a model with both a high generative and predictive performance, at the same time maintaining the synergistic benefits of joint learning (Lasserre et al., 2006).

3.1 Transformer-based Models

Transformers (Vaswani et al., 2017) achieve state-of-the-art performance in both molecule generation (Bagal et al., 2022) and property prediction (Zhou et al., 2023) tasks.

Transformer Encoders and Decoders Transformers used for generation and for property prediction differ in the use of the *self-attention* mechanism. Transformer decoders, used for generative tasks, employ a *causal self-attention*

$$Att_{\rightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\,\mathbf{K}^T}{\sqrt{d}} + \mathbf{M}_{\rightarrow}\right)\mathbf{V}, \tag{3}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{T\times d}$ are *query*, *key* and *value* matrices, respectively, $\mathbf{M}_{\rightarrow} \in \mathbb{R}^{T\times T}$ is a *causal mask*, i.e., a matrix such that $(\mathbf{M}_{\rightarrow})_{ij} = 0$, if $i \geq j$, and $(\mathbf{M}_{\rightarrow})_{ij} = -\infty$, otherwise, T is the sequence length and d is the

head dimension.[1] On the other hand, transformer encoders, used for predictive tasks, employ a *bidirectional self-attention*

$$Att_{\leftrightarrow}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\,\mathbf{K}^T}{\sqrt{d}} + \mathbf{M}_{\leftrightarrow}\right)\mathbf{V}, \tag{4}$$

where $\mathbf{M}_{\leftrightarrow} \in \mathbb{R}^{T\times T}$ is a *bidirectional mask*, i.e., $(\mathbf{M}_{\leftrightarrow})_{ij} = 0$ for all $i, j \in [T]$.

Alternating attention The definition of the transformer decoder and encoder can be generalized by using an alternating attention scheme (Dong et al., 2019):

$$Att_{\texttt{ATT_Type}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\,\mathbf{K}^T}{\sqrt{d}} + \mathbf{M}_{\texttt{ATT_Type}}\right)\mathbf{V}, \tag{5}$$

where $\texttt{ATT_Type} \in \{\rightarrow, \leftrightarrow\}$ and $\mathbf{M}_{\texttt{ATT_Type}} = \mathbf{M}_{\rightarrow}$ is a causal mask upon using the model as a transformer decoder and $\mathbf{M}_{\texttt{ATT_Type}} = \mathbf{M}_{\leftrightarrow}$, otherwise.

Training transformers Training transformers proceeds in a two-step manner, by first *pre-training* the model on an unlabeled dataset and then *fine-tuning* the pre-trained model on a downstream task. Transformer decoders and encoders are pre-trained using different losses.

Pre-training Transformer decoders, optimized for generative performance, are predominantly pre-trained using the negative log-likelihood loss $-\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\ln p_\theta(\mathbf{x})]$. As the causal mask induces a factorization of the transformer decoder into an autoregressive model $p_\theta(\mathbf{x}) = \prod_{t=1}^{T} p_\theta(x_t \mid \mathbf{x}_{<t})$, where $\mathbf{x} = (x_1, \ldots, x_T)$, the generative loss reduces to the *language modeling* (LM) loss

$$\ell_{\text{LM}}(\theta) = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\sum_{t=1}^{T} \ln p_\theta(x_t \mid \mathbf{x}_{<t})\right]. \tag{6}$$

On the other hand, transformer encoders are usually pre-trained using *masked language modeling* (MLM) loss

$$\ell_{\text{MLM}}(\theta) = -\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\mathbb{E}_{\mathcal{M}}\Big[\ln p_\theta(\mathbf{x}_{\mathcal{M}} \mid \mathbf{x}_{\mathcal{R}})\Big], \tag{7}$$

where $\mathbf{x} = (x_1, \ldots, x_T)$, $\mathcal{M}$ is a set of indices drawn uniformly at random from the set of token indices $\{1, \ldots, T\}$ and the set of all tokens whose indices belongs to $\mathcal{M}$ are *masked tokens* $\mathbf{x}_{\mathcal{M}}$. The rest of the tokens $\mathbf{x}_{\mathcal{R}}$ are defined such that $\mathbf{x} = \mathbf{x}_{\mathcal{M}} \cup \mathbf{x}_{\mathcal{R}}$.

Fine-tuning Next, the pretrained model is fine-tuned by defining a predictive head on top of the pretrained model and training it as a predictor on a labeled dataset using the *prediction loss*

$$\ell_{\text{PRED}}(\phi) = -\mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\ln p_\phi(y \mid \mathbf{x})]. \tag{8}$$

4 Hyformer

We propose Hyformer, a joint transformer-based model that unifies a generative decoder with a predictive encoder in a single set of shared parameters, using an alternating training scheme.

4.1 Model Formulation

Hyformer unifies a decoder with an encoder using a transformer backbone $f_\theta(\mathbf{x}; \texttt{[TASK]})$ conditioned on a *task token* $\texttt{[TASK]} \in \{\texttt{[LM]}, \texttt{[PRED]}, \texttt{[MLM]}\}$. The task token facilitates switching between respective losses during

[1] We assume that the dimensions of the query, key, and value matrices are equal.

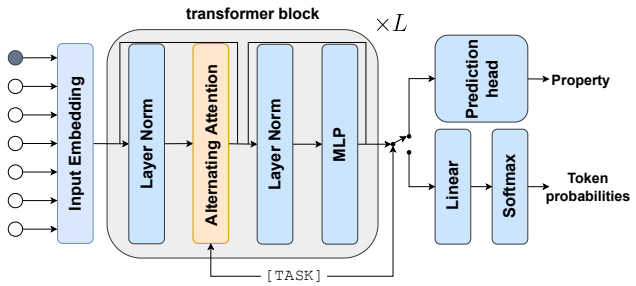

Figure 1: A schematic representation of HYFORMER. Depending on the task token [TASK], HYFORMER uses either a causal or a bidirectional mask, outputting token probabilities or predicted property values.

training (see Section 4.2) and determines whether the backbone f_θ processes input $\mathbf{x}$ in an autoregressive manner using a causal, or a bidirectional mask

$$\texttt{ATT_Type} = \begin{cases} \rightarrow & \text{if } \texttt{[TASK]} = \texttt{[LM]}, \\ \leftrightarrow & \text{if } \texttt{[TASK]} \in \{\texttt{[PRED]}, \texttt{[MLM]}\}. \end{cases}$$

Finally, the generative $p_\theta(\mathbf{x})$ and predictive $p_\theta(y \mid \mathbf{x})$ parts of the joint model, factorized as

$$p_\theta(\mathbf{x}, y) := p_\theta(\mathbf{x}) p_\theta(y \mid \mathbf{x}), \tag{9}$$

are implemented by adding a generative and a predictive head on the top of the shared backbone f_θ.

Algorithm 1 Training of HYFORMER

Input: Dataset $\mathcal{D}$ (labeled or unlabeled); model parameters θ; task probabilities $\mathbf{p}_{\texttt{[TASK]}}$.
For pre-training: [TASK] $\in \{$[LM], [PRED], [MLM]$\}$, for fine-tuning: [TASK] $\in \{$[LM], [PRED]$\}$.

1: **while** stopping criterion not met **do**
2: Sample task [TASK] $\sim$ CAT($\mathbf{p}_{\texttt{[TASK]}}$)
3: Select loss $\ell_{\texttt{[TASK]}}$ and the corresponding attention mask
4: Update model parameters θ using the gradient of $\ell_{\texttt{[TASK]}}$
5: **end while**

4.2 Hyformer Training

As with standard transformer-based models, the training of HYFORMER is divided into a pre-training and a fine-tuning stage.

Joint Pre-training To unify the generative and the predictive functionalities in a single model, we pre-train HYFORMER using a variant of the joint loss (Eq. 2). For the generative part, we use the language modeling loss ℓ_{LM}, while for the predictive part, we use the masked language modeling loss ℓ_{MLM} and the predictive loss ℓ_{PRED}, with the combined loss being defined as:

$$\ell_{\text{HYFORMER}} = \ell_{\text{LM}} + \mu \ell_{\text{MLM}} + \eta \ell_{\text{PRED}}. \tag{10}$$

As *pre-training labels*, we use values analytically computable from the input sequences, e.g., molecular descriptors, such as molecular weight for small molecules, or hydrophobicity for peptides. When the pre-training labels are not available, HYFORMER is pre-trained without the predictive loss ℓ_{PRED}. Analogously to multitask learning (Raffel et al., 2023), the weighted loss ℓ_{HYFORMER} (Eq. 10) is effectively implemented using a vector of task probabilities $\mathbf{p}_{\texttt{[TASK]}} = (p_{\texttt{[LM]}}, p_{\texttt{[MLM]}}, p_{\texttt{[PRED]}})$, which defines how the generative and predictive capabilities of the joint model are balanced.

During training, the shared parameters θ are updated differently depending on the task token. If $\texttt{[TASK]} \in \{\texttt{[PRED]}, \texttt{[MLM]}\}$, a bidirectional attention mask $\mathbf{M}_{\leftrightarrow}$ is applied and all attention module weights are updated, since the bidirectional mask does not restrict information flow. Conversely, if $\texttt{[TASK]} = \texttt{[LM]}$, a causal mask $\mathbf{M}_{\rightarrow}$ is applied, restricting each token to attend only to its left context, altering the gradients of the attention module, due to the functional form of the Jacobian of the softmax function, alleviating gradient interference typical for joint modeling (Appendix D.1).

Fine-tuning We fine-tune Hyformer using the joint loss (Eq. 2), defined as

$$\ell_{\text{Hyformer}} = \ell_{\text{LM}} + \lambda \ell_{\text{PRED}}. \tag{11}$$

Analogously to pre-training, Hyformer alternates between the generative and predictive task, to balance their objectives, based on a pre-defined vector of task probabilities $\mathbf{p}_{\texttt{[TASK]}} = (p_{\texttt{[LM]}}, p_{\text{[PRED]}})$. We assume that *fine-tuning labels* used in loss ℓ_{PRED} are different than in the pre-training phase and are defined by the downstream prediction task. Specifically, we omit the masked language modeling loss, to focus on the downstream task while retaining the generative capabilities of the model.

4.3 Sampling

Sampling from Hyformer exploits the generative $p_\theta(\mathbf{x})$ and predictive part $p_\theta(y \mid \mathbf{x})$ depending on the sampling mode: unconditional or conditional.

Unconditional Generation In unconditional generation, we sample $\mathbf{x} \sim p_\theta(\mathbf{x})$ using the autoregressive part of the model. This addresses a limitation of conditionally trained generative models (Bagal et al., 2022) and joint models trained without a pure unsupervised objective (Born & Manica, 2023), where generating a single example requires conditioning on a fixed property value inferred from a dataset.

Conditional Generation To generate $(\mathbf{x}, y) \sim p_\theta(\mathbf{x}, y)$ that satisfies a condition $Y \subseteq \mathcal{Y}$, Hyformer samples K-many examples $\mathbf{x}_1, \ldots, \mathbf{x}_K \sim p_\theta(\mathbf{x})$ and, for every $k = 1, \ldots, K$, accepts sample $\mathbf{x}_k$, if the predictor $p_\theta(y \mid \mathbf{x})$ classifies $\mathbf{x}_k$ as having property Y. As a simple consequence of the Bayes rule, the above procedure yields a correct conditional sampling procedure, as

$$p(\mathbf{x} \mid y \in Y) \propto p(y \in Y \mid \mathbf{x}) p(\mathbf{x}), \tag{12}$$

for $y \in Y \subseteq \mathcal{Y}$ such that $p(y \in Y) > 0$. Note that the conditional sampling procedure of Hyformer is a variant of best-of-K sampling, a provably near-optimal solution to the KL-regularized RL problem (Yang et al., 2019) that has been shown to outperform other conditional sampling methods for LLMs, including state-of-the-art reinforcement learning methods like PPO and DPO (Touvron et al., 2023; Mudgal et al., 2023; Gao et al., 2023; Rafailov et al., 2023). Crucially, Hyformer leverages a jointly trained predictor $p_\theta(y \mid x)$ over a unified representation space, resulting in tighter alignment between generation and control. This coherence is particularly valuable in drug discovery, where the primary objective is not throughput, but *precision and sample efficiency*, that is, generating a small number of high-quality candidates with minimal false positives.

5 Experiments

We evaluate Hyformer across a broad range of molecular modeling tasks. First, we demonstrate the synergistic benefits of joint modeling in three settings: (i) conditional generation on GuacaMol dataset (Brown et al., 2019), (ii) out-of-distribution (OOD) property prediction on Hit Identification task from the Lo-Hi benchmark (Steshin, 2023) and (iii) representation learning via probing on MoleculeNet benchmark (Wu et al., 2018). Subsequently, we show that Hyformer rivals state-of-the-art generative and predictive models in both unconditional generation on GuacaMol and property prediction on MoleculeNet. Finally, we apply Hyformer to antimicrobial peptide (AMP) design, showcasing the benefits of our joint modeling approach. Experimental details and additional results are provided in Appendix E, F and G.

Table 1: Conditional generative performance on GuacaMol dataset. Best model is marked **bold**.

Model	Joint	Metric	QED	SA	logP	Avg.
MolGPT	✗	MAD ↓	0.087	0.019	0.276	0.127
		SD ↓	0.074	0.017	0.262	0.118
		Validity ↑	0.985	0.986	0.982	0.984
GraphGPT	✗	MAD ↓	0.039	0.011	0.158	0.069
		SD ↓	0.082	0.047	0.653	0.261
		Validity ↑	**0.998**	**0.997**	**0.992**	**0.995**
Hyformer	✗	MAD ↓	0.031 (0.003)	0.015 (0.001)	0.131 (0.010)	0.059 (0.004)
		SD ↓	0.045 (0.004)	0.020 (0.001)	0.170 (0.014)	0.078 (0.006)
		Validity ↑	0.993 (0.003)	0.990 (0.004)	0.985 (0.014)	0.989 (0.007)
	✓	MAD ↓	**0.008 (0.001)**	**0.005 (0.000)**	**0.041 (0.002)**	**0.018 (0.001)**
		SD ↓	**0.015 (0.002)**	**0.009 (0.002)**	**0.051 (0.004)**	**0.025 (0.003)**
		Validity ↑	0.990 (0.007)	0.985 (0.003)	0.987 (0.006)	0.987 (0.005)

5.1 Synergistic Benefits of Hyformer

5.1.1 Conditional Molecule Generation

To demonstrate the synergistic benefits of Hyformer in generating molecules with specific molecular properties, we follow the setup of Bagal et al. (2022) and jointly pre-train Hyformer scaled to 8.5M parameters on GuacaMol dataset with 1.3M molecules, using pre-computed molecular descriptors (Yang et al., 2019). We subsequently jointly fine-tune Hyformer on GuacaMol dataset with QED, SA, and LogP molecular properties, as fine-tuning labels, and generate molecules with specific properties using Hyformer's conditional sampling procedure. Pre-training and experimental details alongside results for all property settings can be found in Appendix E and F.1.

Following (Gao et al., 2024b), we compare Hyformer to MolGPT (Bagal et al., 2022) and GraphGPT (Gao et al., 2024b) using: mean absolute deviation (MAD) from the target property value, standard deviation (SD) of the generated property values and validity of the generated molecules. Evaluation is averaged across three target values per each property: QED:{0.5, 0.7, 0.9}, SA:{0.7, 0.8, 0.9}, and logP:{0.0, 2.0, 4.0}. Additionally, we compare to a non-joint variant of Hyformer, in which the predictive head is fine-tuned with prediction loss, on top of a frozen, pre-trained generative part, i.e., without joint fine-tuning.

The jointly fine-tuned Hyformer achieves the lowest MAD and SD across all properties, while maintaining high validity, outperforming all baselines. Notably, Hyformer improves controllability over it's non-joint counterpart, confirming that joint fine-tuning enhances conditional generation. Although GraphGPT attains slightly higher validity, it does so at the cost of reduced controllability. These results demonstrate that joint modeling enables robust property-conditioned molecular generation across a range of chemically relevant targets.

5.1.2 Out-of-Distribution Molecular Property Prediction

To evaluate the ability of Hyformer to predict molecular properties in an out-of-distribution (OOD) setting, we jointly pre-train Hyformer scaled to 50M parameters on 19M molecules from (Zhou et al., 2023), together with pre-computed molecular descriptors (Yang et al., 2019), and benchmark on the Hit Identification (Hi) task from the Lo-Hi benchmark (Steshin, 2023). The Hi task requires generalization to molecular scaffolds not seen during training, with the test set constructed such that no molecule has a Tanimoto similarity greater than 0.4 (based on ECFP4 fingerprints) to any molecule in the training set. This setup mimics realistic drug discovery scenarios, where generalization beyond known chemical space is essential. For experimental details, see Appendix E and F.2.

We follow the setup of (Steshin, 2023) and compare jointly fine-tuned Hyformer to all models reported in (Steshin, 2023); machine learning models: k-NN, gradient boosting (GB), SVM and MLP, trained on molecular fingerprints (ECFP4, MACCS) and deep learning models: Chemformer (Yang et al., 2019) and Graphformer (Ying et al., 2021; Shi et al., 2022). Moreover, we compare to Hyformer (no-joint), which is a version of our model pre-trained using MLM loss, hence without alternating attention, and fine-tuned using the prediction loss only.

Table 2: Predictive performance (AUPRC) on Hit Identification (Hi) task from Lo-Hi benchmark. Mean and standard deviation across 3 random seeds.

	Dataset, AUPRC (↑)			
Model	DRD2-Hi	HIV-Hi	KDR-Hi	Sol-Hi
Dummy baseline	0.677±0.061	0.040±0.014	0.609±0.081	0.215±0.008
KNN (ECFP4)	0.706±0.047	0.067±0.029	0.646±0.048	0.426±0.022
KNN (MACCS)	0.702±0.042	0.072±0.036	0.610±0.072	0.422±0.009
GB (ECFP4)	0.736±0.050	0.080±0.038	0.607±0.067	0.429±0.006
GB (MACCS)	0.751±0.063	0.058±0.030	0.603±0.074	0.502±0.045
SVM (ECFP4)	0.677±0.061	0.040±0.014	0.611±0.081	0.298±0.047
SVM (MACCS)	0.713±0.050	0.042±0.015	0.605±0.082	0.308±0.021
MLP (ECFP4)	0.717±0.063	0.049±0.019	0.626±0.047	0.403±0.017
MLP (MACCS)	0.696±0.048	0.052±0.018	0.613±0.077	0.462±0.048
Chemprop	0.782±0.062	0.148±0.114	0.676±0.026	0.618±0.030
Graphormer	0.729±0.039	0.096±0.070	-	-
Hyformer (no-joint)	0.778±0.070	0.154±0.108	0.675±0.046	0.601±0.040
Hyformer	0.784±0.082	0.158±0.128	0.701±0.022	0.640±0.036

Hyformer achieves the highest mean AUPRC across all datasets (Table 2), outperforming fingerprint-based baselines and demonstrating the potential of deep learning methods in real-world drug discovery. The consistent ranking in favor of Hyformer shows the benefits of joint modeling in out-of-distribution molecular property prediction, although the differences are not statistically significant at the 95% confidence level.

5.1.3 Molecular Representation Learning

Table 3: Molecular representation learning performance of predictive, generative and joint models on MoleculeNet benchmark, evaluated using linear and KNN probing. Best model within each probing method is marked **bold**.

			Dataset, RMSE ↓			Dataset, AUCROC ↑						
	Type	Model	Esol	Freesolv	Lipo	BBBP	BACE	ClinTox	Tox21	ToxCast	SIDER	HIV
Linear	P.	Uni-Mol	1.350	**2.503**	1.002	65.5	66.3	74.3	70.1	59.9	58.1	73.6
	P.	Hyformer (no-joint)	1.256	2.640	0.894	68.4	73.6	98.8	**73.4**	**61.2**	58.8	75.9
	G.	MolGPT	1.299	4.110	1.033	66.8	79.1	97.8	71.9	60.5	59.2	**77.5**
	J.	MoLeR	**1.223**	4.935	0.938	67.8	**79.5**	84.6	71.1	59.3	58.3	74.6
	J.	RT	2.510	4.515	1.158	54.7	63.1	57.3	50.5	52.8	54.5	65.6
	J.	Graph2Seq	1.498	3.486	0.890	66.0	76.7	72.0	71.2	60.4	50.5	57.1
	J.	Hyformer	1.527	4.294	**0.887**	**68.5**	77.2	**99.5**	72.4	60.7	**60.8**	74.7
KNN	P.	Uni-Mol	1.579	3.403	1.025	60.0	75.9	78.0	64.7	57.5	61.0	64.3
	P.	Hyformer (no-joint)	1.380	3.254	0.978	67.8	75.4	89.0	66.3	57.6	58.1	71.4
	G.	MolGPT	**1.232**	**3.075**	0.987	68.4	71.9	**94.2**	66.0	56.9	61.0	70.5
	J.	MoLeR	1.802	4.061	1.096	59.4	72.0	71.2	64.9	53.3	57.3	67.3
	J.	RT	2.411	4.734	1.242	59.3	56.1	59.4	50.8	52.2	51.2	54.1
	J.	Graph2Seq	1.361	3.796	0.967	**71.0**	**80.6**	56.3	67.7	57.8	49.9	52.4
	J.	Hyformer	1.260	3.999	**0.902**	69.5	78.4	93.8	**71.2**	**59.3**	**64.1**	**71.8**

To assess the quality of molecular representations learned by Hyformer, we introduce a novel probing protocol that emulates a typical drug discovery setting, where fixed molecular embeddings are used as inputs to downstream predictive models. In this setup, we train simple linear models with L2 regularization, and k-nearest neighbor (KNN) predictors on the top of frozen embeddings extracted from the respective pre-trained models. To ensure comparability with MoleculeNet benchmark (Section 5.2.2), we reuse the same datasets, data splits, and model checkpoints. Implementation details are provided in Appendix F.3.

We compare representations extracted from jointly pre-trained Hyformer to those extracted from a range of baselines, including state-of-the-art generative (MolGPT (Bagal et al., 2022)), predictive (Uni-Mol (Zhou et al., 2023)), and joint models: MoLeR (Maziarz et al., 2022), Regression Transformer (RT) (Born & Manica, 2023) and Graph2Seq (Gao et al., 2024b). Moreover, to quantify the effect of alternating attention and joint pre-training, we compare to Hyformer (no-joint), the version of our model trained solely with MLM loss.

The jointly pre-trained representations from Hyformer are the most predictive across both KNN and linear probings, achieving the best performance on 4 out of 10 datasets for linear, and 5 out of 10 datasets for KNN, outperforming all other baselines (Table 3). The next best models, Hyformer (no-joint) and MoLeR for linear

Table 4: Unconditional generative performance on GuacaMol distribution learning benchmarks. The best model in each category is marked **bold**.

Model	FCD ↑	KL div. ↑	Val. ↑	Uniq. ↑	Nov. ↑
Graph-based					
JT-VAE	0.750	0.940	**1.000**	-	-
MoLeR	0.625	0.964	**1.000**	**1.000**	**0.991**
MAGNet	**0.760**	0.950	**1.000**	-	-
MiCaM	0.731	**0.989**	**1.000**	0.994	0.986
SMILES-based					
VAE	0.863	0.982	0.870	0.999	0.974
LSTM	0.913	0.991	0.959	**1.000**	0.912
MolGPT	0.907	0.992	0.981	0.998	**1.000**
Hyformer$_{\tau=0.9}$	0.897 (0.002)	**0.995 (0.000)**	**0.986 (0.001)**	0.999 (0.000)	0.879 (0.006)
Hyformer$_{\tau=1.0}$	**0.918 (0.002)**	0.989 (0.001)	0.978 (0.000)	0.999 (0.000)	0.908 (0.002)
Hyformer$_{\tau=1.1}$	0.894 (0.002)	0.977 (0.001)	0.965 (0.001)	**1.000 (0.000)**	0.931 (0.001)

and MolGPT for KNN probing, rank first on 2 and 3 out of 10 datasets, respectively. Notably, joint models outperform UniMol, the state-of-the-art property predictor, on all datasets, except for Freesolv with linear probing, highlighting the effectiveness of joint modeling for transferable molecular representation learning.

5.2 Generative and predictive performance of Hyformer

We next confirm that Hyformer effectively addresses the challenges of joint training, while it enjoys the synergistic benefits described above, it does not sacrifice generative or predictive performance compared to state-of-the-art models trained separately for these tasks.

5.2.1 Unconditional Molecule Generation

To evaluate the unconditional generative performance of Hyformer, we perform an evaluation on the Guacamol distribution learning benchmark (Brown et al., 2019). We use Hyformer scaled to 8.5M parameters and trained on GuacaMol dataset with 1.3M molecules, together with pre-computed molecular descriptors (Yang et al., 2019), and investigate the impact of sampling temperature τ. For experimental details, see Appendix F.4.

We compare to state-of-the-art unconditional generative models; SMILES-based: VAE (Kingma & Welling, 2013), LSTM (Gers & Schmidhuber, 2001), MolGPT (Bagal et al., 2022) and graph-based: JT-VAE (Jin et al., 2019), MoLeR (Maziarz et al., 2022), MAGNet (Hetzel et al., 2023), MiCaM (Geng et al., 2023). We omit RT (Born & Manica, 2023) and GraphGPT (Gao et al., 2024b) as they do not generate molecules unconditionally or provide results on the GuacaMol benchmark.

Hyformer, with top FCD and KL div. values, outperforms graph-based models, while achieving the highest validity among SMILES-based models. Across various sampling temperatures τ, Hyformer consistently lies on the Pareto front, balancing distributional fidelity (FCD, KL div.), validity and uniqueness. Overall, SMILES-based models outperform those based on theoretically more informative graph representations in terms of FCD, at the expense of not always sampling valid molecules.

5.2.2 Molecular Property Prediction

To evaluate the predictive performance of Hyformer, we use Hyformer scaled to 50M parameters on 19M molecules from (Zhou et al., 2023), together with pre-computed molecular descriptors (Yang et al., 2019), and fine-tune end-to-end on MoleculeNet benchmark (Wu et al., 2018). For experimental details, see Appendix F.5.

We follow the experimental protocol of (Zhou et al., 2023), use scaffold splitting and compare to predictive models: D-MPNN (Yang et al., 2019), AttentiveFP (Xiong et al., 2019), N-gram (Liu et al., 2019) with Random Forest and XGBoost (Chen & Guestrin, 2016), PretrainGNN (Hu et al., 2019), GROVER (Rong et al., 2020), MolCLR (Wang et al., 2022), Mole-BERT (Xia et al., 2023), GraphMVP (Liu et al., 2021b), GEM (Fang et al., 2022), UniMol (Zhou et al., 2023) and a joint model: Graph2Seq (Gao et al., 2024b). We omit RT (Born & Manica, 2023) and other models that use random splitting.

Table 5: Predictive performance of predictive and joint models on the MoleculeNet benchmark. Mean and standard deviation across 3 random seeds. The best model in each category, statistically significant at the 95% confidence level, is marked **bold**.

		Dataset, RMSE ↓			Dataset, AUCROC ↑						
	Model	Esol	Freesolv	Lipo	BBBP	BACE	ClinTox	Tox21	ToxCast	SIDER	HIV
Predictive	D-MPNN	1.050(0.008)	2.082(0.082)	0.683(0.016)	71.0(0.3)	80.9(0.6)	90.6(0.6)	75.9(0.7)	65.5(0.3)	57.0(0.7)	77.1(0.5)
	Attentive FP	0.877(0.029)	2.073(0.183)	0.721(0.001)	64.3(1.8)	78.4(0.02)	84.7(0.3)	76.1(0.5)	63.7(0.2)	60.6(3.2)	75.7(1.4)
	N-GramRF	1.074(0.107)	2.688(0.085)	0.812(0.028)	69.7(0.6)	77.9(1.5)	77.5(4.0)	74.3(0.4)	-	66.8(0.7)	77.2(0.1)
	N-GramXGB	1.083(0.082)	5.061(0.744)	2.072(0.030)	69.1(0.8)	79.1(1.3)	87.5(2.7)	75.8(0.9)	-	65.5(0.7)	78.7(0.4)
	PretrainGNN	1.100(0.006)	2.764(0.002)	0.739(0.003)	68.7(1.3)	84.5(0.7)	72.6(1.5)	78.1(0.6)	65.7(0.6)	62.7(0.8)	79.9(0.7)
	GROVERbase	0.983(0.090)	2.176(0.052)	0.817(0.008)	70.0(0.1)	82.6(0.7)	81.2(3.0)	74.3(0.1)	65.4(0.4)	64.8(0.6)	62.5(0.9)
	GROVERlarge	0.895(0.017)	2.272(0.051)	0.823(0.010)	69.5(0.1)	81.0(1.4)	76.2(3.7)	73.5(0.1)	65.3(0.5)	65.4(0.1)	68.2(1.1)
	GraphMVP	1.029(0.033)	-	0.681(0.010)	72.4(1.6)	81.2(0.9)	79.1(2.8)	75.9(0.5)	63.1(0.4)	63.9(1.2)	77.0(1.2)
	MolCLR	1.271(0.040)	2.594(0.249)	0.691(0.004)	72.2(2.1)	82.4(0.9)	91.2(3.5)	75.0(0.2)	-	58.9(1.4)	78.1(0.5)
	Mole-BERT	1.015 (0.030)	-	0.676 (0.017)	71.9 (1.6)	80.8 (1.4)	78.9 (3.0)	76.8 (0.5)	64.3 (0.2)	-	-
	GEM	0.798(0.029)	1.877(0.094)	0.660(0.008)	72.4(0.4)	85.6(1.1)	90.1(1.3)	78.1(0.1)	69.2(0.4)	67.2(0.4)	80.6(0.9)
	Uni-Mol	0.788(0.029)	**1.480(0.048)**	**0.603(0.010)**	72.9(0.6)	85.7(0.2)	91.9(1.8)	**79.6(0.5)**	69.6(0.1)	65.9(1.3)	80.8(0.3)
Joint	Graph2Seq	0.860(0.024)	1.797(0.237)	0.716(0.019)	72.8(1.5)	83.4(1.0)	-	76.9(0.3)	65.4(0.5)	68.2(0.9)	79.4(3.9)
	Hyformer	**0.774(0.026)**	2.047(0.076)	**0.643(0.002)**	75.9(0.9)	83.8(1.1)	**99.2(0.5)**	**79.2(0.1)**	65.5(0.6)	65.7(1.6)	80.0(1.0)

Hyformer obtains the lowest RMSE on Esol, and highest AUROC on BBBP and ClinTox, outperforming all models on 3 out of 10 datasets (Table 5). Moreover, Hyformer performs better than Graph2Seq, the only other joint model capable of simultaneous molecule generation and property prediction, on 8 out of 10 datasets. Altogether, Hyformer outperforms the other joint learning model, Graph2Seq, and successfully rivals the performance of purely predictive models, demonstrating the efficiency of our joint learning strategy.

Table 6: Conditional generative performance on antimicrobial peptide design. The best model is marked **bold**.

Model	Perplexity[2]	Diversity ↑	Fitness ↑	$\text{HydrAMP}_{\text{MIC}}$ ↑	AMPlify ↑	amPEPpy ↑
PepCVAE	20.08	**0.86**	0.07	0.20	0.49	0.52
AMPGAN	18.49	0.80	0.12	0.32	0.64	0.54
HydrAMP	20.14	**0.86**	0.09	0.49	0.59	0.52
AMP-Diffusion	16.84	0.82	0.12	0.26	0.20	0.38
Hyformer	17.24	0.80	**0.19**	**0.80**	**0.94**	**0.72**

5.3 Antimicrobial Peptide Design

To show the benefits of joint learning in a real-world use case related to drug discovery, we apply Hyformer to the task of antimicrobial peptide (AMP) design (Chen et al., 2023), i.e., generating AMPs with low minimal inhibitory concentration values (MIC) against *E. coli* bacteria. We jointly pre-train Hyformer on 3.5M general-purpose peptide sequences, and subsequently on 1M AMP sequences, together with 39 physicochemical descriptors from *peptidy* package (Özçelik et al., 2025). Next, we jointly fine-tune Hyformer on 4,547 peptides with their MIC values (Szymczak et al., 2023) and conditionally sample 50K peptides with an MIC regressor threshold set to $\leq 10^{0.3} \approx 2\,\mu\text{M}$. For experimental details, see Appendix F.6.

We compare Hyformer AMP generation baselines: PepCVAE (Das et al., 2018), AMPGAN (Van Oort et al., 2021), HydrAMP (Szymczak et al., 2023), and AMP-Diffusion (Chen et al., 2023). Evaluation is based on four criteria: Perplexity (Torres et al., 2025), Diversity and Fitness (Li et al., 2024), and success rates in generating AMPs and low-MIC candidates. For the latter, we use $\text{HydrAMP}_{\text{MIC}}$, Amplify (Li et al., 2022), and amPEPpy (Lawrence et al., 2020) classifiers as state-of-the-art *in-silico* oracles.

Hyformer outperforms all baseline models by a large margin in terms of generating peptides with a high fitness and AMP probability, as evaluated by all oracle classifiers (Table 6). Despite the stringent conditioning MIC threshold of 2 µM, Hyformer maintains competitive perplexity and high diversity. These results suggest that even when constrained to explore less charted regions of sequence space, Hyformer is able to generate biologically plausible and novel peptide candidates.

To further validate the biological relevance of the generated peptides, we show that both unconditional sampling from jointly pre-trained Hyformer, and conditional sampling from the fine-tuned model produces amino-acid distributions in close agreement with the training data (Figure 2a). Despite this very close

[2]We report perplexity, but do not seek to minimize it, as it inherently balances plausibility and novelty.

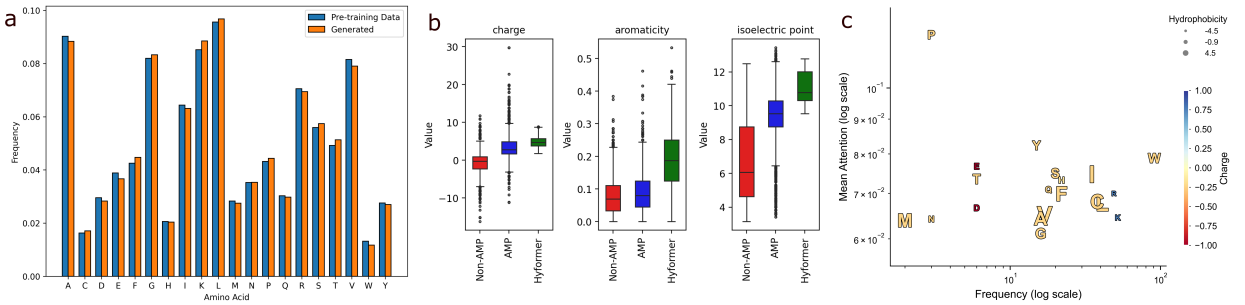

Figure 2: **(a)** Amino-acid distributions between the pre-training data and unconditionally generated sequences. **(b)** Distributions of charge, aromaticity, and isoelectric point (pI) for: non-AMP, AMP and conditionally generated sequences. **(c)** Frequency of crossing an attention threshold (x-axis) vs. mean attention weight (y-axis) for distinct amino-acids, colored by charge and sized by hydrophobicity.

agreement, the conditionally sampled peptides obtain a significant improvement of charge, aromaticity, and isoelectric point over the known non-AMPs, as compared to known AMPs (Fig. 2b). Finally, to gain insight into which amino acids contribute most to antimicrobial activity, we analyze the attention weights of Hyformer (Fig. 2c). The attention mechanism frequently prioritizes highly charged Arginine (R) and Lysine (K), which is expected as high AMP activity is associated with increased charge. The high attention frequency on Tryptophan (W) agrees with previous reports about this amino-acid's unique ability to interact with the interface of the bacterial membrane (Bi et al., 2014). Finally, the high attention that Hyformer puts on Proline (P) agrees with the known high potency of Proline-rich AMPs, which kill bacteria via a specific, non-lytic mechanism (Lai et al., 2019).

6 Discussion

In this paper, we introduced Hyformer, a transformer-based joint model that combines an autoregressive decoder and a bidirectional encoder within a single set of shared parameters, using an alternating attention mechanism and joint pre-training. We showed that Hyformer provides synergistic benefits in conditional sampling, representation learning and out-of-distribution property prediction, with ablations highlighting the specific contributions of alternating attention and joint training. Furthermore, we validated the utility of joint modeling in a real-world antimicrobial peptide design task. Our results indicate that Hyformer successfully unifies molecular generation and property prediction for SMILES-based molecular representations, opening the avenue for the integration into real-world drug discovery pipelines, where informative molecular representations, robustness to OOD examples and robust conditional sampling are crucial.

Limitations & Future Work However, joint modeling introduces an inherent trade-off. While shared parameters promote synergistic benefits and learning unified representations, they may limit task-specific specialization. Therefore, a promising direction for future work is designing dynamic or modular attention architectures that allocate capacity across tasks more flexibly, while preserving synergistic benefits. Moreover, to ensure fair comparison with prior work and isolate the effect of joint learning, we deliberately restricted model scale and relied on a fixed set of analytically computed descriptors. The extent to which the observed synergistic benefits carry over to other modalities, such as 3D structures, morphology or transcriptomics, remains an open question.

References

Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized smiles strings improve the quality of molecular generative models. *Journal of cheminformatics*, 11:1–13, 2019.

Viraj Bagal, Rishal Aggarwal, P. K. Vinod, and U. Deva Priyakumar. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, May 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00600.

Xiaonan Bi, Che Wang, Weibing Dong, Wei Zhu, and Dejing Shang. Antimicrobial properties and interaction of two trp-substituted cationic antimicrobial peptides with a lipid bilayer. *The Journal of Antibiotics*, 67 (5):361–368, 2014.

Christopher M Bishop. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994.

Esben Jannik Bjerrum. Smiles enumeration as data augmentation for neural network modeling of molecules, 2017.

Jannis Born and Matteo Manica. Regression transformer enables concurrent sequence regression and generation for molecular language modelling. *Nature Machine Intelligence*, 5(4):432–444, 2023.

Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.8b00839.

Gabriel Cabas-Mora, Anamaría Daza, Nicole Soto-García, Valentina Garrido, Diego Alvarez, Marcelo Navarrete, Lindybeth Sarmiento-Varón, Julieta H. Sepúlveda Yañez, Mehdi D. Davari, Frederic Cadet, Álvaro Olivera-Nappa, Roberto Uribe-Paredes, and David Medina-Ortiz. Peptipedia v2.0: A peptide sequence database and user-friendly web platform. a major update. *bioRxiv*, 2024. doi: 10.1101/2024.07.11.603053. URL `https://www.biorxiv.org/content/early/2024/07/16/2024.07.11.603053`.

Senqi Cao and Zhongfei Zhang. Deep hybrid models for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4733–4743, 2022.

Tianlai Chen, Pranay Vure, Rishab Pulugurta, and Pranam Chatterjee. AMP-diffusion: Integrating latent diffusion with protein language models for antimicrobial peptide generation. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 11 2024. ISSN 1362-4962. doi: 10.1093/nar/gkae1010. URL `https://doi.org/10.1093/nar/gkae1010`.

Payel Das, Kahini Wadhawan, Oscar Chang, Tom Sercu, Cicero Dos Santos, Matthew Riemer, Vijil Chenthamarakshan, Inkit Padhi, and Aleksandra Mojsilovic. Pepcvae: Semi-supervised targeted design of antimicrobial peptide sequences, 2018.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.

Shikun Feng, Yuyan Ni, Yan Lu, Zhi-Ming Ma, Wei-Ying Ma, and Yanyan Lan. Unigem: A unified approach to generation and property prediction for molecules. *arXiv preprint arXiv:2410.10516*, 2024.

Daniel Flam-Shepherd, Kevin Zhu, and Alán Aspuru-Guzik. Language models can learn complex molecular distributions. *Nature Communications*, 13(1):3293, 2022.

Bowen Gao, Minsi Ren, Yuyan Ni, Yanwen Huang, Bo Qiang, Zhi-Ming Ma, Wei-Ying Ma, and Yanyan Lan. Rethinking specificity in sbdd: Leveraging delta score and energy-guided diffusion. *arXiv preprint arXiv:2403.12987*, 2024a.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Zhangyang Gao, Daize Dong, Cheng Tan, Jun Xia, Bozhen Hu, and Stan Z. Li. A graph is worth k words: Euclideanizing graph using pure transformer. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14681–14701. PMLR, 21–27 Jul 2024b.

Zijie Geng, Shufang Xie, Yingce Xia, Lijun Wu, Tao Qin, Jie Wang, Yongdong Zhang, Feng Wu, and Tie-Yan Liu. De novo molecular generation via connection-aware motif mining, 2023.

F.A. Gers and E. Schmidhuber. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001. doi: 10.1109/72.963769.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276, February 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00572.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one, 2020.

Francesca Grisoni. Chemical language models for de novo drug design: Challenges and opportunities. *Current Opinion in Structural Biology*, 79:102527, 2023.

Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. *arXiv preprint arXiv:2303.03543*, 2023.

Leon Hetzel, Johanna Sommer, Bastian Rieck, Fabian Theis, and Stephan Günnemann. Magnet: Motif-agnostic generation of molecules from shapes, 2023.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.

Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, 11, 1998.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100 *Machine Learning: Science and Technology*, 1(4):045024, October 2020. ISSN 2632-2153. doi: 10.1088/2632-2153/aba947.

Pin-Kuang Lai, Daniel T Tresnak, and Benjamin J Hackel. Identification and elucidation of proline-rich antimicrobial peptides with enhanced potency and delivery. *Biotechnology and bioengineering*, 116(10): 2439–2450, 2019.

Julia A Lasserre, Christopher M Bishop, and Thomas P Minka. Principled hybrids of generative and discriminative models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pp. 87–94. IEEE, 2006.

Travis J Lawrence, Dana L Carper, Margaret K Spangler, Alyssa A Carrell, Tomás A Rush, Stephen J Minter, David J Weston, and Jessy L Labbé. ampeppy 1.0: a portable and accurate antimicrobial peptide prediction tool. *Bioinformatics*, 37(14):2058–2060, 11 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa917. URL `https://doi.org/10.1093/bioinformatics/btaa917`.

Chenkai Li, Darcy Sutherland, S Austin Hammond, Chen Yang, Figali Taho, Lauren Bergman, Simon Houston, René L Warren, Titus Wong, Linda M N Hoang, Caroline E Cameron, Caren C Helbing, and Inanc Birol. AMPlify: attentive deep learning model for discovery of novel antimicrobial peptides effective against WHO priority pathogens. *BMC Genomics*, 23(1):77, January 2022.

Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guotong Xie, and Sen Song. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. *Briefings in Bioinformatics*, 22(6):bbab109, 2021.

Tingting Li, Xuanbai Ren, Xiaoli Luo, Zhuole Wang, Zhenlu Li, Xiaoyan Luo, Jun Shen, Yun Li, Dan Yuan, Ruth Nussinov, Xiangxiang Zeng, Junfeng Shi, and Feixiong Cheng. A foundation model identifies broad-spectrum antimicrobial peptides against drug-resistant bacterial infection. *Nat. Commun.*, 15(1): 7538, August 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. Graphebm: Molecular graph generation with energy-based models. *arXiv preprint arXiv:2102.00546*, 2021a.

Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

Shengchao Liu, Mehmet F Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in neural information processing systems*, 32, 2019.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021b.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. Graphdf: A discrete flow model for molecular graph generation. In *International conference on machine learning*, pp. 7192–7203. PMLR, 2021.

Krzysztof Maziarz, Henry Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs, 2022.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, et al. Controlled decoding from language models. *arXiv preprint arXiv:2310.17022*, 2023.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. In *International Conference on Machine Learning*, pp. 4723–4732. PMLR, 2019.

Rıza Özçelik, Laura van Weesep, Sarah de Ruiter, and Francesca Grisoni. peptidy: A light-weight python library for peptide representation in machine learning. 2025.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020.

Célio Dias Santos-Júnior, Yiqian Duan, Hui Chong, Thomas S.B. Schmidt, Anthony Fullam, Peer Bork, Xing-Ming Zhao, and Luis Pedro Coelho. Ampsphere : the worldwide survey of prokaryotic antimicrobial peptides, May 2022. URL `https://doi.org/10.5281/zenodo.6511404`.

Philippe Schwaller, Daniel Probst, Alain C. Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. Mapping the space of chemical reactions using attention-based neural networks. *ChemRxiv*, 2020. doi: 10.26434/chemrxiv.9897365.v4.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1):120–131, 2018.

Yu Shi, Shuxin Zheng, Guolin Ke, Yifei Shen, Jiacheng You, Jiyan He, Shengjie Luo, Chang Liu, Di He, and Tie-Yan Liu. Benchmarking graphormer on large-scale molecular modeling datasets. *arXiv preprint arXiv:2203.04810*, 2022.

Simon Steshin. Lo-hi: Practical ml drug discovery benchmark. In *Advances in Neural Information Processing Systems*, 2023.

Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

A Sultan, J Sieg, M Mathea, and A Volkamer. Transformers for molecular property prediction: Lessons learned from the past five years. arxiv preprint arxiv: 240403969. 2024.

Paulina Szymczak, Marcin Możejko, Tomasz Grzegorzek, Radosław Jurczak, Marta Bauer, Damian Neubauer, Karol Sikora, Michał Michalski, Jacek Sroka, Piotr Setny, Wojciech Kamysz, and Ewa Szczurek. Discovering highly potent antimicrobial peptides with deep generative model hydramp. *bioRxiv*, 2023. doi: 10.1101/2022.01.27.478054.

Jakub M Tomczak. Deep generative modeling for neural compression. In *Deep Generative Modeling*. Springer, 2022.

Marcelo D. T. Torres, Tianlai Chen, Fangping Wan, Pranam Chatterjee, and Cesar de la Fuente-Nunez. Generative latent diffusion language modeling yields anti-infective synthetic peptides. *bioRxiv*, 2025. doi: 10.1101/2025.01.31.636003. URL https://www.biorxiv.org/content/early/2025/02/01/2025.01.31.636003.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Colin M. Van Oort, Jonathon B. Ferrell, Jacob M. Remington, Safwan Wshah, and Jianing Li. Ampgan v2: Machine learning-guided design of antimicrobial peptides. *Journal of Chemical Information and Modeling*, 61(5):2198–2207, 2021. doi: 10.1021/acs.jcim.0c01441. PMID: 33787250.

Derek van Tilborg, Luke Rossen, and Francesca Grisoni. Molecular deep learning at the edge of chemical space. 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, pp. 429–436, 2019.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.

David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, February 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning, 2018.

Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z. Li. Mole-BERT: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*, 2023.

Yu-Ting Xiang, Guang-Yi Huang, Xing-Xing Shi, Ge-Fei Hao, and Guang-Fu Yang. 3d molecular generation models expand chemical space exploration in drug design. *Drug Discovery Today*, pp. 104282, 2024.

Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

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.

A Impact Statement

The goal of this this work is to improve the field of deep generative modeling and, potentially, drug design. An example of potential malicious use of our approach would be training a deep generative model for generating new toxic molecules. However, the intention of this paper is to provide tools that will facilitate designing new potential medications.

B Acknowledgments

We thank Hassan Akell for insightful discussions and careful review of the theoretical part of this paper. Their feedback substantially improved the clarity, rigor, and presentation of the theoretical analysis.

C Notation

Symbol	Meaning
$[N]$	Set of integers $1, \ldots, N$
$\mathbf{A}$	Matrix
$\mathbf{A}^T$	Transposed matrix $\mathbf{A}$
$\mathbf{A}_i$, $\mathbf{A}_{ij}$, $\mathbf{A}^{ij}$	Matrix indexed for some purpose
$(\mathbf{A})_i, \mathbf{A}[i], A_i$	The i-th row of matrix $\mathbf{A}$
$(\mathbf{A})_{ij}, \mathbf{A}[i,j], A_{ij}$	The i-th, j-th entry of matrix $\mathbf{A}$
$\mathbf{a}$	Vector (column-vector)
$\mathbf{a}_i$, $\mathbf{a}_{ij}$, $\mathbf{a}^{ij}$	Vector indexed for some purpose
$(\mathbf{a})_i, \mathbf{a}[i], a_i$	The i-th entry of vector $\mathbf{a}$
a	Scalar
$\mathcal{X}$	input space, i.e. the space of all possible inputs, data examples
$\mathcal{Y}$	target space i.e. the space of all possible property values
$p(\mathbf{x}, y)$	joint data distribution
$p_\theta(\mathbf{x}, y)$	joint model parametrized by parameters $\theta \in \Theta$
$p_\theta(y \mid \mathbf{x})$	predictive model parametrized by parameters $\theta \in \Theta$
$p_\theta(\mathbf{x})$	generative model parametrized by parameters $\theta \in \Theta$

D Proofs

D.1 Gradient Interference

Lemma D.1. *Let $\mathbf{x} \in \mathbb{R}^I$ and define*

$$a_i = \mathrm{softmax}(\mathbf{x})_i = \frac{\exp x_i}{\sum_{k=1}^{I} \exp x_k} \text{ , for } i = 1, \ldots, I.$$

The Jacobian of the softmax is given by

$$\frac{\partial a_i}{\partial x_j} \;=\; a_i\big(\,\delta_{ij} - a_j\big), \qquad i, j = 1, \ldots, I,$$

where δ_{ij} is the Kronecker delta, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Proof. Differentiate the quotient $a_i = \exp x_i / \sum_k \exp x_k$ using the product and chain rules (Petersen et al., 2008). □

Corollary D.2. *Let $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{T \times d}$ and the attention score matrix $\mathbf{S}_\rightarrow$ with a causal mask $\mathbf{M}_\rightarrow$ be defined as*

$$\mathbf{S}_\rightarrow = \frac{\mathbf{Q}\,\mathbf{K}^T}{\sqrt{d}} + \mathbf{M}_\rightarrow \text{ , where } (\mathbf{M}_\rightarrow)_{ij} = \begin{cases} 0 & \text{, if } i \geq j \\ -\infty & \text{, if } i < j. \end{cases}$$

For a fixed row index $t \in [T]$, define the attention score row-vector $\mathbf{s}_t = (\mathbf{S})_t \in \mathbb{R}^T$ and the corresponding row-wise softmax output as $\mathbf{a}_t = \mathrm{softmax}(\mathbf{s}_t) \in \mathbb{R}^T$. The Jacobian of the softmax output $\mathbf{a}_t$ with respect to masked attention score $\mathbf{s}_t$ is given by

$$\frac{\partial(\mathbf{a}_t)_i}{\partial(\mathbf{s}_t)_j} = (\mathbf{a}_t)_i\big(\,\delta_{ij} - (\mathbf{a}_t)_j\big).$$

Hence, if $i < t$ or $j < t$, while $i \neq j$, then $\frac{\partial(\mathbf{a}_t)_i}{\partial(\mathbf{s}_t)_j} = 0$.

Proof. Lemma D.1 gives the derivative of the softmax. As the causal mask sets $(\mathbf{s}_t)_j = -\infty$ for every $j < t$, the corresponding probabilities satisfy $(\mathbf{a}_t)_j = 0$. □

E Pre-training Details

We implement Hyformer using a LLAMA backbone (Touvron et al., 2023). Depending on the size of the pretraining dataset, we scale Hyformer to 8.7M parameters for GuacaMol and 50M parameters for the UniMol and peptide datasets. These configurations align model capacity with dataset size and ensure a fair comparison with prior work: the 8.7M model is comparable to MolGPT (Bagal et al., 2022), while the 50M variant matches the scale of Uni-Mol (Zhou et al., 2023) and Graph2Seq (Gao et al., 2024b). For GuacaMol, we apply 2× data augmentation using non-canonical SMILES enumeration (Bjerrum, 2017; Arús-Pous et al., 2019) to increase molecular diversity. All models are pretrained using pre-computed molecular descriptors (Yang et al., 2019). The balancing of the tasks $(p_{\texttt{[LM]}}, p_{\texttt{[MLM]}}, p_{\texttt{[PRED]}})$ is set to (0.90, 0.05, 0.05) and (0.80, 0.10, 0.10), respectively.

We use SMILES (Weininger, 1988) or amino acid sequences as molecular representations across all experiments. For tokenization, we adopt an extended character-level tokenizer for SMILES, based on Schwaller et al. (2020), and use the ESM-2 tokenizer (Lin et al., 2023) for peptides.

We pre-train Hyformer using a batch size of 1024 for up to 50K or 250K iterations, depending on model size. Training is performed with the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1 \times 10^{-5}$, weight decay $= 1 \times 10^{-1}$), using a peak learning rate of 6×10^{-4} with cosine decay and 5000 warm-up steps. We use gradient clipping with a maximum norm of 1.0. All input sequences are padded to a fixed length of 128 tokens. Training is conducted using bfloat16 precision on a single NVIDIA H100 80GB HBM3 GPU.

Table 7: Architectural details of Hyformer.

Num. param.	Embed. dim	Hidden dim	#Layers	# Att. Heads
8.7M	256	1024	8	8
50M	512	2048	12	8

F Experimental Details

All fine-tuning and inference is conducted using float32 precision on a single NVIDIA V100 32GB GPU.

F.1 Conditional Molecule Generation

We jointly fine-tune Hyformer, pretrained on GuacaMol dataset, for 10 epochs with a batch size of 256. The peak learning rate is selected from the set $\{1\text{e}{-4}, 2\text{e}{-4}, 3\text{e}{-4}, 4\text{e}{-4}, 5\text{e}{-4}, 5\text{e}{-4}, 6\text{e}{-4}\}$, based on root mean squared error (RMSE) with respect to the target property. During fine-tuning, we set the task probability vector to $(p_{\texttt{[LM]}}, p_{\texttt{[PRED]}}) = (0.5, 0.5)$ and do not perform hyperparameter search over this setting, as it yields satisfactory performance by default. For the non-joint variant of Hyformer, we freeze the pretrained model and fine-tune only the prediction head. This avoids catastrophic forgetting of the generative capability when removing the generative loss during training. For each target property value, we sample 100K unique molecules, with a wall-clock time of 78 ± 1 seconds, and retain those passing a manually defined threshold,

using multinomial top-k sampling with $\tau = 0.9$ and $k = 10$. Note that reported SA scores are normalized, following (Gao et al., 2024b).

Table 8: Conditional generative performance on GuacaMol dataset across all targets. Best model is marked **bold**.

	Pretrain	Joint	Metric	QED=0.5	QED=0.7	QED=0.9	SA=0.7	SA=0.8	SA=0.9	logP=0.0	logP=2.0	logP=4.0	Avg.
MolGPT	✗	✗	MAD ↓	0.081	0.082	0.097	0.024	0.019	0.013	0.304	0.239	0.286	0.127
			SD ↓	0.065	0.066	0.092	0.022	0.016	0.013	0.295	0.232	0.258	0.118
			Validity ↑	0.985	0.985	0.984	0.975	0.988	0.995	0.982	0.983	0.982	0.984
GraphGPT-1W-C	✗	✗	MAD ↓	0.041	0.031	0.077	0.012	0.028	0.031	0.103	0.189	0.201	0.079
			SD ↓	0.079	0.077	0.121	0.055	0.062	0.070	0.460	0.656	0.485	0.229
			Validity ↑	0.988	0.995	0.991	0.995	0.991	0.998	0.980	**0.992**	0.991	0.991
	✓	✗	MAD ↓	0.032	0.033	0.051	**0.002**	0.009	0.022	**0.017**	0.190	0.268	0.069
			SD ↓	0.080	0.075	0.090	0.042	0.037	0.062	0.463	0.701	0.796	0.261
			Validity ↑	**0.996**	**0.998**	**0.999**	**0.995**	**0.999**	0.996	0.994	0.990	0.992	**0.995**
Hyformer	✓	✗	MAD ↓	0.035 (0.000)	0.032 (0.001)	0.027 (0.007)	0.020 (0.001)	0.016 (0.000)	0.009 (0.001)	0.131 (0.012)	0.135 (0.007)	0.127 (0.011)	0.059 (0.004)
			SD ↓	0.049 (0.000)	0.046 (0.002)	0.039 (0.010)	0.027 (0.002)	0.021 (0.000)	0.012 (0.001)	0.162 (0.015)	0.174 (0.010)	0.175 (0.016)	0.078 (0.006)
			Validity ↑	0.993 (0.003)	0.993 (0.003)	0.993 (0.004)	0.986 (0.009)	0.985 (0.001)	**0.999 (0.002)**	0.978 (0.031)	0.983 (0.004)	**0.995 (0.007)**	0.989 (0.007)
	✓	✓	MAD ↓	**0.010 (0.001)**	**0.009 (0.000)**	**0.006 (0.001)**	0.008 (0.001)	**0.005 (0.000)**	**0.001 (0.000)**	0.033 (0.005)	**0.044 (0.001)**	**0.046 (0.001)**	**0.018 (0.001)**
			SD ↓	**0.018 (0.002)**	**0.018 (0.002)**	**0.010 (0.003)**	**0.015 (0.003)**	**0.009 (0.002)**	**0.004 (0.000)**	**0.037 (0.007)**	**0.057 (0.003)**	**0.059 (0.001)**	**0.025 (0.003)**
			Validity ↑	0.983 (0.009)	0.990 (0.006)	0.996 (0.005)	0.976 (0.005)	0.981 (0.002)	**0.999 (0.002)**	**1.000 (0.000)**	0.991 (0.007)	0.971 (0.011)	0.987 (0.005)

Algorithm 2 Conditional sampling with Hyformer

Input: Number of examples to sample K, batch size B, condition Y, model parameters θ.

1: $\mathcal{D}_{sampled} = \emptyset$
2: **while** $|\mathcal{D}_{sampled}| < K$ **do**
3: Sample B many examples $(\mathbf{x}, y) \sim p_\theta(\mathbf{x}, y)$
4: Accept examples $\mathcal{D}_{batch} = \{(\mathbf{x}, y) \mid y \in Y\}$
5: Append dataset $\mathcal{D}_{sampled} = \mathcal{D}_{sampled} \cup \mathcal{D}_{batch}$
6: **end while**

F.2 Out-of-Distribution Molecular Property Prediction Task

We use Hyformer pre-trained on UniMol dataset and perform a grid search over hyperparameters, as detailed in Table 9, with end-to-end joint fine-tuning, with early stopping triggered if the validation loss does not improve for 5 consecutive epochs. Results in Table 2 are reported from (Steshin, 2023).

Table 9: Hyperparameter ranges for the grid search hyperparameter optimization on out-of-distribution molecular property prediction task.

Hyperparameter	Search Range
Max Epochs	{20, 50, 100}
Batch Size	{64, 128, 256}
Learning Rate	[1e-5, 6e-4]
Weight Decay	[1e-2, 1e-1]
Pooler Dropout	[0.0, 0.2]
Learning Rate Decay	{True, False}
$(p_{\texttt{[LM]}}, p_{\texttt{[PRED]}})$	{(0.0, 1.0), (0.1, 0.9)}

F.3 Molecular Representation Learning Task

For KNN probe, we use the euclidean norm to pick K most similar molecules. For each dataset, we search the parameter K in the set $\{1, 3, 5, 100, 300, 500, 1000, 3000, 5000\}$ and pick K with the best performance on the validation split. For linear probe, we report the results of linear probe with L2 regularization added. If the validation loss between the epochs does not decrease by more than 0.0001 for 10 consecutive epochs, we terminate the training process early. All results in Table 3 are ours.

F.4 Molecule Generation Task

For generation, we use HYFORMER pre-trained on GuacaMol and sample using multinomial top-k sampling, with $k = 10$ and varying temperature $\tau = \{0.9, 1.0, 1.1\}$.

In Table 4, baseline results for JTVAE and MAGNeT are reported from (Hetzel et al., 2023), for MoLeR and MiCaM from (Geng et al., 2023), for VAE, LSTM from (Brown et al., 2019), for MolGPT from (Bagal et al., 2022).

F.5 Molecular Property Prediction Task

We use HYFORMER pre-trained on UniMol dataset and perform a grid search over hyperparameters, as detailed in Table 10, with end-to-end predictive fine-tuning run for a maximum of 20 epochs, with early stopping triggered if the validation loss does not improve for 5 consecutive epochs. Results in Table 5 are reported from (Zhou et al., 2023; Gao et al., 2024b).

Table 10: Hyperparameter ranges for the grid search hyperparameter optimization on molecular property prediction task.

HYPERPARAMETER	SEARCH RANGE
BATCH SIZE	{16, 64, 128, 256}
LEARNING RATE	[1E-5, 1E-3]
WEIGHT DECAY	[1E-2, 3E-1]
POOLER DROPOUT	[0.0, 0.2]
LEARNING RATE DECAY	{TRUE, FALSE}

F.6 Antimicrobial Peptide Design

Dataset We construct a general-purpose peptide dataset and an AMP-specific dataset. For the general purpose dataset, we collect 3459247 peptide sequences with length 8-50 from the combined Peptipedia (Cabas-Mora et al., 2024) and UniProt (Consortium, 2024) datasets and apply CDHIT filtering with a similarity threshold of 90%. For the AMP-specific dataset, we collect 1056321 sequences from combining the Peptipedia (Cabas-Mora et al., 2024), filtered with Antigram (-), Antigram (+), Antibacterial and Antimicrobial keywords, Uniprot (Consortium, 2024) with the keywords antimicrobial and AMPSphere (Santos-Júnior et al., 2022), and applying CDHIT filtering with a similarity threshold of 90%.

Pre-trainig We pre-train HYFORMER in a two-stage manner, by first training on the general-purpose, followed by training on the AMP specific dataset with peak learning rate equal to 4e−4. All additional details follow Appendix E.

Fine-tuning We fine-tune HYFORMER for a maximum of 10 epochs, with batch size 64, peak learning rate 5e−5 and early stopping, with task probabilities $(p_{\texttt{[LM]}}, p_{\texttt{[PRED]}})$ equal to (0.6, 0.4). Additionally, we freeze the first four layers of the model.

G Additional Experiments

G.1 Unconditional Molecule Generation on MOSES benchmark

To additionally evaluate the unconditional generative performance of HYFORMER, we perform an evaluation on the MOSES benchmark. Analogously to unconditional molecule generation in Section 5.2.1, we scale HYFORMER to 8.5M parameters and follow all the training details in Appendix E for GuacaMol dataset. We compare HYFORMER, across various sampling temperatures τ, to baseline unconditional and few-shot generative models, as reported in (Gao et al., 2024b).

HYFORMER successfully generates valid, unique, novel and diverse molecules, rivaling other unconditional and few-shot generative models.

Table 11: Unconditional generative performance on MOSES benchmark. The best model in each category is marked **bold**.

Model	Validity ↑	Unique ↑	Novelty ↑	IntDiv1 ↑	IntDiv2 ↑
Unconditional					
HMM	0.076	0.567	**0.999**	0.847	0.810
NGram	0.238	0.922	0.969	**0.874**	0.864
Combinatorial	**1.000**	0.991	0.988	0.873	**0.867**
CharRNN	0.975	0.999	0.842	0.856	0.850
VAE	0.977	0.998	0.695	0.856	0.850
AEE	0.937	0.997	0.793	0.856	0.850
LatentGAN	0.897	0.997	0.949	0.857	0.850
JT-VAE	**1.000**	0.999	0.914	0.855	0.849
MolGPT	0.994	**1.000**	0.797	0.857	0.851
Hyformer$_{\tau=0.9}$	0.996	**1.000**	0.701	0.851	0.845
Hyformer$_{\tau=1.0}$	0.991	**1.000**	0.749	0.856	0.850
Hyformer$_{\tau=1.1}$	0.986	**1.000**	0.791	0.861	0.855
Few-Shot					
GraphGPT-1W$_{s=0.25}$	**0.995**	0.995	0.255	0.854	0.850
GraphGPT-1W$_{s=0.5}$	0.993	0.996	0.334	0.856	0.848
GraphGPT-1W$_{s=1.0}$	0.978	0.997	0.871	**0.860**	**0.857**
GraphGPT-1W$_{s=2.0}$	0.972	**1.000**	**1.000**	0.850	0.847

G.2 Qualitative Evaluation of Generated Molecules

To investigate the effect of sampling temperature on the structural diversity and chemical quality of generated molecules, we show molecules sampled in the unconditional generation task (Section 5.2.1), at temperatures $\tau = 0.9$, 1.0, and 1.1. For each sampled molecule, we additionally report four chemical properties: molecular partition coefficient (LogP), topological polar surface area (TPSA), quantitative estimate of drug-likeness (QED) and molecular weight (MW). At $\tau = 0.9$, the model generates drug-like molecules, with the majority exhibiting QED ≥ 0.7 and MW < 500 g/mol (Fig. 3). At $\tau = 1.0$, the sampling process yields molecules with greater structural diversity (Fig. 4). Despite the increased exploration of chemical space, some molecules exhibit lower QED values. At $\tau = 1.1$, the model produces molecules with less common substituent patterns. Some of these structures exceed traditional drug-likeness thresholds, such as MW > 500 g/mol or LogP > 5, according to Lipinski's Rule of Five (Fig. 5). Additionally, we investigate molecules generated in the conditional generation task in Section 5.1.1 (Figure 6, 7 and 8).

G.3 Qualitative Evaluation of Learned Representations

We next examine the Hyformer embeddings in the context of the chemical properties of the molecules (Fig. 9). To this end, we randomly sample 20,000 molecules and pass them through Hyformer's encoder, pre-trained for molecular property prediction in Section 5.2.2, to obtain molecule embeddings. We visualize the embeddings in two dimensions through principal components analysis (PCA) and color them according to their four chosen chemical properties (LogP, TPSA, QES, MW).

Qualitatively, the spatial arrangement of molecules is clearly connected to their chemical properties. Furthermore, embeddings exhibit a smooth profile of change w.r.t. each property. These observations indicate that Hyformer learns well-behaved, information-rich molecular representations.

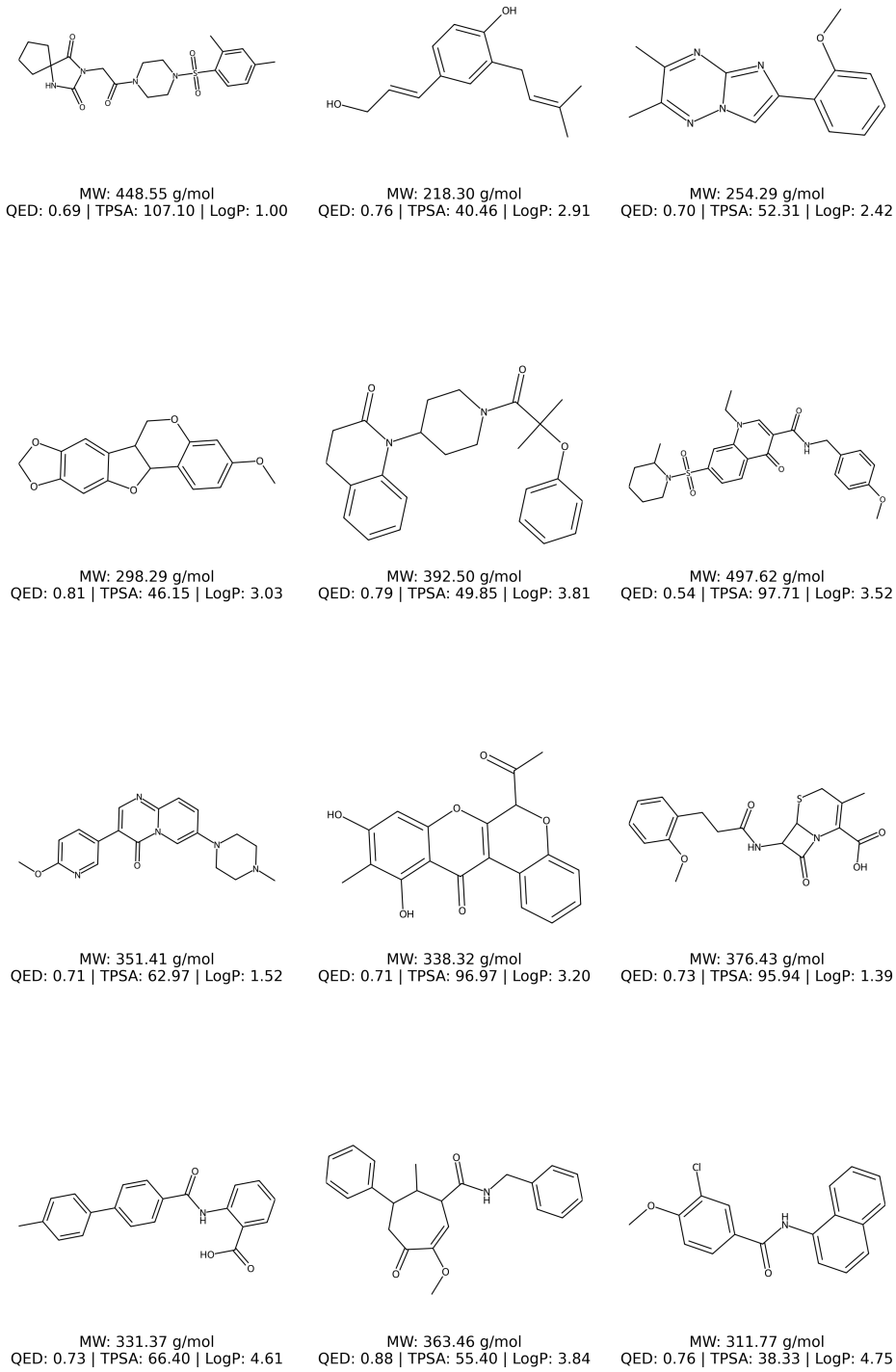

Figure 3: Structures of the twelve generated molecules with Hyformer when the sampling temperature is 0.9, visualized using RDKit, together with their properties.

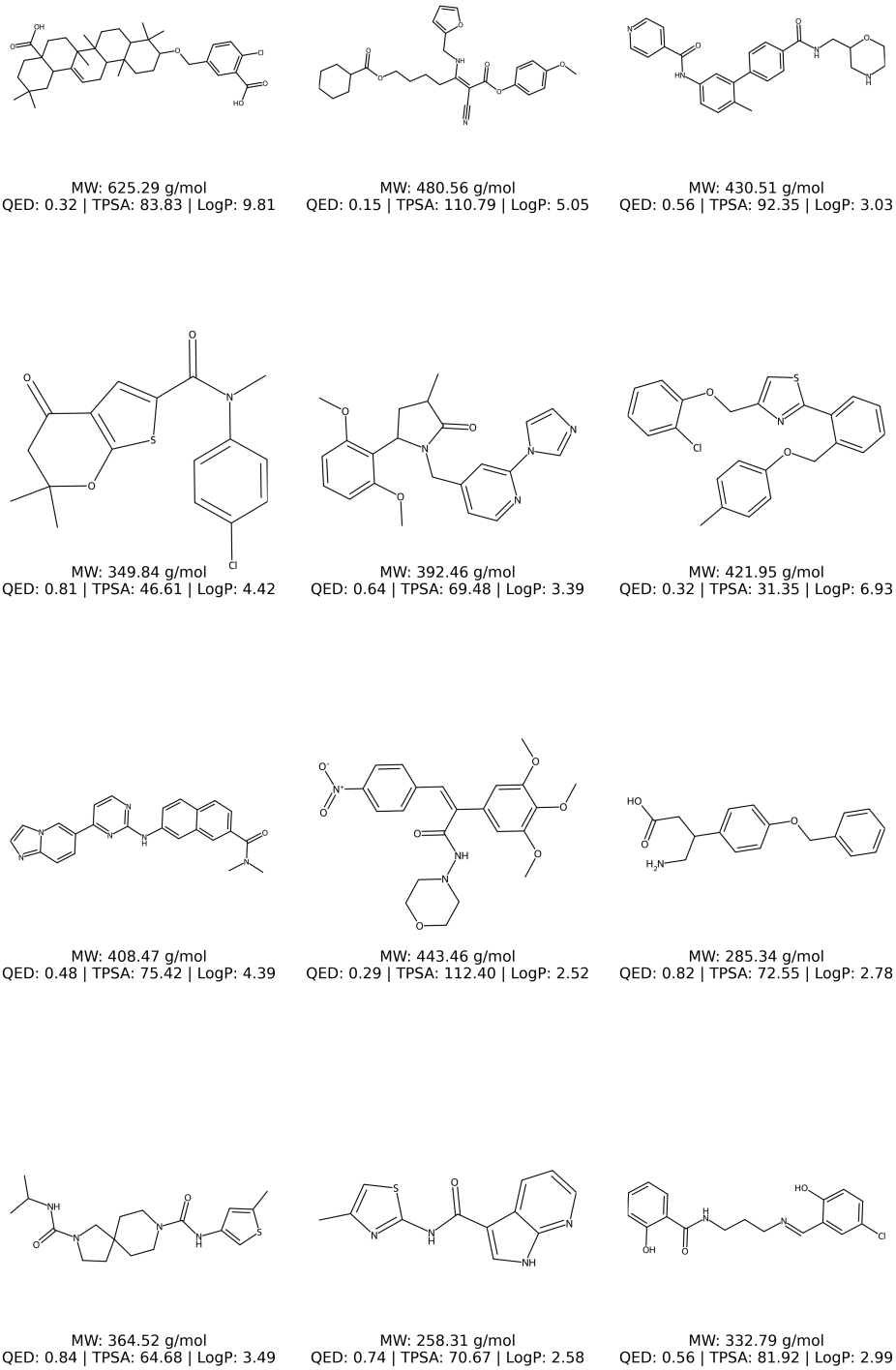

Figure 4: Structures of the twelve generated molecules with Hyformer when the sampling temperature is 1.0, visualized using RDKit, together with their properties.

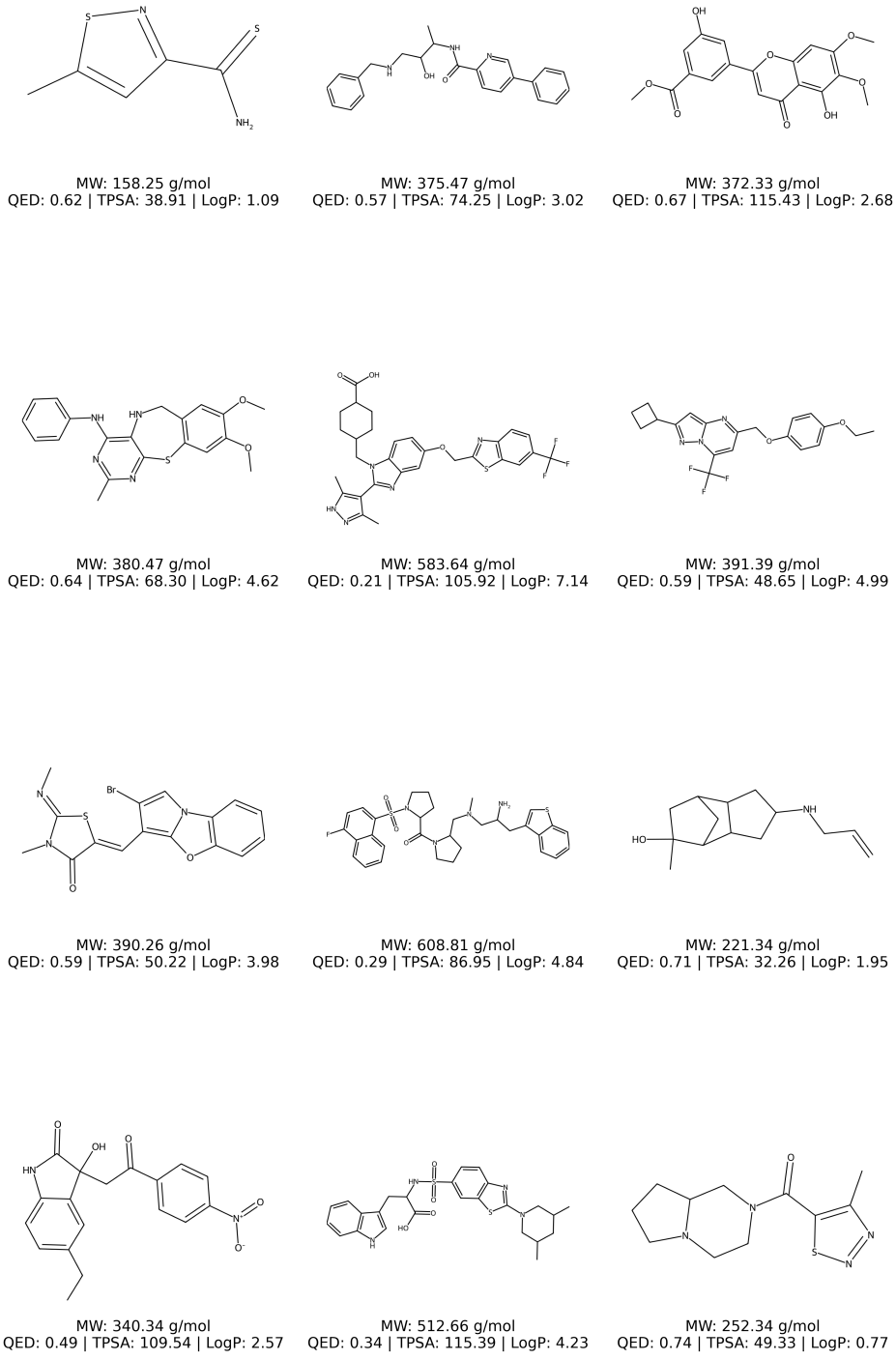

Figure 5: Structures of the twelve generated molecules with Hyformer when the sampling temperature is 1.1, visualized using RDKit, together with their properties.

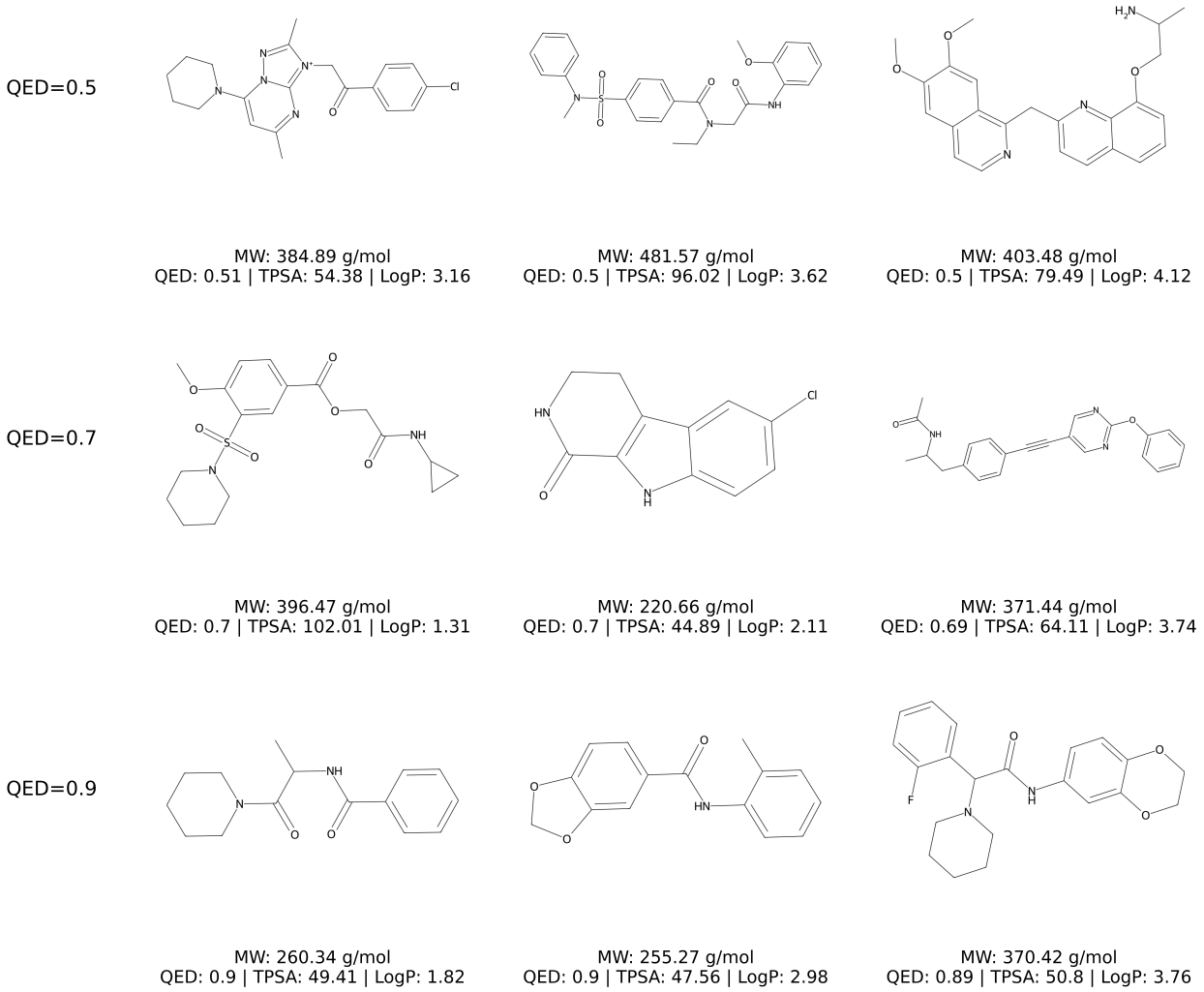

Figure 6: Structures of molecules generated by Hyformer conditioned on QED values, visualized using RDKit, along with their chemical properties.

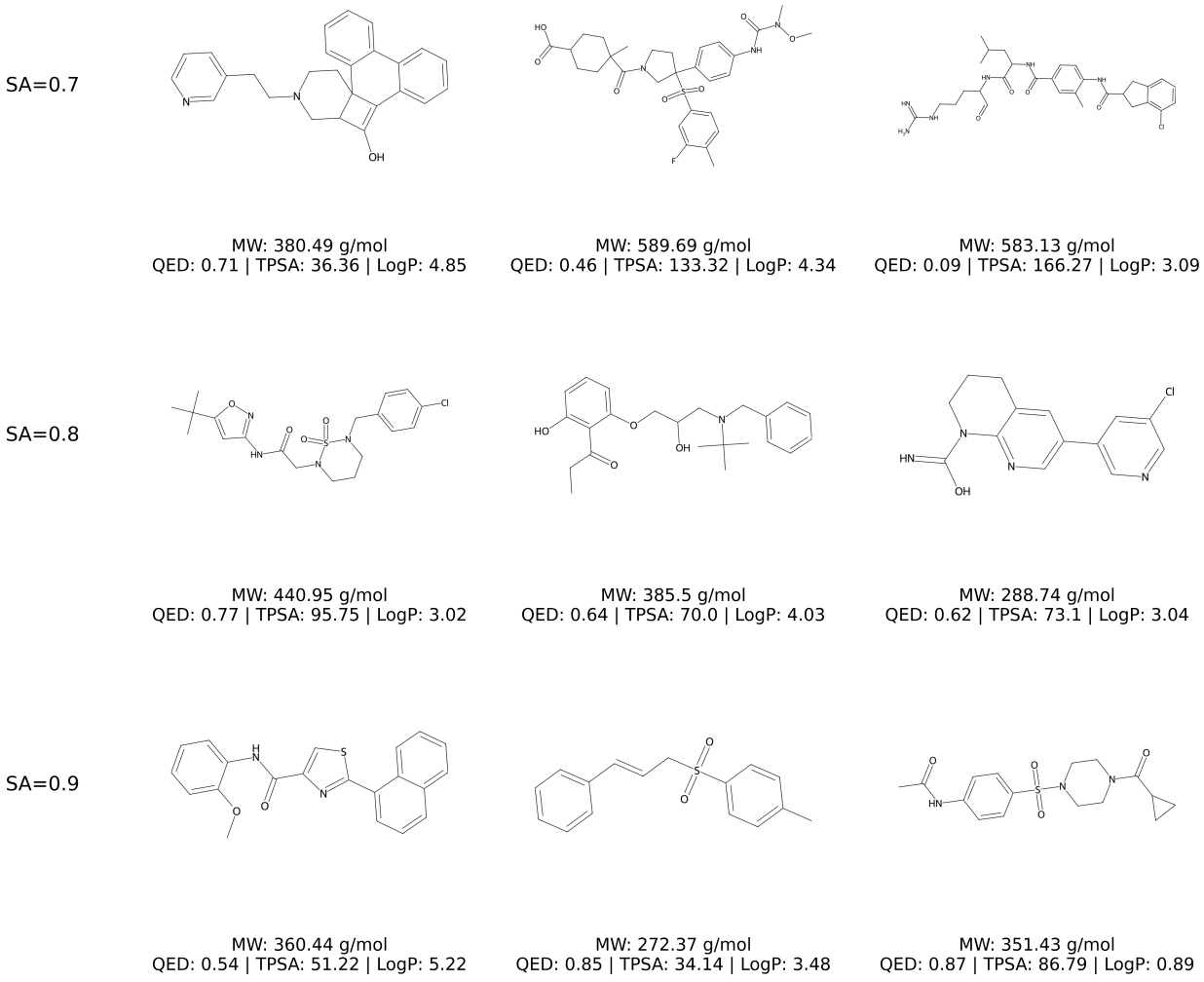

Figure 7: Structures of molecules generated by Hyformer conditioned on SA score, visualized using RDKit, along with their chemical properties.

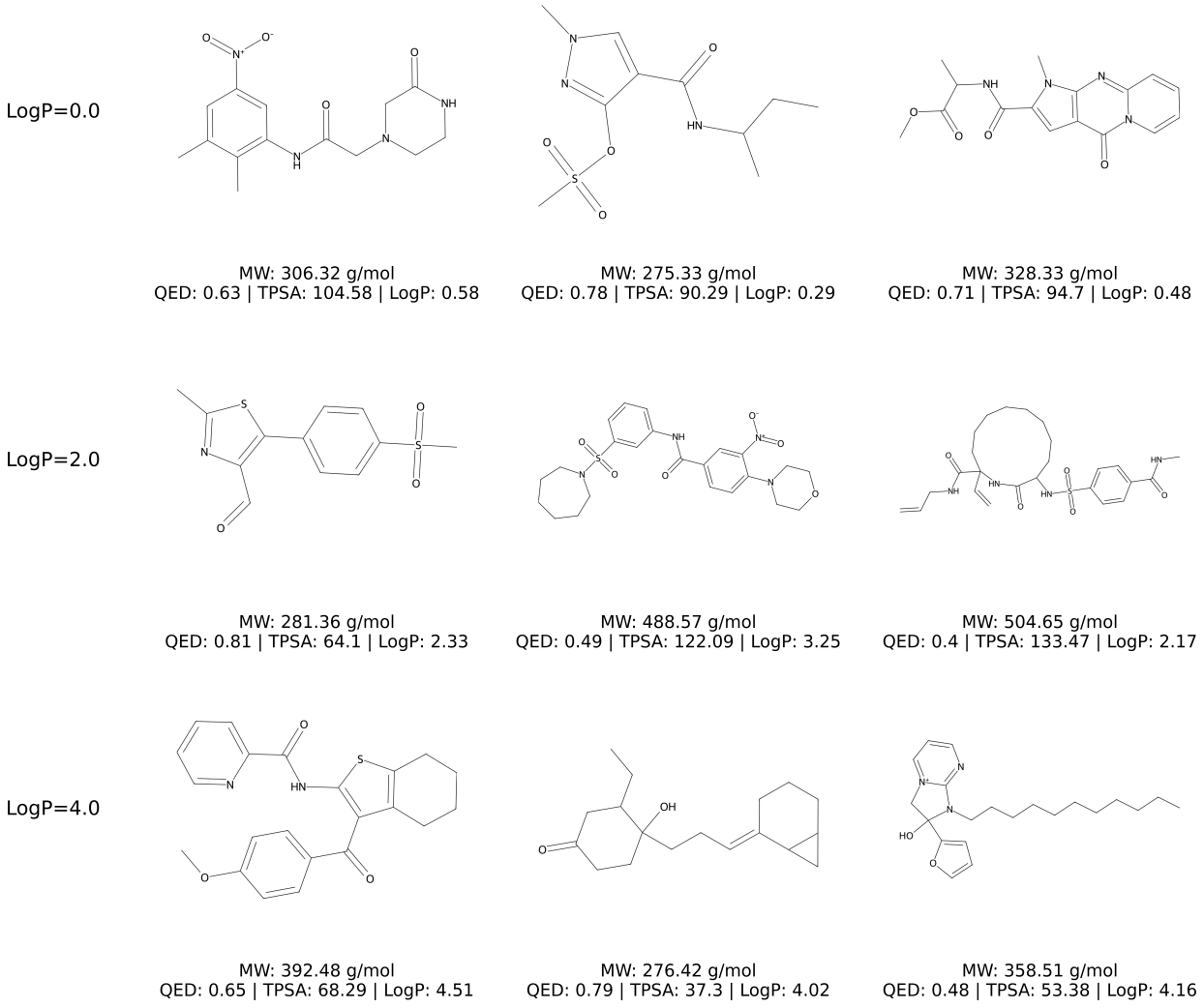

Figure 8: Structures of molecules generated by Hyformer conditioned on LogP values, visualized using RDKit, along with their chemical properties.

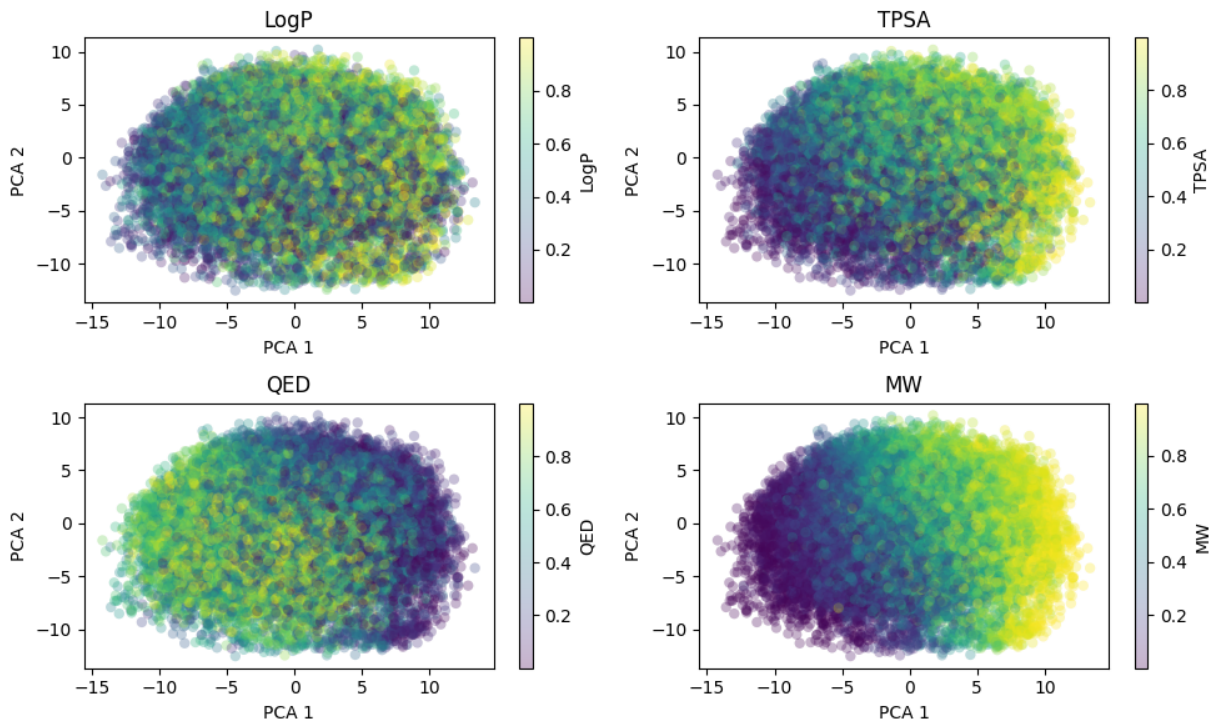

Figure 9: Hyformer's molecular embeddings. The considered chemical properties are normalized to lie in the $[0, 1]$ interval.