Tail Extrapolation in target-aware conditional molecule generation

Firstname1 Lastname1 [*1] **Firstname2 Lastname2** [*1 2] **Firstname3 Lastname3** [2] **Firstname4 Lastname4** [3] **Firstname5 Lastname5** [1] **Firstname6 Lastname6** [3 1 2] **Firstname7 Lastname7** [2] **Firstname8 Lastname8** [3] **Firstname8 Lastname8** [1 2]

Abstract

Generative models, such as diffusion-based models and large language models, have become increasingly popular in cheminformatics research. These models have shown promise in accelerating the discovery of molecules. However, they are hindered by data scarcity and struggle to accurately generate molecules when the desired properties lie outside the range of the training data, a task known as *tail extrapolation* in statistics. To this end, we propose *tail-extrapolative generative models* in this work. The key idea is to adapt pre-additive noise models, which can provably perform tail extrapolation in classical regression tasks, to a variety of conditional generative models. Across empirical studies, we find that tail-extrapolative generative models exhibit improved extrapolation capabilities. They enable the generation of molecules with properties that more closely align with desired targets. Furthermore, these models enhance the diversity of the generated molecules compared to existing approaches, representing an advancement in molecular design.

1. Introduction

Discovering new molecules with desired properties is needed to address critical technological challenges ranging from energy storage to drug-molecule development. The traditional trial-and-error approach of synthesizing and testing molecules is expensive, inefficient, and time-consuming. Likewise, it is too computationally expensive to use purely quantum mechanical calculations to adequately screen the vast space of possible molecules for target applications. Thus, it is important to develop machine learning methods that efficiently navigate chemical space to discover molecules capable of addressing important problems.

Generative models are a promising approach to address the molecule discovery challenge (Bilodeau et al., 2022). Although generative modeling approaches have demonstrated utility for generating new molecules with target properties, they are hindered by data scarcity and struggle to accurately generate molecules when the desired properties lie outside the range of the training data, a task known as tail extrapolation in statistics. It is important to emphasize that when trying to generate new molecules, researchers typically look for molecules that have *exceptional* properties, thus making them rare and likely outside the range of the training data. Here we aim to address this challenge via tail extrapolative conditional molecule generation.

Generally speaking, we are interested in property-constrained sampling (Bilodeau et al., 2022; Anstine & Isayev, 2023). Formally, let $f : \mathcal{M} \mapsto \mathcal{Y}$ be the oracle function that maps the molecule $M \in \mathcal{M}$ to the properties $y \in \mathcal{Y}$. Property-constrained sampling learns the distribution $p(\mathcal{M})$ given a set of N molecules $\{M_i\}_{i=1}^N$ and samples new molecules that satisfy the property constraints $M_{\text{new}} \sim p(M \mid y)$, given the desired property y.

1.1. Molecule representations and conditional generative models

One popular representation of a molecule M is a molecular formula string, expressed as a sequence of symbols $(s_0, s_1, \ldots, s_n)$, such as Simplified molecular-input line-entry system (SMILES) (Weininger, 1988) and Self-referencing embedded strings (SELFIES) (Krenn et al., 2020). If the 3D geometry is also available, a molecule M containing n atoms can be expressed as $M = (\mathbf{x}, \mathbf{v})$ where $\mathbf{x} \in \mathbb{R}^{n\times 3}$ denotes the Cartesian coordinates for each of the $j = 1, \ldots, n$ atoms in the system, $\mathbf{v} \in \mathbb{R}^{n\times d}$ denotes the attribute matrix, containing d atom features (e.g., atom type, atom charge, etc.). Given different molecule representations, different generative models are designed accordingly.

1.1.1. Conditional generative models for 3D molecular data

For data that contains continuous 3D coordinates, one can leverage diffusion-based models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Song et al., 2021; Song & Ermon, 2019; Song et al., 2020), which have achieved great success in image generation tasks and demonstrated an even higher quality of image synthesis performance (Dhariwal &

Nichol, 2021) compared to variational autoencoders (VAEs) (Kingma & Welling, 2022; van den Oord et al., 2017; Razavi et al., 2019), flows (Dinh et al., 2017; Kingma & Dhariwal, 2018; Rezende & Mohamed, 2015), auto-regressive models (Menick & Kalchbrenner, 2018; Oord et al., 2016), and generative adversarial networks (GANs) (Goodfellow et al., 2014; Karras et al., 2020; 2021).

Diffusion-based property-constrained sampling often assumes a *conditional forward diffusion* process on the concatenated 3D atom coordinates and atom attributes $[\mathbf{x}, \mathbf{v}]$ and a *conditional generative denoising process* using equivariant neural networks, both defined as Markov chains. The conditional forward diffusion process gradually injects Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ into data, $q(\mathbf{z}_t \mid [\mathbf{x}, \mathbf{v}]) = \alpha_t[\mathbf{x}, \mathbf{v}] + \sigma_t \boldsymbol{\varepsilon}_t$, where $\alpha_t, \sigma_t \in \mathbb{R}^+$ are hyperparameters that controls the noising process, and $\mathbf{z}_t = [\mathbf{z}_t^{(x)}, \mathbf{z}_t^{(v)}]$, $t = 0, 1, \ldots, T$ is a sequence of the concatenated latent representation of the positions and the latent representation of the atom attributes. The conditional reverse process starts from a noise distribution $p(\mathbf{z}_T \mid y) = p(\mathbf{z}_T)$, and learns to recover data distribution with $p_\theta([\mathbf{x}, \mathbf{v}] \mid y) = \int p(\mathbf{z}_T)\, p_\theta([\mathbf{x}, \mathbf{v}], \mathbf{z}_{0:T-1} \mid \mathbf{z}_T, y)\, \mathrm{d}\mathbf{z}_{0:T}$, with the reverse denoising process $p_\theta([\mathbf{x}, \mathbf{v}], \mathbf{z}_{0:T-1} \mid \mathbf{z}_T, y) = p_\theta([\mathbf{x}, \mathbf{v}] \mid \mathbf{z}_0, y) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, y)$. The conditional distribution $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, y)$ can be parametrized in terms of a noise prediction model, where a neural network $\boldsymbol{\varepsilon}_\theta$ is trained to predict the Gaussian noise $\boldsymbol{\varepsilon}$ with $\mathbf{z}_t$ (Kingma et al., 2021) with the tth step Kullback–Leibler (KL) divergence loss ℓ_t, defined as $\ell_t := \mathrm{KL}\big(q(\mathbf{z}_{t-1} \mid \mathbf{z}_t, (\mathbf{x}, \mathbf{v}), y) \,\|\, p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, y)\big) = \mathbb{E}_{\boldsymbol{\varepsilon}}\big[\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\mathbf{z}_t, y)\|^2\big]$.

Equivariant molecule generative models As molecules are naturally represented as graphs, where atoms are nodes and chemical bonds are edges, Graph Neural Networks (GNNs) become an ideal choice to equip these generative models for modeling the relationships between atoms and the overall topology of the molecule. Furthermore, the remarkable physical properties that the molecular systems possess can be readily introduced into the generative model as an inductive bias for improved learning performance. Equivariant Graph Neural Networks (GNNs) that preserve permutation symmetry and respect $\mathrm{SE}(3)$ symmetries have been developed to model $\boldsymbol{\varepsilon}_\theta(\mathbf{z}_t, y)$ and shown great improvement over the generic GNNs (Köhler et al., 2020; Satorras et al., 2021; Xu et al., 2022; Guan et al., 2023).

Classifier and classifier-free guidance with unlabeled data In practice, it is often the case that the available labeled dataset (with y known) is small because of the high computational cost of calculating properties. On the other hand, unlabeled data (with y unknown) can often be gathered with a much lower cost are usually and are more accessible. To make use of the abundant unlabeled data, classifier guidance (CG) (Dhariwal & Nichol, 2021) or classifier-free guidance (CFG) (Ho & Salimans, 2022) can be used in combination with generative models (Zeni et al., 2023). In particular, CG is a technique used in generative models to steer the generation process toward desired classes or conditions by using an auxiliary classifier $p(y \mid \mathbf{z}_t)$ trained on the unlabeled samples, whereas CFG allows the generative model to be trained to handle both conditional and unconditional generation without relying on an auxiliary classifier. It applies a guidance factor ω to the conditional distribution $p(\mathbf{z}_t \mid y)$, such that $p_\omega(\mathbf{z}_t \mid y) \propto p(y \mid \mathbf{z}_t)^\omega p(\mathbf{z}_t) \propto p(\mathbf{z}_t \mid y)^\omega p(\mathbf{z}_t)^{1-\omega}$ is used instead of $p(\mathbf{z}_t)$ when evaluating the model score during the reverse process in the conditional setting. In the sampling stage, given the target property value y, instead of $\varepsilon_\theta(\mathbf{z}_t, y)$, CFG computes the tth step noise variable $\boldsymbol{\varepsilon}_t$

$$\boldsymbol{\varepsilon}_t = (1 + \omega)\, \boldsymbol{\varepsilon}_\theta(\mathbf{z}_t, y) - \omega\, \boldsymbol{\varepsilon}(\mathbf{z}_t). \tag{1}$$

In this paper, we use diffusion models with classifier-free guidance with $\omega = 5$ to utilize the unlabeled data samples.

1.1.2. Conditional generative models for molecular strings

The string representation enables the application of language modeling techniques to the field of cheminformatics (M. Bran et al., 2024). By treating molecular sequences similarly to natural language texts, one can leverage Natural Language Processing (NLP) models, such as transformers, to understand and generate new molecular structures (Born & Manica, 2023; Bagal et al., 2022; Frey et al., 2023).

For unconditional molecule generation, transformer-based models for molecule language modeling factorize the probability of the molecule $p(M) = q(s_1, \ldots, s_n)$ into a product of conditional probabilities $q(s_1, \ldots, s_n) = q(s_1) \prod_{i=2}^{n} q(s_i \mid s_1, \ldots, s_{i-1})$ (Frey et al., 2023). There are various ways to adapt unconditional molecule generation for conditional generation. Born & Manica (2023) proposed converting numerical property values into tokens; the input for transformers is then defined by a concatenation of property tokens and textual tokens. This approach leverages existing tokenization and embedding mechanisms, simplifying implementation. With pretrained embeddings, numerical tokens might benefit from rich contextual information derived from large corpora. However, converting numerical values to tokens can lead to a loss of precision, and pretrained embeddings might not effectively capture the nuances of numerical values, as they are primarily trained on text data. The loss of precision and contextual irrelevance can be especially problematic for tail extrapolation.

To capture the precise nature of numerical values, one can augment the unconditional large language model with an additional encoder that maps property conditions to con-

tinuous embeddings. The resulting embeddings are concatenated at the start of the sequence of token embeddings (Bagal et al., 2022). Such a separate encoder can capture the precise nature of numerical values, allowing for finer granularity and better handling of continuous data. Conceptually, it models the following conditional probabilities (with the property conditioning through an encoder), $p(s_1, \ldots, s_n \mid y) = p(s_1 \mid y) \prod_{i=2}^{n} p(s_i \mid s_1, \ldots, s_{i-1}, y)$. The augmented transformer is trained such that the predicted tokens result from attention to both the previous tokens and the conditions. The model is trained by minimizing the cross-entropy loss between the predicted logits and the sequence token embeddings of the molecule string s (Bagal et al., 2022).

1.2. Engression and extrapolation

Shen & Meinshausen (2023) proposed engression, a distributional regression technique for pre-additive noise models (pre-ANMs), to enable extrapolation for nonlinear regression. Under the regression setup where the goal is to model the nonlinear relationship between the covariate X and the response variable Y, the conventional regression model assumes $Y = g(X) + \eta$ where noise is added after applying a nonlinear transformation g, referred to as the post-additive noise model (post-ANM). With engression, the noise is added to the covariates before applying a nonlinear transformation, that is, it assumes a model $Y = g(WX + \eta)$ where W is some linear operation. For simplicity and without loss of generality, we illustrate the idea with W being an identity linear operator.

The training of pre-ANMs is performed by minimizing the energy score-based loss (Gneiting & Raftery, 2007) to find the optimal $\tilde{g}$ with $\tilde{g} \in \arg\min_g \mathbb{E}_X[\mathbb{E}_{Y,\eta}|Y - g(X+\eta)| - \frac{1}{2}\mathbb{E}_{\eta,\eta'}|g(X+\eta) - g(X+\eta')|]$, where η and η' are independent draws from some noise distribution.

The intuition behind the extrapolability lies in the fact that the pre-ANM captures the full distribution around each data point $X + \eta$, including the X-values at the boundary of the training data support. This knowledge about the full distribution in the neighborhood lends extrapolability outside of the training data support. This is in particularly useful in practical applications, where statistical and machine learning models often encounter data points that go beyond the support of the training data.

1.3. Main contributions

In this paper, we demonstrate how engression can be integrated into conditional molecule generation for tail extrapolation. Our main contributions include

(1) We first present engression for distributional extrapolation in the conditional generation $p(x \mid y)$ setup in Section 2.

(2) We introduce the algorithms to integrate engression into conditional molecule generation using classifier-free guided equivariant diffusion models and large language models via pre-ANMs, accordingly, in Section 2.1 and 2.2.

(3) We demonstrate that when there are few labeled and unlabeled samples around the target property values for generating molecules, engression can significantly enhance the performance of baseline conditional generative models in tail extrapolation tasks, particularly in terms of property value accuracy, diversity, and uniqueness. However, when the number of unlabeled samples in the neighborhood increases, the baseline conditional generative models with engression do not benefit as much as those without engression. This highlights the need for developing "engression for unlabeled samples".

2. Engression for distributional extrapolation in conditional molecule generation

Following the idea of engression for nonlinear regression extrapolation introduced in Shen & Meinshausen (2023), we inject noise to the property value y before feeding it into the generative models. That is, instead of modeling $p(x \mid y)$ as $x = g(y) + \eta$, where η denotes some noise distribution, we adopt the pre-ANMs approach, setting $x = g(y + \eta)$. This change enables extrapolation within some radius of δ. Formally, let $\mathcal{F}$ be a distribution class of $y \in \mathbb{R}^d$, and $D_{y'}(P, P') := D(P(x \mid y), P'(x \mid y))$, $\forall y \in \mathcal{Y}$, be a divergence measure of the two distributions $P, P' \in \mathcal{F}$. The distributional extrapolation uncertainty of $\mathcal{F}$ is defined as

$$\mathcal{U}_{\mathcal{F}} := \sup_{y' : \mathrm{d}(y', \mathcal{Y}) \le \delta} \; \sup_{\substack{P, P' \in \mathcal{F} \\ D_y(P, P') = 0, \, \forall y \in \mathcal{Y}}} D_{y'}(P, P'). \quad (2)$$

Proposition 2.1. *Modeling $p(x \mid y)$ with $x = g(y + \eta)$ is distributionally extrapolable within the radius of $\delta > 0$, i.e., distributional extrapolation uncertainty of $\mathcal{F}$, defined as in (2), $U_{\mathcal{F}}(\delta) = 0$.*

2.1. Conditional molecule generation using CFG equivariant diffusion models via pre-ANMs

The most recent equivariant diffusion models for conditional 3D molecule generation require all data samples to be labeled with target property values known Hoogeboom et al. (2022); Guan et al. (2023). This is not often the case given the high computational cost of property value computation. We use classifier-free guidance (Ho & Salimans, 2022) in combination with equivariant diffusion models to learn ε_θ to make full use of unlabeled data. For implementation, we

impute the missing property y value in the unlabeled data with the median value $\bar{y}$. In addition, we also append an NaN indicator δ, which takes value $\delta = 1$ if y is known and 0 otherwise. That is, at each time step t, the conditional diffusion models learn $\boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon}_\theta(\mathbf{z}_t, y, \delta = 0)$ for labeled samples and $\boldsymbol{\varepsilon}_\theta(\mathbf{z}_t, \bar{y}, \delta = 1)$ for unlabeled samples.

Training To train CFG diffusion models via pre-ANMs, we first create two noised copies of the property $y + \eta_j$, $j = 1, 2$, but only for the ones with $\delta = 0$, as we only want to capture the distribution characteristic around valid property values. We propose the following modified CFG loss function for any given sample with property value y: (i) $\ell_\theta = \frac{1}{2}\sum_{j=1}^{2} \|\boldsymbol{\varepsilon}_\theta(\mathbf{z}_t,\, y + \eta_j,\, \delta_j) - \boldsymbol{\varepsilon}\| - \frac{1}{2}\|\boldsymbol{\varepsilon}_\theta(\mathbf{z}_t,\, y + \eta_1,\, \delta_1) - \boldsymbol{\varepsilon}_\theta(\mathbf{z}_t,\, y + \eta_2,\, \delta_2)\|$ if y is unknown ($\delta = 1$); (ii) $\|\boldsymbol{\varepsilon}_\theta(\mathbf{z}_t,\, y,\, \delta) - \boldsymbol{\varepsilon}\|$ otherwise. The loss function introduces the pre-additive noise η to y, and an additional term that matches the conditional distribution given $y+\eta_1$ and $y+\eta_2$, enhancing robustness against the sampled noise term η.

Sampling To sample from target property value y, CFG equivariant diffusion models via pre-ANMs first samples m number of noised property values, $\tilde{y}_j = y + \eta_j$, $j = 1, \ldots, m$, and modifies the computation $\boldsymbol{\varepsilon}_t$ in (1) with these noised samples

$$\tilde{\boldsymbol{\varepsilon}}_t = (1+\omega)\,\frac{1}{m}\sum_{j=1}^{m} \boldsymbol{\varepsilon}_\theta(\mathbf{z}_t,\, \tilde{y}_j,\, \delta = 0) - \omega\, \boldsymbol{\varepsilon}_\theta(\mathbf{z}_t,\, \bar{y},\, \delta = 1).$$

We present the training and sampling algorithms in Appendix B.

2.2. Conditional molecule generation using large language models via pre-ANMs

We extend the unconditional molecule generation model ChemGPT (Frey et al., 2023) for conditional molecule generation. ChemGPT is a generative pre-trained transformer 3 (GPT3)-style model based on GPT-Neo (Brown et al., 2020; Radford et al., 2019) with a tokenizer for SELFIES representations of molecules. During the training, if the property value is unavailable (an unlabeled sample), the molecule strings are mapped to the token embedding $\mathbf{e}_s$ and fed into the transformer ϕ as in the unconditional generation. When the property value y is available (a labeled sample), we use a separate encoder ψ to map y to continuous embedding $\mathbf{e}_y = \psi(y)$. The input of the transformer ϕ now becomes the concatenated embedding $[\mathbf{e}_y; \mathbf{e}_s]$ where the property embedding $\mathbf{e}_y$ is concatenated at the start of the sequence of the token embeddings $\mathbf{e}_s$.

Using the attention mechanism in transformers for conditional molecule generation differs slightly from its use in text generation, where tokens at earlier positions in the sequence cannot "see" tokens at later positions and therefore should not attend to them. For conditional molecule generation with target property values, the attention masks are designed so that each token attends only to the previous tokens and the property condition, whereas the property condition attends to all tokens in the sequence. This approach is different from the conditional generation proposed in Bagal et al. (2022).

To train transformer-based large language models for conditional molecule generation, we further consider property reconstruction loss in addition to the cross-entropy loss on molecule strings, $\ell(s)$. A decoder $\tilde{\psi}$ is used to reconstruct the property values from the last hidden states of the transformers. The training loss is computed as follows: (i) $\ell_\theta(s, y) = \mathrm{CrossEntropy}(\phi(s),\, s)$ for unknown y; (ii) $\ell_\theta(s, y) = \mathrm{CrossEntropy}(\phi([\psi(y); s]),\, s) + \kappa\|\tilde{\psi}(y) - y\|$ for known y, where κ controls the strength of the labeled samples and θ denotes the model parameters for all submodels involved ($\phi, \psi, \tilde{\psi}$).

Training To train transformer-based large language models via pre-ANMs with two noised property values, $\tilde{y}_j = y + \eta_j$, $j = 1, 2$, the training loss is modified to be (i) $\ell_\theta(s, y) = \mathrm{CrossEntropy}(\phi(s),\, s)$ for unknown y; (ii) $\ell_\theta(s, y) = \mathrm{NLL}(\frac{1}{2}\sum_{j=1}^{2}\mathrm{softmax}(\phi([\psi(\tilde{y}_j); s])),\, s) + \frac{1}{2}\sum_{j=1}^{2}\|\tilde{\psi}_\theta(\tilde{y}_j) - y\| - \frac{1}{2}\|\tilde{\psi}_\theta(\tilde{y}_1) - \tilde{\psi}_\theta(\tilde{y}_2)\|$ for known y.

Sampling To sample from transformer-based large language models via pre-ANMs, the generation starts with m noised property values $\tilde{y}_j = y + \eta_j$, $j = 1, \ldots, m$. At each step, given the current generated string s_{curr}, the predicted probabilities for the next tokens are computed by averaging over the results from m noised copies, $\frac{1}{m}\sum_{j=1}^{m}\mathrm{softmax}(\phi([\psi(\tilde{y}_j),\, s_{\mathrm{curr}}])$, from which the next token is sampled.

We present the training and sampling algorithms in Appendix C.

3. Empirical Studies

3.1. Tail extrapolative experimental design on pseudo-labeled QM9

We use QM9 (Ramakrishnan et al., 2014) to evaluate the generative performance for the property LUMO energy (multiplied by 1 Hartree $\approx$ 27.2114 eV) across the models. QM9 contains molecular properties and atom coordinates for 130K small molecules with up to 29 atoms (of type H, C, N, O, F). To evaluate the property values of the 3D molecules generated by the diffusion model, we generate pseudo labels using the property predictor provided in Hoogeboom et al. (2022) on all available samples from QM9 with a train/valid/test split, $0.84/0.15/0.01$.

To evaluate the performance of different methods for tail extrapolative tasks in a more realistic setting, we designed two experiments. We first define a controllable property value region, $\mathcal{D}_C := [1.857, 2.385]$, and vary the number of labeled and unlabeled samples available from $\mathcal{D}_C$. No data samples are accessible beyond this region, $\mathcal{D}_C^+ := (2.385, +\infty)$ (for extrapolation propose). Additionally, 90% of the samples from the region $\mathcal{D}_C^- = (-\infty, 1.857)$ are unlabeled, reflecting common real-world scenarios where labeled data is often scarce. The training data distribution and test data distribution for the two experiment setups are provided in Appendix A.

3.2. Evaluation metrics

Target property value evaluation Target evaluation focuses on assessing how well the generated molecules achieve the desired properties set by the conditional targets. Specifically, it measures the deviation of the properties of generated molecules from the target values, aiming for minimal deviation to ensure high accuracy in meeting the specified conditions.

Vendi score for diversity The second critical metric is diversity, which evaluates the variety among the generated molecules. A diverse set of molecules indicates the model's ability to explore a wide chemical space, reducing redundancy and increasing the likelihood of discovering novel compounds. We use Vendi Score to measure the diversity of the generated molecules (Friedman & Dieng, 2022; Pasarkar & Dieng, 2023). The Vendi Score is defined as the exponential of the Shannon entropy of the eigenvalues λ_i of a similarity matrix K, $\mathrm{VS}(K) = \exp(-\sum_i \lambda_i \log \lambda_i)$. For molecule generation, the similarity function is chosen to be the Morgan fingerprint similarity (radius 3), implemented in RDKit (Landrum, 2016). The higher Vendi score is, the more diverse the generated molecule set is.

Molecule stability We assess the validity of a generated molecule set in terms of a molecule stability metric, which expresses what percentage of a generated molecule set has expected valence behavior (e.g., checking if carbon atoms make four bonds as expected of its' valence behavior). This metric is assessed by computing the percentage of a molecule set that passes successful execution of standard RDKit cheminformatic routines that rely on proper valencies of the molecules, namely using the subroutines `SANITIZE_PROPERTIES` (valency check), `SANITIZE_ADJUSTHS` (implicit hydrogen check), and `SANITIZE_CLEANUPCHIRALITY` (corrects chirality inconsistencies). Lastly, we convert any generated SMILES string into RDKit's canonical SMILES form, which performs an additional valency check internally to the function. The percentage of the molecule set that passes these series of checks represents the molecule stability of the generated set.

Uniqueness We assess the uniqueness of a SMILES batch of generated molecules by converting the batch into a Python-native set variable. Python-native sets are hash tables and remove duplicate entries by nature of their implementation. Uniqueness is calculated by calculating the ratio of the SMILES batch-set size divided by the original SMILES batch size.

Together, these metrics provide a comprehensive evaluation framework, ensuring that the generated molecules are not only accurate and diverse but also valid and unique, thereby enhancing the overall utility and effectiveness of the molecule generation process.

3.3. Results

In this section, we provide empirical studies on the proposed equivariant diffusion models for 3D molecule data and large language models for molecule strings via pre-ANMs. Among all experiments, we set the number of noise variables η to be $m = 100$, and the noise standard deviation to be $\eta = 0.3$. For performance comparison on conditional CFG equivariant diffusion models introduced in Section 2.1, we train the model on pseudo-label QM9 data from scratch for 3000 epochs. For performance comparison on conditional large language models, we fine-tune the constructed pseudo-label QM9 data on the pre-trained unconditional large language model ChemGPT (Frey et al., 2023), with 19 million non-embedding parameters, available on HuggingFace (Frey et al., 2022). We train the conditional language model introduced in Section 2.2 for 100 epochs.

Figure 1 shows the property evaluation of generative molecules from conditional CFG equivariant diffusion models, with and without engression, under Experiment Setup I and Experiment Setup II. In both experiments, engression enhances the performance of conditional CFG equivariant diffusion models by generating molecules with property values closer to the target values and smaller variations.

Table 1. **Composition of training data samples in the designed experiment setups.** The only difference between the two different experimental setups is the number of unlabeled samples in $\mathcal{D}_C^-$ that are available. The number of training samples in $\mathcal{D}_C^+$ is zero.

Setup	Training data composition[1]			
	# Labeled in $\mathcal{D}_C$	# Unabeled in $\mathcal{D}_C$	# Labeled in $\mathcal{D}_C^-$	# Unlabeled in $\mathcal{D}_C^-$
I	1000	1000	10000	90000
II	1000	9000	10000	90000

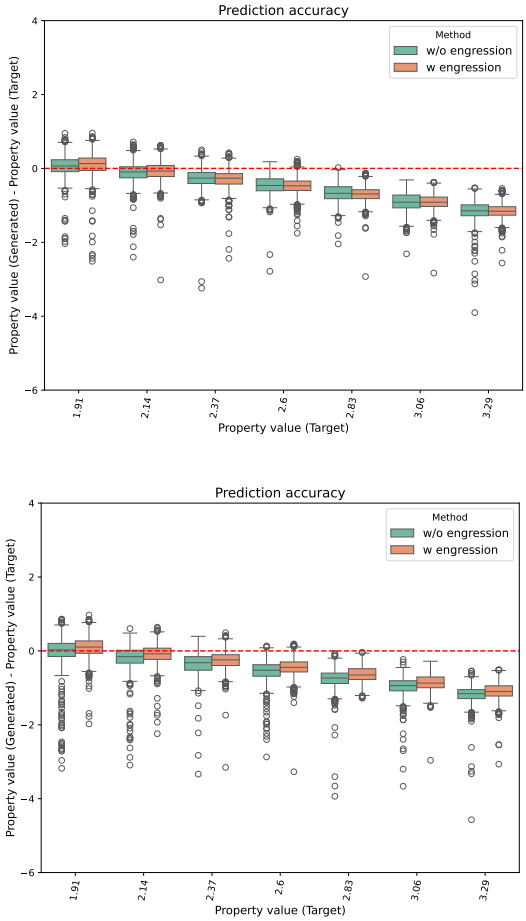

Figure 1. **Property evaluation of the generative molecules from CFG equivariant diffusion models, with and without engression, under Experiment Setup I (Left) and Experimental Setup II (Right), accordingly.** Under Experiment Setup I, around 1000 labeled samples within the property range $\mathcal{D}_C = [1.857, 2.385]$. Under Experiment setup II, around 9000 labeled samples within the property range $\mathcal{D}_C$. In both experiments, engression improves the performance of CFG equivariant diffusion models as it can generate molecules whose property values are closer to the target property values with smaller variations.

We evaluated molecule stability on the raw generated 3D molecular data and assessed diversity using the Vendi Score and Uniqueness on the sanitized generated 3D molecular data. The metric evaluation results for conditional CFG equivariant diffusion models and conditional large language models are presented in Tables 2 and 3, respectively.

The experimental results are generally consistent across the two generative models. In Experiment Setup I, where unlabeled samples are scarce in $\mathcal{D}_C$, engression improves diversity and facilitates extrapolation in the tail region for both diffusion-based models and large language models. However, performance may decline when extrapolative tasks extend further into $\mathcal{D}_C^+$. This aligns with the engression methodology, which injects noise of a certain size to aid extrapolation in the neighborhood around the property values of the available training samples.

In Experiment Setup II, when the number of unlabeled training samples in $\mathcal{D}_C$ increases from 1000 to 9000, the metrics for the baseline generative models without engression show significant improvement compared to those from Experiment Setup I. This demonstrates the effectiveness of generative modeling in learning the latent representation of molecular features. In contrast, the metrics for baseline generative models with engression do not improve as much (for diffusion models in Table 2) or even decline (for large language models in Table 3), resulting in worse performance compared to the baseline models without engression in Experiment Setup II. The increased number of unlabeled samples in $\mathcal{D}_C$ diminishes the effects of engression, where labeled samples are leveraged for tasks in the extrapolative region $\mathcal{D}_C^+$. This calls for developing engression technique for unlabeled samples.

4. Discussion

In this paper, we introduce engression for distributional extrapolation in the conditional molecular generation setup and present algorithms to integrate engression into conditional molecule generation using classifier-free guided equivariant diffusion models and large language models via pre-ANMs. We demonstrate that when there are few labeled and unlabeled samples around the target property values for generating molecules, engression can significantly enhance the performance of baseline conditional generative models in tail extrapolation tasks, particularly in terms of

Table 2. **CFG equivariant diffusion models with engression outperform the models without regression under Experiment Setup I in stability, diversity, and uniqueness metric evaluation, but underperform under Experiment Setup II when there are substantially more unlabeled samples in the region $\mathcal{D}_C$.** The underlined values indicate better comparison performance between the generative models with and without engression.

Experiment Setup I. Around 1000 labeled samples within the property range $\mathcal{D}_C = [1.857, 2.385]$.								
Metric	Method	Target property value						
		1.91	2.14	2.37	2.60	2.83	3.06	3.29
Molecule Stability	w/o engression	98.40	98.70	98.65	98.75	98.75	98.85	98.80
	w engression	97.25	98.30	97.55	96.90	87.25	86.35	80.35
Vendi Score	w/o engression	402.86	371.95	322.97	309.62	290.27	302.77	313.24
	w engression	520.44	469.35	375.54	299.04	216.28	179.95	168.31
Uniqueness	w/o engression	64.75	62.40	57.15	56.60	52.60	54.15	55.80
	w engression	89.45	84.30	75.25	66.10	52.70	48.30	42.30

Experiment Setup II. Around 9000 unlabeled samples within the property range $\mathcal{D}_C = [1.857, 2.385]$.								
Metric	Method	Target property value						
		1.91	2.14	2.37	2.60	2.83	3.06	3.29
Molecule Stability	w/o engression	98.25	98.10	98.55	98.40	98.75	98.45	98.25
	w engression	97.50	97.45	97.95	97.10	96.40	93.80	91.80
Vendi Score	w/o engression	554.58	517.40	472.07	425.95	386.78	342.32	295.83
	w engression	518.95	473.14	391.09	320.61	265.57	245.29	238.14
Uniqueness	w/o engression	90.55	88.25	83.80	78.15	72.40	66.40	61.05
	w engression	89.55	85.20	75.65	66.95	59.75	55.40	52.50

property value accuracy, diversity, and uniqueness. Note that the performance of engression can be influenced by hyperparameters such as the number of injected noise variables m and the standard deviation of these noise variables η. Therefore, experiments should be conducted to understand the sensitivity of engression's performance to these hyperparameters.

Empirical studies also show that as the number of unlabeled samples in the neighborhood increases, baseline conditional generative models with engression do not benefit as much as those without engression. This highlights the need for developing "engression for unlabeled samples". This observation is consistent across both diffusion-based models and large language models. Future work will focus on two key areas: developing the engression technique for unlabeled samples and integrating engression with a broader range of generative models, such as flow matching-based approaches for molecule generation. This will further demonstrate the effectiveness of the engression concept across various conditional generative models.

The current empirical studies rely on a pretrained property predictor to create pseudo-label (property) datasets and evaluate the generated molecules. The next step is to conduct experiments on real data. Evaluations should be performed using more accurate quantum mechanical approaches, such as density functional theory (DFT).

References

Anstine, D. M. and Isayev, O. Generative models as an emerging paradigm in the chemical sciences. *Journal of the American Chemical Society*, 145(16):8736–8750, 2023.

Bagal, V., Aggarwal, R., Vinod, P. K., and Priyakumar, U. D. Molgpt: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62:2064–2076, 2022.

Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R., and Jensen, K. F. Generative models for molecular discovery: Recent advances and challenges. *WIREs Computational Molecular Science*, 12(5), 2022.

Born, J. and Manica, M. Regression transformer enables concurrent sequence regression and generation for molecular language modelling. *Nature Machine Intelligence*, 5: 432–444, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G.,

Table 3. **Conditional large language models with engression outperform the models without regression under Experiment Setup I in diversity and uniqueness metric evaluation, but underperform under Experiment Setup II when there are substantially more unlabeled samples in the region $\mathcal{D}_C$.** The underlined values indicate better comparison performance between the generative models with and without engression.

Experiment Setup I. Around 1000 labeled samples within the property range $\mathcal{D}_C = [1.857, 2.385]$.

Metric	Method	Target property value						
		1.91	2.14	2.37	2.60	2.83	3.06	3.29
Molecule Stability	w/o engression	88.35	88.74	88.17	87.98	88.27	87.76	86.95
	w engression	87.17	86.04	85.51	83.03	83.28	82.98	81.65
Vendi Score	w/o engression	666.30	663.59	680.60	714.56	760.64	777.68	778.31
	w engression	882.24	885.70	908.77	802.68	795.98	779.48	778.41
Uniqueness	w/o engression	41.32	40.88	41.08	42.15	43.80	44.36	43.40
	w engression	50.28	49.38	51.30	46.43	47.36	47.30	45.94

Experiment Setup II. Around 9000 unlabeled samples within the property range $\mathcal{D}_C = [1.857, 2.385]$.

Metric	Method	Target property value						
		1.91	2.14	2.37	2.60	2.83	3.06	3.29
Molecule Stability	w/o engression	90.16	90.43	90.64	90.73	90.56	90.22	90.81
	w engression	93.38	93.02	93.83	93.63	93.59	92.39	93.14
Vendi Score	w/o engression	777.15	748.79	775.87	733.91	730.88	712.34	710.78
	w engression	680.08	701.04	678.56	657.93	696.48	700.71	751.42
Uniqueness	w/o engression	52.44	50.48	52.07	50.71	49.70	48.98	48.57
	w engression	48.26	49.44	50.47	50.36	49.87	49.96	51.89

Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp, 2017.

Frey, N., Soklaski, R., Axelrod, S., Samsi, S., Gomez-Bombarelli, R., Coley, C., and Gadepally, V. Pre-trained chemgpt-19m model, 2022. URL https://huggingface.co/ncfrey/ChemGPT-19M.

Frey, N. C., Soklaski, R., Axelrod, S., Samsi, S., Gómez-Bombarelli, R., Coley, C. W., and Gadepally, V. Neural scaling of deep chemical models. *Nature Machine Intelligence*, 5:1297–1305, 2023.

Friedman, D. and Dieng, A. B. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680, 2014.

Guan, J., Qian, W. W., Peng, X., Su, Y., Peng, J., and Ma, J. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

Ho, J. and Salimans, T. Classifier-free diffusion guidance, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 6840–6851, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, 2022.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, 2020.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *2021 IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 43, pp. 4217–4228, 2021.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10236–10245, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2022.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In *Neural Information Processing Systems*, 2021.

Köhler, J., Klein, L., and Noé', F. Equivariant flows: Exact likelihood generative learning for symmetric densities. In *37th International Conference on Machine Learning*, 2020.

Krenn, M., Häse, F., Nigam, A., Friederich, P., and Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

Landrum, G. RDKit: Open-source cheminformatics software, 2016. URL https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6: 525–535, 2024.

Menick, J. and Kalchbrenner, N. Generating high fidelity images with subscale pixel networks and multidimensional upscaling, 2018.

Oord, A. V., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pp. 1747–1756, 2016.

Pasarkar, A. P. and Dieng, A. B. Cousins of the vendi score: A family of similarity-based diversity metrics for science and machine learning. *arXiv preprint arXiv:2310.12952*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. In *OpenAI blog*, volume 1, 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1:1–7, 2014.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, pp. 1530–1538. JMLR.org, 2015.

Satorras, V. G., Hoogeboom, E., Fuchs, F. B., Posner, I., and Welling, M. E(n) equivariant normalizing flows. In *Advances in Neural Information Processing Systems*, 2021.

Shen, X. and Meinshausen, N. Engression: Extrapolation for nonlinear regression?, 2023.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 2256–2265, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 11918–11930, 2019.

Song, Y., Sohl-Dickstein, J. N., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

van den Oord, A., Vinyals, O., and kavukcuoglu, k. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Weininger, D. 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, 1988.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. GeoDiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

Zeni, C., Pinsler, R., Zügner, D., Fowler, A., Horton, M., Fu, X., Shysheya, S., Crabbé, J., Sun, L., Smith, J., Tomioka, R., and Xie, T. MatterGen: A generative model for inorganic materials design, 2023.

A. Pseudo-label QM9 data distribution

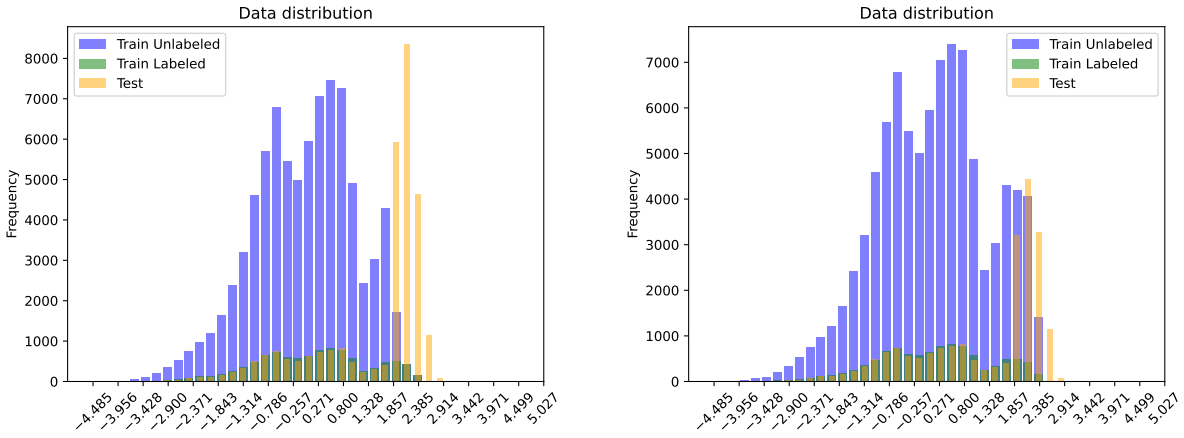

Figure 2. Data distribution for the designed Experiment Setup I (Left) and Experiment Setup II (Right).

B. Algorithms to implement CFD equivariant diffusion models via pre-ANMs

Algorithm 1 Training classifier-free guided equivariant diffusion models via pre-ANMs

input N molecules with target property values $(\mathbf{x}_i, \mathbf{v}_i, y_i)_{i=1}^N$, a neural network ε_θ, noise standard deviation ξ.

repeat
  Sample $t \sim \mathcal{U}(0, 1, \ldots, T)$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
  Sample $\{\mathbf{x}, \mathbf{v}, y\} \sim \{\mathbf{x}_i, \mathbf{v}_i, y_i\}_{i=1}^N$
  Subtract center of mass from $\boldsymbol{\varepsilon}^{(x)}$ in $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}^{(x)}, \boldsymbol{\varepsilon}^{(h)}]$.
  Compute $\mathbf{z}_t = \alpha_t[\mathbf{x}, \mathbf{v}] + \sigma_t \boldsymbol{\varepsilon}$
  if $y == \emptyset$ **then**
    $y \leftarrow \bar{y}$ {Impute the unknown property value}
    $\delta \leftarrow 1$ {Unknown property value indicator set to True}
    $\ell_\theta \leftarrow \|\varepsilon_\theta(\mathbf{z}_t, y, \delta) - \boldsymbol{\varepsilon}\|$
  else
    Sample noise variables $\eta_1, \eta_2 \sim \mathcal{N}(0, \xi)$.
    $\tilde{y}_1, \tilde{y}_2 \leftarrow y + \eta_1, y + \eta_2$ {Create two noised copies}
    $\delta_1, \delta_2 \leftarrow 0, 0$ {Unknown property value indicator set to False}
    $\ell_\theta \leftarrow \frac{1}{2} \sum_j \|\varepsilon_\theta(\mathbf{z}_t, \tilde{y}_j, \delta_j) - \boldsymbol{\varepsilon}\| - \frac{1}{2}\|\varepsilon_\theta(\mathbf{z}_t, \tilde{y}_1, \delta_j) - \varepsilon_\theta(\mathbf{z}_t, \tilde{y}_2, \delta_j)\|$ {Engression training loss}
  end if
  Minimize ℓ_θ and update $\boldsymbol{\theta}$.
until converged

Algorithm 2 Sampling from classifier-free guided equivariant diffusion models via pre-ANMs

input Target value y^*, learned model $\boldsymbol{\varepsilon}_\theta$, guidance strength ω, number of noised copies m, noise standard deviation ξ.
output Generated molecular $(\mathbf{x}, \mathbf{v})$.

Sample $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
for t in $T, \ldots, 2, 1$ **do**
  Sample $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Subtract center of mass from $\boldsymbol{\epsilon}^{(x)}$ in $\boldsymbol{\varepsilon} = [\boldsymbol{\varepsilon}^{(x)}, \boldsymbol{\varepsilon}^{(h)}]$.
  Sample m noise variables $\eta_j \sim \mathcal{N}(0, \xi)$, $j = 1, \ldots, m$.
  for j in $1 \cdots, m$ **do**
    $\tilde{y}_j \leftarrow y^* + \eta_j$ {Create m noised copies}
  end for
  $\tilde{\boldsymbol{\varepsilon}}_t = (1+\omega)\frac{1}{m}\sum_j \boldsymbol{\varepsilon}_\theta(\mathbf{z}_t, \tilde{y}_j, \delta = 0) - \omega\,\boldsymbol{\varepsilon}_\theta(\mathbf{z}_t, \bar{y}, \delta = 1)$ {Classifier free sampling}
  $\tilde{\mathbf{z}}_t = (\mathbf{z}_t - \sigma_t \tilde{\boldsymbol{\varepsilon}}_t) \,/\, \alpha_t$
  if $t > 1$ **then**
    $\mathbf{z}_{t-1} = \tilde{\boldsymbol{\mu}}_{t-1|t}(\mathbf{z}_t, \tilde{\mathbf{z}}_t) + (\tilde{\sigma}_{t-1|t})^{1-\nu}(\sigma_{t|t-1})^\nu\,\boldsymbol{\varepsilon}$
  else
    $\mathbf{z}_{t-1} = \tilde{\mathbf{z}}_t$
  end if
end for
return $[\mathbf{x}, \mathbf{v}] \sim p([\mathbf{x}, \mathbf{v}] \mid \mathbf{z}_0)$.

C. Algorithms to implement conditional large language models via pre-ANMs

Algorithm 3 Training large language models via pre-ANMs

input N molecule strings with target property values $(s_i, y_i)_{i=1}^N$, transformer $\boldsymbol{\phi}_\theta$ and property encoder $\boldsymbol{\psi}_\theta$, noise standard deviation ξ, strength of unlabeled samples κ.

repeat
  Sample $\{s, y\} \sim \{s_i, y_i\}_{i=1}^N$
  Convert s into token embeddings $\mathbf{e}_s$.
  if $y == \emptyset$ **then**
    $\mathbf{u} \leftarrow \boldsymbol{\phi}_\theta(\mathbf{e}_s)$ {Compute logits for next token prediction via transformers}
    $\ell_\theta \leftarrow \kappa\,\mathrm{CrossEntropy}(\mathbf{u}[:-1], s[1:])$ {Next token prediction loss}
  else
    Sample noise variables $\eta_1, \eta_2 \sim \mathcal{N}(0, \xi)$.
    for j in $1, 2$ **do**
      $\tilde{y}_j \leftarrow y + \eta_j$ {Create noised copy}
      $\mathbf{e}_{y,j} \leftarrow \boldsymbol{\psi}_\theta(y_j)$ {Construct property embeddings}
      $\mathbf{e}_{s,j} \leftarrow [\mathbf{e}_s; \mathbf{e}_{y,j}]$ {Concatenate property embedding with the string embedding}
      $\mathbf{p}_j \leftarrow \mathrm{softmax}(\boldsymbol{\phi}_\theta(\mathbf{e}_{s,j}))$ {Compute probabilities for next token prediction via transformers}
    end for
    $\mathbf{p} = \frac{1}{2}(\mathbf{p}_1 + \mathbf{p}_2)$
    $\ell_\theta \leftarrow \mathrm{NLL}(\mathbf{p}[:-1], s[1:]) + \frac{1}{2}\sum_{j=1}^2 \|\tilde{\psi}_\theta(\tilde{y}_j) - y\| - \frac{1}{2}\|\tilde{\psi}_\theta(\tilde{y}_1) - \tilde{\psi}_\theta(\tilde{y}_2)\|$ {Next token prediction loss}
  end if
  Minimize ℓ_θ and update $\boldsymbol{\theta}$.
until converged

Algorithm 4 Sampling from large language models (LLM) via pre-ANMs

input Target value y^*, learned transformer $\boldsymbol{\phi}_\theta$ and property encoder $\boldsymbol{\psi}_\theta$, temperature T, number of noised copies m, noise standard deviation ξ, max length L.

output Generated molecular string s.

Sample m noise variables $\eta_j \sim \mathcal{N}(0,\, \xi)$, $j = 1, \ldots, m$.
for j in $1, \cdots, m$ **do**
  $\tilde{y}_j \leftarrow y^* + \eta_j$ {Create m noised copies}
  $\mathbf{e}_{\mathrm{start},j} \leftarrow \boldsymbol{\psi}_\theta(\tilde{y}_j)$ {Construct property embeddings}
end for
Set token s_0 to be CLS token.
$s \leftarrow [s_0]$ {Start generating strings}
for t in $1, \cdots, L$ **do**
  if s_0 is EOS token **then**
    break {Signal to end generation}
  end if
  Convert s_0 into token embedding $\mathbf{e}_{\mathrm{next}}$.
  for j in $1, \cdots, m$ **do**
    $\mathbf{e}_{\mathrm{start},j} \leftarrow [\mathbf{e}_{\mathrm{start},j}; \mathbf{e}_{\mathrm{next}}]$
    $\mathbf{u}_j \leftarrow \boldsymbol{\phi}_\theta(\mathbf{e}_{\mathrm{start},j})$ {Predicted logits for next token}
    $\mathbf{p}_j \leftarrow \mathrm{softmax}(\frac{\mathbf{u}_j}{T})$ {Convert to probabilities with temperature scaling}
  end for
  $\mathbf{p} \leftarrow \frac{1}{m} \sum_j \mathbf{p}_j$
  Sample the next token s_0 from the probability distribution $\mathbf{p}$.
  $s \leftarrow [s, s_0]$
end for
return s.