ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

Towards Efficient Molecular Property Optimization with Graph Energy Based Models

Luca Miglior[1], Lorenzo Simone[1], Marco Podda[1] and Davide Bacciu[1] *

1 - University of Pisa - Department of Computer Science
Largo Bruno Pontecorvo 3, Pisa, Italy

Abstract. Optimizing chemical properties is a challenging task due to the vastness and complexity of chemical space. Here, we present a generative energy-based architecture for implicit chemical property optimization, designed to efficiently generate molecules that satisfy target properties without explicit conditional generation. We use Graph Energy Based Models and a training approach that does not require property labels. We validated our approach on well-established chemical benchmarks, showing superior results to state-of-the-art methods and demonstrating robustness and efficiency towards *de novo* drug design.

1 Introduction

Generating molecules that satisfy specific chemical constraints is crucial in modern *de novo* drug design. Traditional generative approaches for property optimization generally involve training deep learning models on large datasets as general-purpose molecule generators. Prior works explored property optimization using Reinforcement Learning (RL) methods to guide molecular generation, such as GraphDF [7] and GraphAF [8], or Bayesian Optimization (BO) on the latent space as proposed in JT-VAE [4]. MoFlow [10] uses Classifier-Based (CB) guidance to direct the generation towards specific objectives. GraphEBM [6] tackles molecular generation with energy-based models (EBM), but the results of property optimization remain unclear, as it restricts to optimizing numerically quantifiable properties and exhibits high variance in generation, which affects consistency and reliability of generated molecules. Despite their advancements, the methodologies referenced above present several limitations: RL methods are known to be unstable and computationally demanding, while BO and CB methods rely on supervised learning and require large labeled datasets to work properly, which are rarely available in the chemical domain.

Motivated by the need of overcoming existing limitations in optimization methods, especially in real-world scenarios where labeled data is scarce and computational efficiency is crucial, we present a novel Unspervised Energy-based Molecule Optimization (UEMO) capable of implicitly learning chemical properties and efficiently generating molecular graphs without the need of labeled data or explicit optimization strategies. Our method leverages the capabilities of Graph Neural Networks (GNNs) [1] to process molecular graphs and utilizes Langevin dynamics [2] at sampling time to generate new molecules. We tested

*This work is partially supported by the Next Generation EU programme under project FAIR (PE00000013), Spoke 1

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

our approach on ZINC-250K [3], evaluating its performance in optimizing QED and LogP chemical properties, and providing results on the constrained optimization of existing compounds.

2 Methodology

Whilst most models are trained as general-purpose generators, with property-specific optimization treated as a separate post hoc task, we exceed such limitations by proposing a novel method that implicitly embeds a strong bias toward the desired objective directly in the generative model, eliminating the need for supervision. Given a molecular input space $\mathcal{X}$, let $\mathcal{S} \subseteq \mathcal{X}$ the subset of molecules that satisfy given property constraints. The goal is to train the generative model that maximizes the probability of generating samples coming from $\mathcal{S}$. Such a result can be pursued by resorting to domain-specific datasets guaranteed to reflect the targeted property (e.g., binding affinity to a specific property), or by employing off-the-shelf property predictors to filter molecules that satisfy desired constraints from existing datasets. This concept-based approach is versatile and applicable to any desired chemical objective. In contrast to baseline methods – GraphEBM [6] in particular, whose optimization strategy is limited by numerical property estimates to scale the energy term – we provide a generalizable setting that allows for the optimization of arbitrary properties without requiring explicit values or labeled data. This is particularly advantageous when dealing with complex or non-numerical targets, such as binding affinity or toxicity.

Let $g = (\mathbf{A}, \mathbf{X})$ be a molecular graph, consisting of n atoms and m atom types, where $\mathbf{A} \in \mathbb{R}^{n \times n \times (e+1)}$ is the adjacency tensor describing chemical bonds[1], and $\mathbf{X} \in \mathbb{R}^{n \times m}$ is the one-hot-encoded atom representation. The energy function is defined as a function $E_\theta(g) : \mathcal{X} \to \mathbb{R}$ with learnable parameters θ that assigns a scalar value to each molecule in the input space. The probability distribution over the input space is given by:

$$p_\theta(g) = \frac{1}{Z} \exp(-E_\theta(g)),$$

where $Z = \int_{\mathcal{X}} \exp(-E_\theta(g))\, dg$ is the intractable marginalization term. In this work, we model the energy function E_θ using a Graph Convolutional Network (GCN) [5]. The GCN consists of L message-passing layers, after which a graph-level embedding $\mathbf{z} \in \mathbb{R}^{d_L}$ is computed by applying a mean-pooling operation on the L-th node representations as follows: $\mathbf{z} = \text{MeanPooling}(\mathbf{H}^{(L)})$. Finally, the energy value is obtained by passing $\mathbf{z}$ through J Fully Connected (FC) layers:

$$E_\theta(g) = \text{EnergyHead}(\mathbf{z}) = FC_J(FC_{J-1}(\dots FC_1(\mathbf{z})))$$

The schematic representation of the proposed architecture is shown in Figure 1. Given the energy $E_\theta(g) \in \mathbb{R}$, generating samples from the associated probability distribution is not trivial due to the intractability of the marginalization term.

[1] e is the number of bond types, with an additional type to indicate the absence of a bond.

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

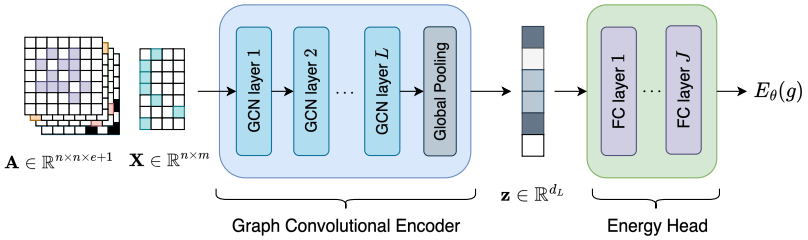

Fig. 1: UEMO architecture. In the example, $n = 7, m = 4$, and $e = 3$.

In this work, we use Langevin dynamics which directly uses the gradient of the energy function to generate samples:

$$g_{i+1} = g_i - \frac{\epsilon}{2}\nabla_g E_\theta(g_i) + \omega_i,\ \omega_i \sim \mathcal{N}(0, \epsilon),$$

where ϵ is the step size. For $i \to \infty$ and $\epsilon \to 0$, this process is guaranteed to generate samples from the true distribution p_θ. UEMO is trained by modeling the energy of the space of the molecules $\mathcal{X}$ through the distribution implicitly defined by E_θ. This is achieved by minimizing the negative log likelihood $\mathcal{L}(\theta) = -\mathbb{E}_{g\sim\mathcal{S}}[\log p_\theta(g)]$, under the empirical distribution of the dataset $\mathcal{S}$. This objective is known to have the formulation:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{g^+\sim\mathcal{S}}\left[\nabla_\theta E_\theta(g^+)\right] - \mathbb{E}_{g^-\sim p_\theta}\left[\nabla_\theta E_\theta(g^-)\right],$$

here we rely on Langevin dynamics to generate negative samples $g^- \sim p_\theta$. Initial sampling hyperparameters for Langevin dynamics were chosen according to [2, Appendix A.11]. As initial configuration for the chain $g_1, \dots g_i$, we leverage Persistent Contrastive Divergence (PCD) [9], by implementing a limited-size replay buffer $\mathcal{B}$ to store samples generated in previous iterations: in our implementation 95% of the initializations are drawn from $\mathcal{B}$ and 5% from standard Gaussian noise. New molecules are sampled from the trained model by initializing the Langevin dynamics with molecular graphs g drawn from Gaussian noise, such that $\mathbf{A}_0, \mathbf{X}_0 \sim \mathcal{N}(0, 1)$. At each step of the sampling chain, the generated adjacency tensor is symmetrized to ensure consistency and used as initialization for the next step, following $\mathbf{A}_{i+1} = \frac{1}{2}(\mathbf{A}_i + \mathbf{A}_i^T)$. Additionally, we compute the target property of the intermediate molecule at every step of the chain to maintain stability and prevent the local representation from collapsing.

3 Experiments

We evaluated UEMO in two different property optimization tasks: Quantitative Drug Likeliness (QED) and Octanol-Water Partition Coefficient (LogP). To perform these experiments, we used the publicly available dataset ZINC-250k, which consists of 250,000 chemical compounds annotated with the above-mentioned

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

properties. To facilitate implicit optimization, we divided the dataset in two partitions, each one reflecting a desired property constraint. Specifically, our partition matched high QED score (≥ 0.75) and high LogP coefficient (≥ 3) constraints. The partitioning was chosen to reflect real-world scenarios, where datasets targeting specific molecular properties are often small. The final partitions consisted of 54k and 34k molecules for QED and LogP, respectively. In addition to property evaluation, we performed benchmarks on molecular graph generation using three widely adopted metrics: validity, novelty, and uniqueness. These metrics are computed on 5,000 randomly generated molecules. We compared our methodology against five strong baselines [4, 6–8, 10] by running their original source code and generating graphs with their property optimization strategies to ensure a fair comparison. JT-VAE results are from [7]. Firstly, we report results for the QED optimization task. We trained UEMO on the high-QED partition of the dataset and we randomly sampled a batch of molecules. Quantitative results are presented in Table 1.

Model	**Validity**	**Uniqueness**	**Novelty**	**Avg. QED ↑**	**Top 1 ↑**
GraphDF	100%	100%	100%	0.53 ± 0.28	**0.948**
GraphAF	100%	100%	100%	0.52 ± 0.17	**0.948**
GraphEBM	100%	83.1% ± 2.7%	100%	0.44 ± 0.11	0.781
MoFlow	100%	96.8% ± 1.6%	100%	0.59 ± 0.23	**0.948**
JT-VAE	100%	100%	100%	n.a.	0.925
UEMO	**100%**	**100%**	**100%**	**0.79 ± 0.12**	**0.948**
				Avg. LogP ↑	**Top 1 ↑**
GraphDF	100%	100%	100%	9.13 ± 1.00	14.12
GraphAF	100%	100%	100%	8.15 ± 3.90	**14.82**
GraphEBM	100%	82.6% ± 2.8%	100%	-1.50 ± 2.34	3.29
MoFlow	100%	95.4% ± 1.6%	100%	-2.30 ± 5.37	3.93
JT-VAE	100%	100%	100%	n.a.	n.a.
UEMO	**100%**	**100%**	**100%**	**9.30 ± 2.13**	12.66

Table 1: Metrics for molecules generated in both optimization tasks.

Our model achieves perfect validity, uniqueness, and novelty ratios, matching the performance of all tested models except GraphEBM, which exhibits a substantially lower uniqueness ratio. Furthermore, our model significantly outperforms all other baselines with an average QED score of $\mathbf{0.79 \pm 0.12}$, demonstrating that our implicit approach is more effective at sampling various and complex molecular configurations from the learned energy landscape. Additionally, it shows lower variance compared to the baselines, highlighting the robustness of our method in directing the generative process towards optimizing the desired chemical property. As illustrated in the right pane of Figure 2, the Kernel Density Estimation (KDE) plot for the QED property of the generated molecules aligns closely with that of the training partition. This alignment confirms our model's ability to sample challenging molecules from a small, specific region of chemical space. The plot further emphasizes the stability of the generative process, validating the reliability of our implicit optimization strategy. We then report LogP optimization results in Table 1 (bottom). Our

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

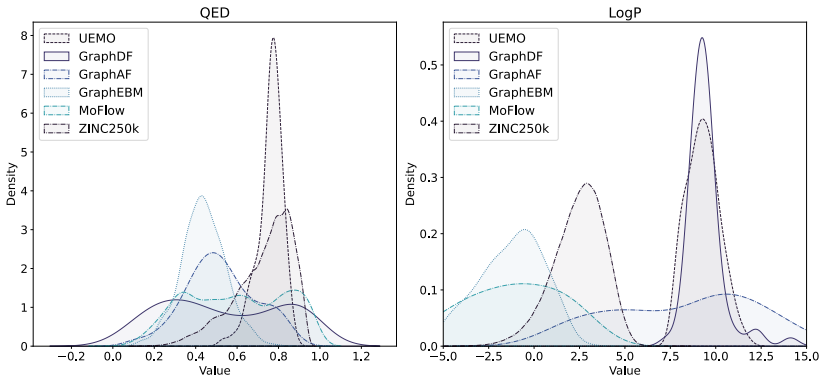

Fig. 2: QED (left) and LogP (right) distributions of the molecules generated by the different methods compared to the full, not partitioned, ZINC-250k dataset.

experiments demonstrate that our model achieves performance comparable to state-of-the-art autoregressive strategies [7, 8], generating molecules with an average LogP of $\mathbf{9.30 \pm 2.13}$. Moreover, UEMO outperforms one-shot generative methods [4, 6, 10], which either failed the task or produced molecules that did not satisfy the required constraints. Although maximizing LogP alone does not have direct practical applications and is considered a relatively straightforward task (since high LogP values can be trivially achieved by adding long chains of carbon atoms), success in this task highlights the flexibility of our proposed methodology across diverse and distinct generative tasks, effectively capturing chemical information from small and domain-specific datasets. This result can be visualized in Figure 2, where the majority of molecules generated by UEMO exhibit high LogP values with low variance, emphasizing the robustness of our approach. A visual representation of UEMO's generated molecules is presented in Figure 3 for the QED optimization task.

While achieving state-of-the-art performance in property optimization tasks such as QED and LogP, UEMO also stands out for its remarkable sampling efficiency. Unlike baseline strategies, which require significant computational resources, our model demonstrates a highly efficient generative process. Notably, UEMO is able to generate molecules at an average rate of $\mathbf{14.6 \pm 1.2s/100mol}$. We provide a comprehensive comparison of sampling times in Table 2. This

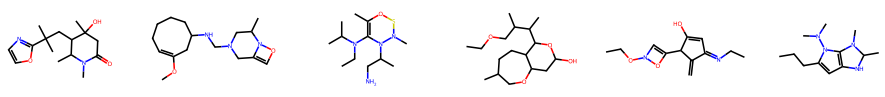

Fig. 3: Curated samples of molecules generated in the QED optimization task.

ESANN 2025 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 23-25 April 2025, i6doc.com publ., ISBN 9782875870933. Available from http://www.i6doc.com/en/.

UEMO	GraphDF	GraphAF	GraphEBM	MoFlow	JT-VAE
14.6 ±1.2	174 ± 6.1	41.5 ± 5.3	212.1 ± 4.1	366 ± 3.7	n.a.

Table 2: Time taken (in seconds) to generate 100 molecules.

substantial efficiency arises from two key factors. First, our implicit optimization strategy eliminates the need for expensive sampling strategies, which are employed by other methods. Second, UEMO architecture is exceptionally lightweight, consisting in 0.25M trainable parameters, which is orders of magnitude smaller than many state-of-the-art models. This efficiency not only reduces computational overhead, but also makes UEMO particularly well suited for applications that require a rapid and focused exploration of the chemical space.

4 Conclusions

In this paper we presented UEMO, a novel EBM for molecular generation and chemical property optimization. Our implicit optimization approach demonstrated strong performance, outperforming existing models. Moreover, our lightweight model and innovative approach showed outstanding sampling efficiency, significantly reducing graph generation times. In the future, we plan to extend this work to larger and drug-specific datasets targeting diverse properties to generate highly specific graphs. Additionally, we plan to develop training methodologies that incorporate chemical knowledge into the energy landscape, ensuring a *chemically informed* generative process that adheres to chemical rules.

References

[1] D. Bacciu, F. Errica, A. Micheli, and M. Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 2020.

[2] Y. Du and I. Mordatch. Implicit Generation and Modeling with Energy Based Models. In *NeurIPS 2019*, pages 3603–3613, 2019.

[3] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman. ZINC: A free tool to discover chemistry for biology. 52(7):1757–1768.

[4] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *ICML 2019*, pages 2323–2332. PMLR, 2018.

[5] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017*. PMLR, 2017.

[6] M. Liu, K. Yan, B. Oztekin, and S. Ji. Graphebm: Molecular graph generation with energy-based models, 2021.

[7] Y. Luo, K. Yan, and S. Ji. Graphdf: A discrete flow model for molecular graph generation. In *ICML 2021*, volume 139. PMLR, 2021.

[8] C. Shi*, M. Xu*, Z. Zhu, W. Zhang, M. Zhang, and J. Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In *ICLR*, 2020.

[9] T. Tieleman and G. E. Hinton. Using fast weights to improve persistent contrastive divergence. In *ICML 2009*, volume 382, pages 1033–1040.

[10] C. Zang and F. Wang. Moflow: An invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD*, KDD '20, pages 617–626, 2020.