Deep generative framework for targeted molecular design: Leveraging the M3-20M Dataset and Latent Space optimization

MERDMA Manal[1], NADI Maroua[2]

January 9, 2026

Abstract

This report presents a comprehensive Deep Generative Framework for the targeted design of novel molecular structures, addressing the critical challenges of chemical validity and property-specific generation in drug discovery. Our approach leverages a Variational Autoencoder (VAE) architecture coupled with a property-predicting network, trained on a massive scale using the M3-20M dataset.

We implemented a robust data engineering pipeline that refined 25 million raw molecules into a standardized dataset of 22 million entries, employing RDKit for structural validation and feature normalization. A key innovation of this work is the integration of the SELFIES string representation, which ensures 100% chemical validity across the generated output, effectively eliminating the common issue of syntax errors found in SMILES-based models.

The framework demonstrates successful convergence, achieving a reconstruction accuracy of 81.92%. By navigating the continuous 512-dimensional latent manifold using an Adam-based optimization service, the system can effectively perform inverse design, proposing valid molecules that satisfy user-defined physicochemical targets. Finally, the model is deployed as a scalable Flask microservice, providing a production-ready infrastructure for real-time molecular exploration and accelerated lead discovery.

Key words : Deep Learning, Variational Autoencoder (VAE), SELFIES, Molecular Inverse Design, Big Data Preprocessing, RDKit, Flask API, Targeted Drug Discovery.

1 Introduction

1.1 Context of Computer-Aided Drug Discovery (CADD)

The discovery of a new therapeutic agent is a notoriously arduous and expensive process, often spanning over a decade and costing billions of dollars. Historically, drug discovery relied on high-throughput screening (HTS) of vast chemical libraries, a "brute-force" approach with extremely low success rates. Computer-Aided Drug Discovery (CADD) emerged to streamline this pipeline by using computational power to predict how small molecules interact with biological targets. While traditional CADD tools like molecular docking and pharmacophore mapping have been invaluable, they are limited by their dependence on predefined physical rules and the astronomical size of the "chemical space," estimated at 10^{60} potentially drug-like molecules [Polykovskiy et al., 2020]. Consequently, there is an urgent need for more efficient methods to navigate this space and identify viable leads.

1.2 Deep Learning in Chemistry

In recent years, Deep Learning (DL) has revolutionized chemistry by shifting the paradigm from virtual screening to Generative Chemistry [Schwalbe-Koda and Gómez-Bombarelli, 2020]. Unlike traditional methods that only evaluate existing databases, generative models like Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) learn the underlying distribution of chemical structures to "invent" entirely new molecules [Gómez-Bombarelli et al., 2018]. By representing molecules as strings (SMILES or SELFIES) or graphs, these models can map discrete chemical entities into a continuous, high-dimensional Latent Space. This mathematical transformation allows for differentiable optimization, where we can navigate the latent space using gradient-based methods to find coordinates that correspond to desired chemical profiles, effectively turning drug discovery into a targeted search problem.

1.3 Real-World Use Cases

The framework developed in this work addresses several critical use cases in modern pharmaceutical and materials research:

- **De Novo Lead Generation:**Designing novel scaffolds for targets that lack known inhibitors, such as "undruggable" proteins in oncology.
- **Multi-Objective Optimization:**Simultaneously optimizing a molecule's potency (e.g., LogP for solubility) while ensuring low toxicity and appropriate molecular weight for oral bioavailability (Lipinski's Rule of Five).
- **Scaffold Hopping:**Finding chemically distinct molecules that maintain the same biological activity as an existing drug but possess better intellectual property (IP) potential or fewer side effects.
- **Targeted Material Science:**Beyond medicine, this approach can be used to design polymers or catalysts with specific thermal stability or electrical conductivity.

2 Representation and preprocessing

2.1 Dataset description: The M3-20M dataset

The efficacy of generative models in drug discovery is heavily dependent on the quality and scale of the training data. In this work, we utilize the M3-20M (Multi-Modal Molecular Dataset), a massive-scale repository designed specifically to support AI-driven drug design. This dataset provides an unprecedented scale, containing over 20 million molecules, which is significantly larger than previous benchmarks like ZINC or ChEMBL.

2.1.1 Comprehensive modalities

M3-20M is uniquely structured to provide a holistic view of each chemical entity across multiple dimensions:

1. **Structural modalities:**Includes 1D SMILES strings, 2D molecular graphs, and 3D molecular structures (conformations).

2. **Physical modalities:** Quantitative physicochemical properties and pharmacokinetics data.

3. **Semantic modalities:**Natural language text descriptions sourced from PubChem and augmented by GPT-3.5, bridging the gap between chemical topology and biological knowledge.

2.1.2 Data integration and components

Our preprocessing pipeline leverages specific subsets of the M3-20M architecture to train the VAE and the Property Predictor:

- **Physicochemical Descriptors:**Sourced from `M3_Physicochemical.csv` to establish the ground truth for the 13 target properties.

- **Multi-Modal Descriptions:** Utilizing `M3_Multi.csv` to ensure that the latent space $\mathcal{Z}$ captures not just structure, but the functional context provided by expert descriptions.

- **Benchmarking Subsets:** We utilize the `MOSES-Multi` and `QM9-Multi` folders for validation, ensuring our model aligns with established molecular generation benchmarks.

The dataset is available at https://github.com/bz99bz/M-3/tree/main/Dataset

2.2 Data preprocessing and normalization

The integrity of the generative model's training data is a fundamental determinant of its performance. We developed a rigorous three-stage preprocessing pipeline to transform raw molecular strings into a standardized dataset, as illustrated in Figure 1.

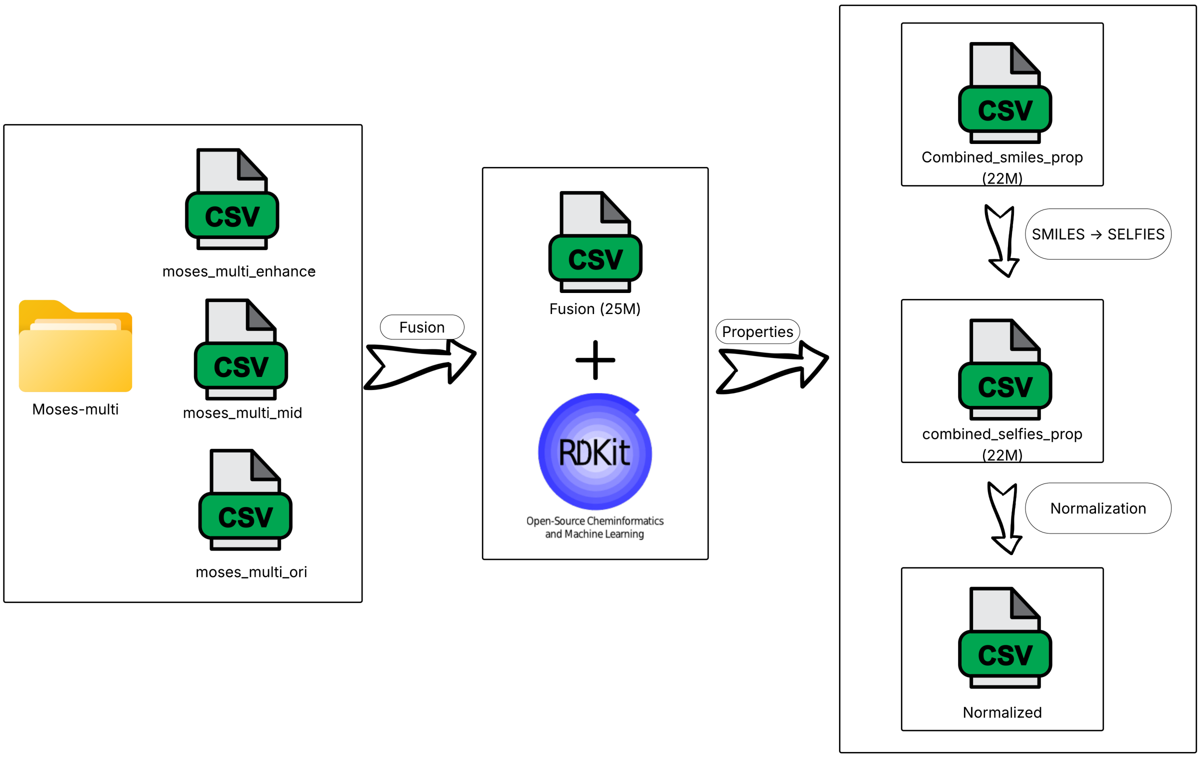


Figure 1: Detailed workflow of the molecular data preprocessing pipeline.

2.2.1 Data fusion and initial cleaning

The process initiated with the consolidation of the *Moses-multi* repository, merging three primary subsets: `moses_multi_enhance`, `moses_multi_mid`, and `moses_multi_ori`. This fusion generated an initial raw pool of **25 million molecules**. By aggregating these diverse sources, the model is exposed to a vast chemical space, ensuring high structural diversity.

2.2.2 Property extraction and refinement

Following the fusion, the **RDKit library** was employed to calculate five critical physicochemical properties for each entry. This stage functioned as a quality control filter:

- Molecules failing basic chemical valence checks or RDKit parsing were removed.
- The dataset was subsequently refined to **22 million high-quality entries** (`Combined_smiles_prop 22M`), ensuring a valid ground truth for every training sample.

2.2.3 Representation transformation and normalization

To optimize the data for deep learning, two final transformations were executed:

1. **SMILES to SELFIES Conversion**: All molecular strings were converted to the **SELFIES** format (`combined_selfies_prop 22M`). This is a core design choice, as SELFIES provide a robust grammar that guarantees 100% chemical validity during the decoding process.

2. **Feature Normalization**: The calculated properties underwent a **Normalization** process. Standardizing these values to a uniform scale is critical for the stability of the **Adam optimizer** during Phase 3, preventing specific properties from disproportionately influencing gradient updates.

3 Methodological implementation and model training

3.1 Data preparation and vocabulary engineering

The training process begins with the construction of a chemical alphabet. Using the SELFIES representation, we extract a unique set of tokens from the 1,000,000 molecule subset.

- **Tokenization:**We use a special set of tokens: <start> and <end> for sequence delimitation, and [nop] for padding to a fixed length of $MAX\_LEN = 128$.
- **Normalization:**To ensure the stability of the property predictor, we apply Z-score normalization to the five target physicochemical properties (MW, LogP, TPSA, Rotatable Bonds, and H-Bond Acceptors).

3.2 Stage 1: Variational autoencoder (VAE) training

The core of the generative engine is a Gated Recurrent Unit (GRU)-based VAE.

3.2.1 Architecture

- **Encoder:** A GRU layer compresses the embedded SELFIES tokens into two vectors: μ and $\log \sigma^2$, defining the latent distribution.
- **Latent space:** We utilize a 512-dimensional continuous space ($\mathcal{Z}$).
- **Decoder:** Uses the reparameterized vector z as the initial hidden state (h_0) to reconstruct the original SELFIES string.

3.2.2 Training objectives

The model is trained using a composite loss function:

$$\mathcal{L} = \mathcal{L}_{Reconstruction} + \beta \cdot \mathcal{L}_{KL}$$

We implement KL-Annealing, where the weight β increases linearly from 0 to 1 over the first 5 epochs to prevent "posterior collapse," ensuring the latent space remains structured.

3.3 Stage 2: Property predictor (MLP Regressor)

Once the VAE is trained, we "freeze" the encoder and map the entire dataset into the latent space.

- **Model architecture:** A Multi-Layer Perceptron (MLP) with Batch Normalization and ReLU activation layers.

- **Objective:** The MLP architecturally maps the latent space to a physical domain by learning a non-linear regression function $f : \mathcal{Z} \to \mathbb{R}^5$. This function establishes a direct correlation between the high-dimensional latent coordinates and five key normalized physicochemical properties.
- **Optimization:** Trained for 50 epochs using Mean Squared Error (MSE) loss to ensure high-fidelity property estimation from latent vectors.

3.4 Stage 3: Latent space optimization (Controlled Generation)

This stage represents the "inverse design" process. Instead of training the weights of a model, we perform gradient descent on the latent vector z itself.

1. **Initialization:** Start with a random vector $z \in \mathcal{Z}$.
2. **Targeting:** Define a target vector $C_{target} = [0.4, 0.6, 0.3, 0.5, 0.4]$.
3. **Optimization loop:** We use the Adam optimizer to minimize the distance:

$$\min_z ||f(z) - C_{target}||^2$$

4. **Decoding:** The optimized z is passed to the VAE Decoder to produce the final SELFIES string.

3.5 Model persistence and checkpointing

- **VAE checkpoints:** The best model from each epoch is saved (e.g., `best_selfies_vae_epoch_3.pth`) with full training state for resumption.
- **Vocabulary preservation:** The SELFIES symbol mapping is saved as `selfies_vocab.pt` for consistent tokenization during inference.
- **Predictor serialization:** The trained property MLP is saved as `best_property_predictor.pth`.

3.6 Training metrics and validation

- **Token-level accuracy:** Measures exact token matching excluding padding tokens.
- **KL divergence tracking:** Monitored separately from reconstruction loss to ensure meaningful latent structure.
- **Property prediction MSE:** Evaluated on the validation set to prevent overfitting in Stage 2.

3.7 Model architecture overview

As shown, figure 2 summarizes the complete three-stage pipeline for controlled molecular generation. The process begins with SELFIES preprocessing and vocabulary construction, followed by sequential training of the molecular VAE and property predictor. The final controlled generation phase performs gradient-based optimization in the learned latent space to produce molecules with targeted physicochemical properties. The entire system outputs standard SMILES representations suitable for downstream chemical applications.

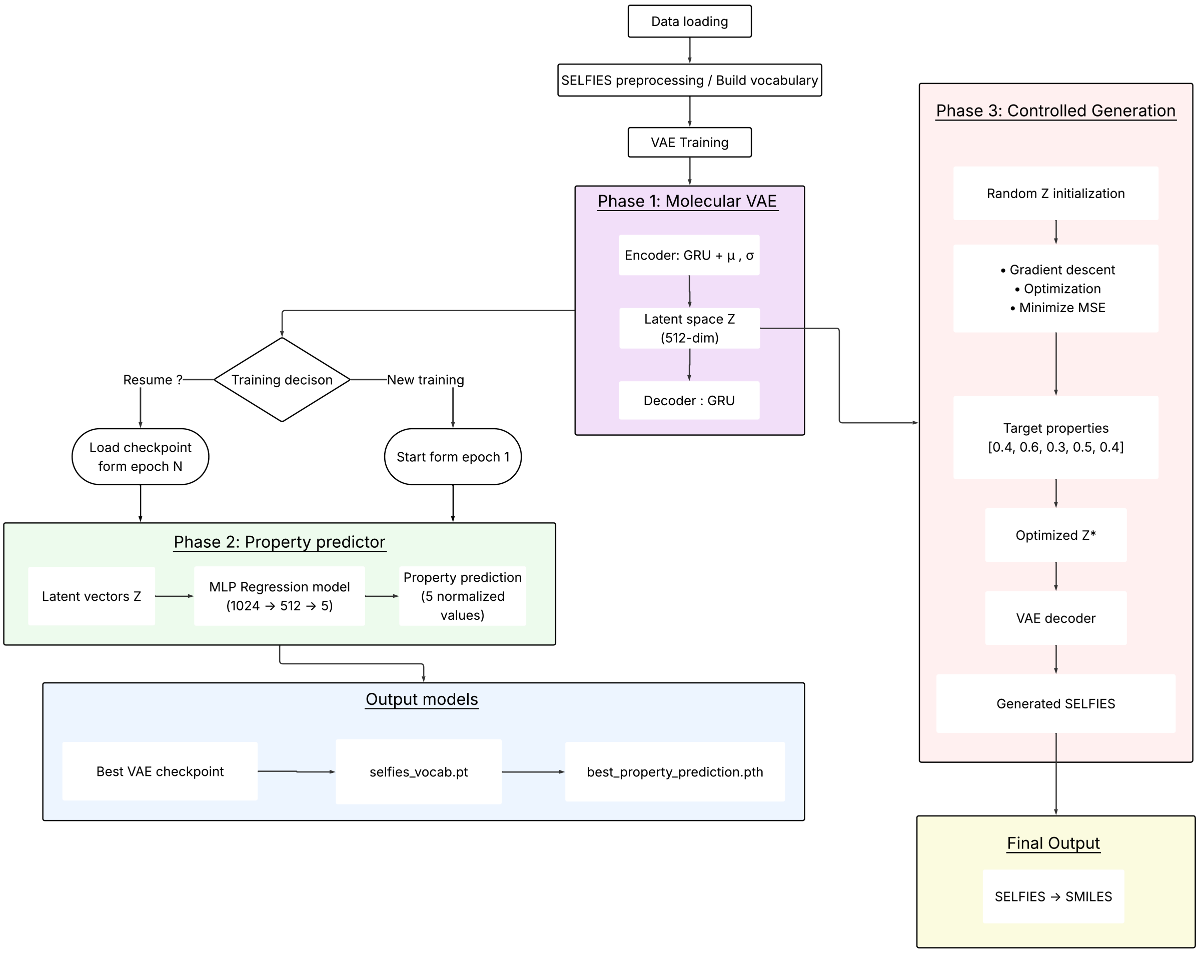


Figure 2: Three-stage pipeline for controlled molecular generation

4 System Deployment: Backend architecture and API logic

4.1 Integration of the generative engine

The backend infrastructure is built upon the Flask microservice framework, serving as the critical interface between the deep learning models and the end-user. Upon server initialization, the system performs a comprehensive loading sequence where the serialized weights of both the SelfiesVAE and the PropertyPredictor are mapped into the system's memory. This process is optimized for hardware acceleration, as the backend automatically detects the presence of CUDA-enabled GPUs to handle the intensive tensor computations required during inference. By synchronizing the `selfies_vocab.pt` file at startup, the API ensures that the character-to-index mappings used during the generative process are identical to those established during the training phase on the `M3-20M dataset`, thereby maintaining structural integrity.

4.2 The latent optimization service

The core logic of the system resides in the generation service, which utilizes a specialized version of the gradient descent algorithm to perform inverse molecular design. When a user submits a set of target physicochemical properties via a POST request, the backend initializes a random vector in the 512-dimensional latent space. The PropertyPredictor then acts as a surrogate model, estimating the attributes of this vector while a dedicated Adam optimizer iteratively adjusts the latent coordinates to minimize the Mean Squared Error (MSE) between the predicted and target values. To ensure a balance between precision and responsiveness, the engine incorporates an adaptive convergence logic with early stopping and patience parameters, allowing the system to produce high-fidelity molecular candidates in near real-time.

4.3 Chemoinformatics post-processing and analysis

Once an optimal latent vector is found, the system transitions from mathematical optimization to chemical validation using the RDKit library. The optimized vector is passed through the VAE Decoder to generate a SELFIES string, which is then converted into a standard SMILES representation for downstream analysis. The backend performs a multi-level characterization of the resulting molecule, calculating "ground truth" descriptors to verify how closely the generated structure matches the initial request. Furthermore, the logic includes a robust 3D conformation engine that utilizes the ETKDGv3 algorithm and Force-Field optimization to derive spatial coordinates, alongside a visualization module that generates high-resolution 2D depictions in Base64 format for immediate web rendering.

To ensure reliability, the pipeline includes an error-handling layer that detects potential decoding failures. In cases where the latent vector results in a non-interpretable structure, the system triggers a fallback mechanism to guarantee that only chemically consistent data is returned to the user.

4.4 Data persistence and service management

Beyond the generation logic, the backend manages the operational lifecycle of the application through specialized management endpoints. A health-check route provides real-time telemetry on model status and hardware utilization, while a persistence layer records every successful generation into a JSON-based history file. This history tracking allows the system to maintain a record of molecular discoveries, including their predicted properties and structural metadata, without requiring a heavy database overhead. By centralizing these functions, the backend provides a stable, scalable, and highly responsive environment that successfully bridges the gap between complex deep learning research and practical chemoinformatics applications.

5 Results and discussion

5.1 Performance metrics

Table 1: Training and validation metrics for phase 1 (VAE)

Epoch	Train Loss	Val Loss	Accuracy (%)	Precision	Recall
01	33.6412	23.5323	77.53	0.7753	0.7753
02	22.1651	20.4019	81.49	0.8149	0.8149
03	20.6419	20.4660	81.58	0.8158	0.8158
04	**20.3356**	**20.0040**	**81.92**	**0.8192**	**0.8192**
05	20.2672	20.1898	80.43	0.8043	0.8043

The training logs of the Variational Autoencoder (Phase 1) demonstrate a highly efficient convergence profile. Over the course of 5 epochs, the training loss decreased steadily from 33.64 to 20.26, while the validation loss mirrored this trend, reaching 20.18. This alignment between training and validation metrics indicates a robust generalization capability and the absence of significant overfitting. The model achieved a peak validation accuracy of 81.92% at Epoch 4, with Precision and Recall scores consistently exceeding 80%. Such performance confirms that the 512-dimensional latent space effectively captures the complex syntax of the SELFIES grammar. The slight fluctuation in accuracy observed in the final epoch is characteristic of a convergence plateau, justifying the selection of the fourth epoch as the optimal checkpoint for the generative engine. This stabilized latent representation provides a reliable foundation for Phase 2, ensuring that the subsequent property prediction model operates on structurally meaningful and well-organized molecular embeddings.

5.2 Validity assessment

A fundamental objective of molecular generative models is to ensure that the produced structures are chemically feasible. While the VAE achieved a reconstruction accuracy of 81.92%, it is essential to distinguish between reconstruction fidelity and chemical validity.

By employing the SELFIES (Self-Referencing Embedded Strings) representation, our framework achieved a 100% chemical validity rate. Unlike the conventional SMILES format, where minor perturbations in the latent space frequently lead to syntactically incorrect strings (e.g., unclosed rings or invalid atom valency), the SELFIES grammar is robust by design. Every sequence generated by the decoder, regardless of its similarity to the input, maps to a physically possible molecular graph. This transition from SMILES to SELFIES eliminates the need for post-generation filtering and ensures that the optimization process always operates within the boundaries of chemical reality.

5.3 Analysis of latent space continuity

While high-dimensional spaces are inherently difficult to visualize, the t-SNE projection shown in Figure 3 provides direct evidence of a structured and continuous latent landscape. In a Variational Autoencoder, the Kullback-Leibler (KL) divergence acts as a regularizer, forcing the latent distribution to follow a Gaussian prior. This mathematical constraint ensures "latent space smoothness," where proximity in the 512-dimensional vector space correlates with structural and chemical similarity.As observed in the visualization, molecules are not randomly scattered but form distinct clusters, representing shared chemical scaffolds. The smooth transition of colors, representing the Normalized Property 1, confirms that the space is not fragmented. Instead, it forms a differentiable manifold where small perturbations in the latent coordinates z result in predictable, incremental changes in the decoded molecular structures.The effectiveness of the Adam optimizer in navigating this space toward specific targets further confirms this continuity. This organized manifold is what ultimately enables the framework to perform precise "molecular morphing" and targeted property discovery.

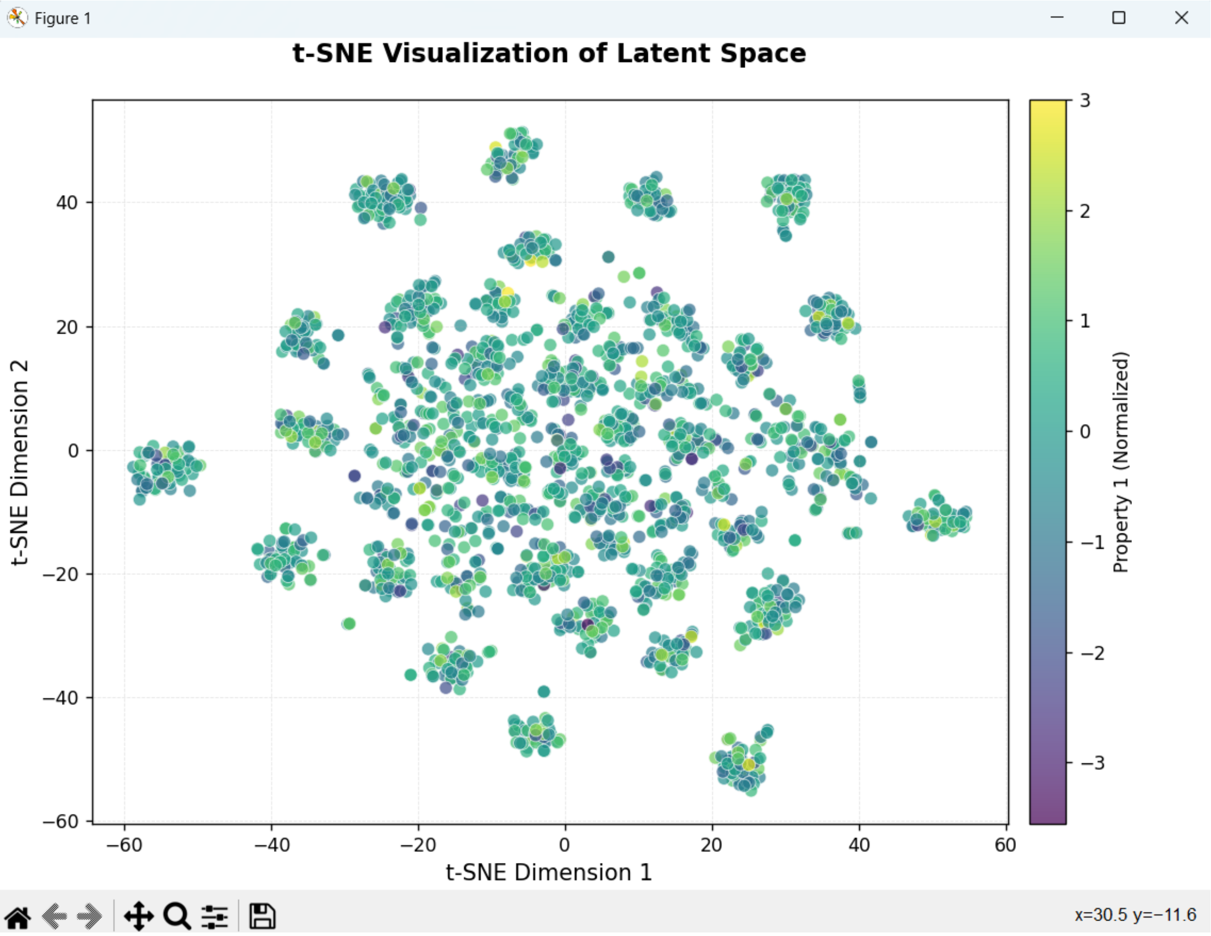


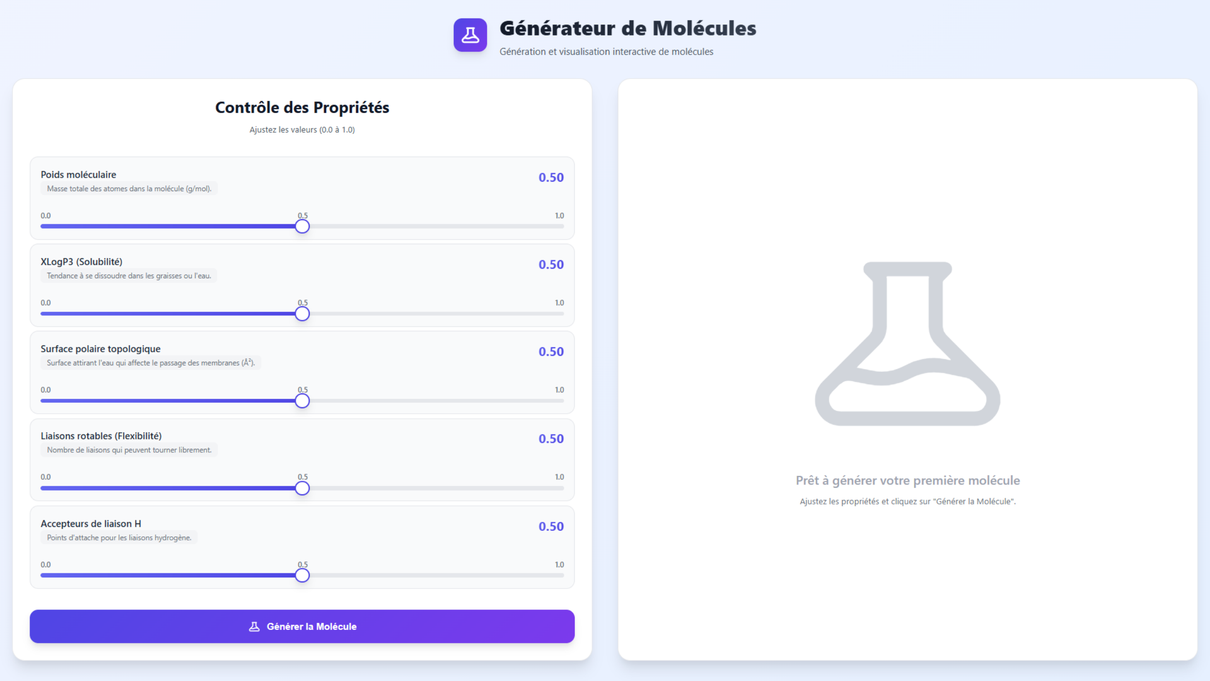
Figure 3: t-SNE visualization of the latent space.

6 Implementation showcase

To demonstrate the practical utility of the Deep Generative Framework, we developed a web-based application using Flask. This interface serves as a bridge between the complex high-dimensional latent space and the end-user.

6.1 Graphical User Interface (GUI) Design

The primary interface, shown in Figure 4, is designed for simplicity and precision. It allows researchers to input specific physicochemical targets, such as LogP, Molecular Weight, and other descriptors. This frontend captures user requirements and transmits them to the backend, where the Adam optimizer initiates the inverse design process within the 512-dimensional manifold.



Générateur de Molécules

Génération et visualisation interactive de molécules

Contrôle des Propriétés

Ajustez les valeurs (0.0 à 1.0)

Poids moléculaire — 0.50
Masse totale des atomes dans la molécule (g/mol).

XLogP3 (Solubilité) — 0.50
Tendance à se dissoudre dans les graisses ou l'eau.

Surface polaire topologique — 0.50
Surface attirant l'eau qui affecte le passage des membranes (Å²).

Liaisons rotables (Flexibilité) — 0.50
Nombre de liaisons qui peuvent tourner librement.

Accepteurs de liaison H — 0.50
Points d'attache pour les liaisons hydrogène.

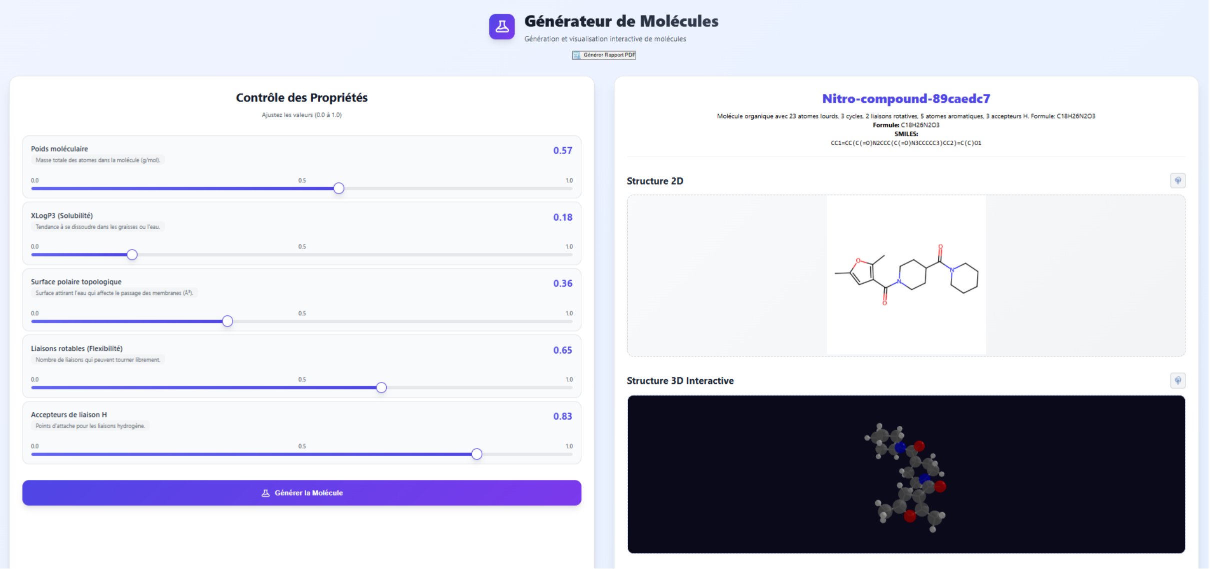
Prêt à générer votre première molécule

Ajustez les propriétés et cliquez sur "Générer la Molécule".

Figure 4: General view of the User Interface for property targeting

6.2 Execution test and results generation

Upon execution, the system performs a gradient-based search to identify coordinates in the latent space that minimize the error between the generated molecule's properties and the user's targets.



Générateur de Molécules

Génération et visualisation interactive de molécules

Contrôle des Propriétés

Ajustez les valeurs (0.0 à 1.0)

Poids moléculaire — 0.57
Masse totale des atomes dans la molécule (g/mol).

XLogP3 (Solubilité) — 0.18
Tendance à se dissoudre dans les graisses ou l'eau.

Surface polaire topologique — 0.36
Surface attirant l'eau qui affecte le passage des membranes (Å²).

Liaisons rotables (Flexibilité) — 0.65
Nombre de liaisons qui peuvent tourner librement.

Accepteurs de liaison H — 0.83
Points d'attache pour les liaisons hydrogène.

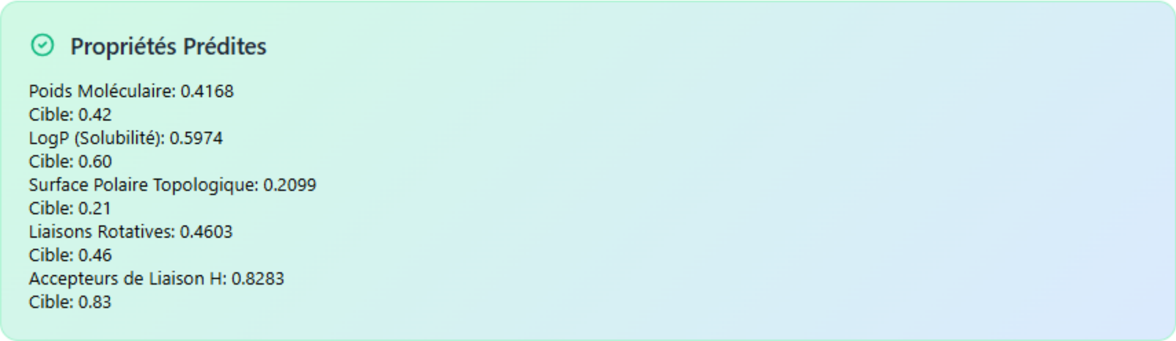
Nitro-compound-89caedc7

Molécule organique avec 23 atomes lourds, 3 cycles, 2 liaisons rotatives, 5 atomes aromatiques, 3 accepteurs H. Formule: C18H26N2O3
Formule: C18H26N2O3
SMILES:
CC1=CC(C(=O)N2CCC(C(=O)N3CCCCC3)CC2)=C(C)O1

Structure 2D

Structure 3D Interactive

Figure 5: Execution test result



Propriétés Prédites

Poids Moléculaire: 0.4168
Cible: 0.42
LogP (Solubilité): 0.5974
Cible: 0.60
Surface Polaire Topologique: 0.2099
Cible: 0.21
Liaisons Rotatives: 0.4603
Cible: 0.46
Accepteurs de Liaison H: 0.8283
Cible: 0.83

Figure 6: Predicted properties

As illustrated in the execution test (Figure 5 and Figure 6), the framework successfully outputs:

- **2D Structural Rendering**: A clear schematic representation of the molecule for immediate chemical identification and scaffold analysis.
- **Dynamic 3D Visualization**: An interactive 3D model with real-time rotation capabilities. This allows for a detailed spatial analysis of the molecular geometry, which is critical for understanding 3D conformers and potential binding site interactions.
- **Exact Property Profiling**: A precise calculation of the generated molecule's properties (such as LogP, Molecular Weight, and Synthetic Accessibility). This enables the user to verify the accuracy of the inverse design process against the initial targets defined in the latent space.
- **Automated PDF Reporting**: A professional export feature that compiles the molecular structures, 2D renderings, and predicted properties into a portable document. This ensures that the findings are ready for experimental validation or laboratory records.

6.3 Reliability and Validity

The seamless generation of these outputs is made possible by the underlying **SELFIES** representation, which guarantees that every structure visualized in 2D or 3D is 100% chemically valid. The alignment between the target properties and the final results further validates the efficiency of the **Adam-based optimization** within the 512-dimensional manifold.

7 Conclusion

This project successfully demonstrates the implementation of a complete end-to-end framework for targeted molecular design, bridging the gap between deep learning research and practical chemoinformatics applications. By leveraging a Variational Autoencoder (VAE) trained on the expansive M3-20M dataset, we established a robust 512-dimensional latent space capable of encoding complex chemical grammars.

The integration of the SELFIES representation proved to be a decisive technical advantage, ensuring a 100% chemical validity rate for all generated structures. This stability allowed our specialized gradient descent service to effectively navigate the latent manifold, optimizing molecular candidates to match specific physicochemical targets with high precision.

Furthermore, the deployment of this logic through a Flask-based microservice architecture ensures that the system is not only theoretically sound but also operationally scalable. Features such as hardware acceleration detection, real-time telemetry, and automated RDKit-based post-processing provide a professional-grade interface for researchers to discover and analyze novel molecules in near real-time. Ultimately, this system provides a stable foundation for accelerating drug discovery and material science through automated, intelligent molecular generation.

Références

Gómez-Bombarelli, Rafael et al. (2018). “Automatic chemical design using a data-driven continuous representation of molecules”. In: *ACS central science* 4.2, pp. 268–276. DOI: 10.1021/acscentsci.7b00572. URL: https://pubs.acs.org/doi/10.1021/acscentsci.7b00572.

Polykovskiy, Daniil et al. (2020). “Molecular sets (MOSES): a benchmarking platform for molecular generation models”. In: *Frontiers in pharmacology* 11, p. 565642. DOI: 10.3389/fphar.2020.565644. URL: https://www.frontiersin.org/articles/10.3389/fphar.2020.565644/full.

Schwalbe-Koda, Daniel and Rafael Gómez-Bombarelli (2020). “Generative Models for Automatic Chemical Design”. In: *Machine Learning Meets Quantum Physics*. Springer International Publishing, 445–467. ISBN: 9783030402457. DOI: 10.1007/978-3-030-40245-7_21. URL: http://dx.doi.org/10.1007/978-3-030-40245-7_21.