JOINT FOUNDATION MODEL FOR SYNTHESIZABLE MOLECULAR GENERATION AND PROPERTY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular foundation models deliver strong property prediction from limited labels but produce encoders that cannot propose new molecules, while latent-variable generators are trained for reconstruction rather than transfer and decode unreliably away from their data. Design needs a third ingredient neither supplies: assurance that what is proposed can actually be synthesized. The model presented here, that is a generative version of the foundation model CheMeleon, puts all three on one representation. It pairs the descriptor-supervised CheMeleon D-MPNN with an autoregressive transformer decoder, optimizing Mordred descriptor regression and SMILES reconstruction jointly at foundation scale, so that a single 2048-dimensional embedding serves as property representation, generative latent and search domain. A rectified flow fit to cached embeddings replaces the intractable Gaussian prior and doubles as a projection operator that keeps latent search decodable. Synthesizability enters the same space rather than a post hoc filter, as a differentiable head distilled from a retrosynthesis planner and, in a second arm, as hard projection onto routed analogs. Invertibility costs 0.022 RMSE on property prediction across 30 MoleculeACE assays. The generative capabilities, tested on the GuacaMol benchmark, are competitive to state of the art models for both unconditional and goal-oriented generation tasks. Driven by property-head gradients under Langevin dynamics, the model proposes novel, potent, synthesizable candidates on 30 MoleculeACE assays, each with an explicit synthetic route.

1 INTRODUCTION

Drug discovery increasingly relies on foundation models pretrained on broad chemical corpora and adapted with limited labels (Bommasani et al., 2021). The dominant recipe is self-supervision over unlabeled structures, via masked-token objectives on strings (ChemBERTa (Chithrananda et al., 2020), MoLFormer (Ross et al., 2022)) or graph-level contrastive and context-prediction objectives (GROVER (Rong et al., 2020), MolCLR (Wang et al., 2022), and the strategies of Hu et al. (2020)); a complementary line substitutes supervised pretraining on cheap labels, whether bioassay collections (Klaser et al., 2024) or quantum targets (Zhou et al., 2023). All produce an encoder, optimized for discrimination rather than reconstruction, and so cannot propose molecules. CheMeleon exemplifies the supervised route, pretraining a depth-6 directed message-passing network (D-MPNN) (Yang et al., 2019; Heid et al., 2024; Graff et al., 2026) to regress thousands of deterministic Mordred descriptors (Moriwaki et al., 2018) from the graph. Because that signal is noise-free by construction, it avoids the systematic error and sparsity limiting assay-derived supervision, and its 2048-dimensional embedding transfers strongly to the 30 MoleculeACE benchmarks (van Tilborg et al., 2022; Burns et al., 2025). Suggestively, Burns et al. (2025) report that this representation places similar molecules close together, frustrating activity-cliff discrimination but shaping the geometry as a generative latent space wants. Can it be made decodable without sacrificing that predictive signal?

Medicinal chemistry needs generation. Chemical language models emit SMILES or SELFIES autoregressively (Segler et al., 2018; Krenn et al., 2020), scaled in MolGPT (Bagal et al., 2022), Chemformer (Irwin et al., 2022) and SAFE-GPT (Noutahi et al., 2024), but expose no property-aligned

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

latent space; latent-variable models pair encoder with decoder for continuous optimization (Gómez-Bombarelli et al., 2018; Jin et al., 2018; Reidenbach et al., 2022); discrete search and RL act directly on structures (Olivecrona et al., 2017; Blaschke et al., 2020; Jensen, 2019; Nigam et al., 2020; 2022) and remain strong baselines under fixed search budgets (Gao et al., 2022a), as measured on GuacaMol (Brown et al., 2019) and MOSES (Polykovskiy et al., 2020). Latent-variable methods are the natural contact point with a predictive foundation model, and a substantial literature optimizes in such spaces by constrained or weighted retraining, trust-region and joint surrogate–generator methods, inverse design, and flow-based traversal (Griffiths & Hernández-Lobato, 2020; Tripp et al., 2020; Maus et al., 2022; Stanton et al., 2022; Eckmann et al., 2022; Wei et al., 2024; Lipman et al., 2023). The recurring failure is a mismatch between where the optimizer explores and where the decoder can invert. Gómez-Bombarelli et al. (2018) report SMILES validity ranging from 70% to under 1% when decoding latent points, and Griffiths & Hernández-Lobato (2020) show that unconstrained Bayesian optimization in that latent space "tends to produce invalid molecular structures," recovering validity only by constraining the search; weighted retraining addresses the same mismatch from the data side (Tripp et al., 2020). The difficulty is not that encoder and decoder are trained separately, but that a latent optimized for reconstruction alone is neither a validated property representation nor reliably invertible away from the data it saw. Separately, models readily exploit scoring functions with unsynthesizable structures (Gao & Coley, 2020); practitioners either score synthesizability post hoc (Ertl & Schuffenhauer, 2009; Coley et al., 2018; Genheden et al., 2020; Thakkar et al., 2021) or *enforce* it by generating pathways (Gao et al., 2022b) or projecting onto synthesizable space (Luo et al., 2024; Luo & Coley, 2025b).

In this work we propose Generative CheMeleon, a generative counterpart of the CheMeleon foundation model. We use CheMeleon as an encoder and we pair it with an autoregressive transformer decoder aiming at reconstructing SMILES from the embedding. The two objectives are trained together on up to 4M molecules. We argue that keeping the Mordred regression loss of the original model is what lets the latent space retain its predictive character while acquiring the expressivity a generator needs, so that a single 2048-dimensional embedding, i.e. the CheMeleon latent space, can serve as property representation, generative latent and search domain at once. We then target the prior mismatch that this width creates by fitting a rectified flow to cached embeddings, which supplies both a sampler and a projection operator that keeps latent search decodable. Finally, we embed synthesizability directly in the generation loop, either as a differentiable head distilled from a retrosynthesis planner or as a hard projection onto routed analogs, rather than as a filter applied afterwards. Section §2 develops the model: architecture, the joint pretraining and adaptation scheme, and the unconditional and conditional generation algorithms. Section §3 evaluates it in each of its roles in turn, as a predictor, an unconditional generator on GuacaMol, a black-box optimizer, and a designer across 30 MoleculeACE assays. Lastly, section §4 concludes.

2 METHOD

2.1 ARCHITECTURE: CHEMELEON ENCODER AND MEMORY-PROJECTED DECODER

We inherit the CheMeleon encoder unchanged (Burns et al., 2025): SMILES are featurized and passed to a BondMessagePassing D-MPNN (Yang et al., 2019; Heid et al., 2024; Graff et al., 2026) of hidden size 2048 and depth 6, mean-aggregated into $h \in \mathbb{R}^{2048}$, the vector originally consumed by a head regressing $\sim$1,613 Mordred descriptors (Moriwaki et al., 2018). It is exactly this vector we require to be decodable: the CheMeleon embedding is the generative latent space, with no further bottleneck, which keeps property heads, prior and search on the object the predictive model was validated on, at the cost of a latent space one to two orders of magnitude wider than is typical (§2.3).

To invert it we attach an autoregressive transformer decoder (Figure 1a). Since a single wide vector is a poor conditioning interface for attention, a linear memory projection $W_m : \mathbb{R}^{2048} \to \mathbb{R}^{M \times d_{\text{model}}}$ reshapes h into $M{=}8$ pseudo-tokens of width $d_{\text{model}}{=}384$, in the spirit of learned latent arrays (Jaegle et al., 2021). A 6-layer Pre-Layer Norm stack ($\sim$20.6M parameters) applies causal self-attention over generated tokens and cross-attention over the memory, giving $p_\theta(x \mid h) = \prod_i p_\theta(x_i \mid x_{<i}, W_m h)$. Encoder and decoder communicate only through h, so anything producing a 2048-dimensional vector composes with the decoder without retraining it.

2.2 JOINT PREDICTIVE–GENERATIVE PRETRAINING

Rather than fit a decoder to a frozen encoder, we train both jointly so h is shaped simultaneously by descriptor supervision and invertibility:

$$\mathcal{L} = \alpha\, \mathcal{L}_{\text{Mordred}} + \beta\, \mathcal{L}_{\text{recon}}, \tag{1}$$

where $\mathcal{L}_{\text{Mordred}}$ is a random-dropout MSE over Mordred targets (masking undefined or non-finite descriptors) and $\mathcal{L}_{\text{recon}}$ is autoregressive cross-entropy. We set $\alpha{=}\beta{=}1$ and train from a random initialization with both terms active from the first step, so the descriptor loss supplies the pretraining signal rather than acting as a regularizer over inherited weights. The terms are complementary: descriptor regression rewards a smooth, chemically organized embedding, reconstruction an injective one. However, injectivity on the data distribution is not enough: the decoder must stay reliable in a neighbourhood of each embedding if mutations, gradients or prior samples are to decode. We compare two ways of widening it. Stochastic smoothing (Stoch.) conditions the decoder on $\tilde{h} = h + \sigma\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, preserving the geometry descriptor pretraining produced but leaving the prior unspecified, which §2.3 supplies. The variational autoencoder (VAE) treatment instead reads $\mu = h$, predicts $\log \sigma^2$, and adds $\gamma D_{\text{KL}}(q(z \mid x) \,\|\, \mathcal{N}(0, I))$ to Eq. equation 1 (Gómez-Bombarelli et al., 2018; Bowman et al., 2016; Kingma et al., 2016), buying a samplable prior at the cost of constraining that geometry. Orthogonally, we train against canonical SMILES and SELFIES (Krenn et al., 2020), whose grammar guarantees validity, moving that burden from model to representation; SELFIES are converted to SMILES before RDKit evaluation, so all four configurations are comparable.

2.3 SAMPLING THE LATENT SPACE WITH RECTIFIED FLOW MATCHING

Unconditional generation needs a distribution over $\mathbb{R}^{2048}$, and an isotropic Gaussian is a poor surrogate for where an encoder places data, leaving prior holes: regions of high prior mass the decoder never saw, which motivate learned or ex-post priors (Tomczak & Welling, 2018; Dai & Wipf, 2019; Ghosh et al., 2020). Retaining CheMeleon's width worsens this. In high dimension the norm of an isotropic Gaussian vector concentrates sharply about $\sigma\sqrt{d}$, with deviations of order σ independent of d (Vershynin, 2018; Blum et al., 2020), so essentially all prior mass sits on a thin shell whose overlap with the embedding manifold is negligible and whose angular distribution is uninformed by chemistry. We therefore learn the prior, caching embeddings of $\sim$500k pretraining molecules and fitting a rectified flow (Lipman et al., 2023; Liu et al., 2023) that transports noise onto them. With $x_t = (1-t)z_0 + t\,h$, $z_0 \sim \mathcal{N}(0, I)$, $t \sim \mathcal{U}[0, 1]$, we regress

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,h,z_0} \left\| v_\phi(x_t, t) - (h - z_0) \right\|_2^2, \tag{2}$$

and sample by integrating $\dot{x} = v_\phi(x, t)$ in K Euler or midpoint steps; because rectified flows learn nearly straight paths, few steps suffice, which matters when the prior is queried thousands of times per search. Crucially, the same field yields a reusable notion of "on-manifold", a map pulling arbitrary points back toward the decodable region that §2.5 exploits. The flow is fit per encoder checkpoint and frozen thereafter.

2.4 DOWNSTREAM ADAPTATION

We initialize encoder and decoder from a joint checkpoint and attach a fresh head $f_\psi(h)$. Which parameters to update is not merely plasticity versus overfitting, because every generative component is defined relative to the embedding geometry: the decoder inverts vectors from a particular distribution and the flow prior is fit to a sample from it, so moving the encoder moves both targets. Under frozen-backbone adaptation only f_ψ trains, so one prior fit once on cached embeddings is reused across all assays and search gradients live in the coordinates the decoder inverts. Under full fine-tuning everything updates, letting the representation specialize (the regime in which CheMeleon's gains were demonstrated (Burns et al., 2025)); we then retain reconstruction, optimizing $\mathcal{L}_{\text{ft}} = \mathcal{L}_{\text{assay}} + \beta_{\text{ft}}\mathcal{L}_{\text{recon}}$, and refit the prior on the re-encoded corpus before generating (Appendix B). We default to the frozen variant, reserving full fine-tuning for when endpoint fidelity binds; either way §2.5 and §2.6 see the same interface.

2.5 BLACK-BOX GOAL-DIRECTED SEARCH: A FLOW-PROJECTED LATENT GENETIC ALGORITHM

For non-differentiable objectives, such as substructure matches, scaffold hops, MPOs or docking, we also search with the objective as an oracle. We evolve latents $z \in \mathbb{R}^{2048}$ with encoder, decoder and prior frozen (Figure 1b; Algorithm 1 in Appendix A), making the scoring function a genuine plug-in: any SMILES-to-scalar callable substitutes without retraining anything, unlike RL policies that must be re-optimized per objective (Olivecrona et al., 2017; Blaschke et al., 2020). The population initializes from the prior; each generation every latent is decoded, scored and archived by canonical SMILES; the top N_e elites are retained, offspring are mutated copies of elites, and N_r slots refill from the prior. The search budget is $N \times g$, with N being the size of the population and g the number of generations.

The critical step is what happens before an offspring is admitted. Unconstrained mutation in 2048 dimensions leaves the decodable region almost immediately and the decoder emits invalid strings, the failure mode reported throughout the latent-optimization literature (Wei et al., 2024; Griffiths & Hernández-Lobato, 2020; Tripp et al., 2020). We counter with flow projection, an SDEdit-style partial noise-and-denoise pass through the learned prior (Meng et al., 2022): given strength $s \in [0, 1]$ we set $t_0 = 1 - s$, re-noise to $x_{t_0} = t_0 z + (1 - t_0) z_0$ with $z_0 \sim \mathcal{N}(0, I)$, then integrate v_ϕ from t_0 to $t{=}1$ in K steps, giving $\tilde{z} = \textsc{FlowProject}(z; s, K)$. Here $s{=}0$ returns z unchanged (a plain latent GA) and $s{=}1$ discards it for a prior sample; intermediate values retain the elite-directed content while re-anchoring the candidate, so selection pressure and decodability are enforced by separate mechanisms rather than traded inside one loss.

2.6 PROPERTY-DRIVEN GENERATION WITH LANGEVIN DYNAMICS

The core setting is designing against a learned, differentiable objective. After adaptation, $f_\psi(z)$ is differentiable on the space the decoder inverts, so we replace random mutation with

$$z_{k+1} = z_k + \gamma \frac{\nabla_z f_\psi(z_k)}{\|\nabla_z f_\psi(z_k)\|_2} + \delta\, \varepsilon_k, \qquad \varepsilon_k \sim \mathcal{N}(0, I). \tag{3}$$

Equation 3 differs from the standard Langevin sampler in two ways. The gradient is unit-normalized, so γ has geometric rather than scale-dependent meaning and the dynamics ignore the arbitrary units of assay labels. And the prior score $\nabla_z \log p_\phi(z)$ is realized geometrically by flow-projecting each proposal, so the prior acts as a projection operator rather than an additive force. The coefficient δ anneals over generations, and the update replaces mutation in the loop, leaving archiving and elitism unchanged. The result is a guided-diffusion-style sampler (Wei et al., 2024; Eckmann et al., 2022) in which potency pushes and the prior pulls back.

Optimizing synthesizability. Potency gradients alone might propose molecules far off synthesizable chemistry (Gao & Coley, 2020). Our first remedy makes synthesizability a term in the objective: we label pretraining molecules with AiZynthFinder (Genheden et al., 2020) and distill that expensive signal into a feed-forward synth head $g_\varphi(h) \in [0, 1]$ trained with soft BCE on frozen embeddings, giving a differentiable surrogate on the property head's own coordinates that is queryable at population scale where full retrosynthesis is not. Candidates are ranked by

$$S(x) = f_\psi(x) - \lambda_s \max(0, \tau_s - g_\varphi(x)) - \lambda_n \max(0, T(x) - \tau_n) - \lambda_{\text{ood}} \max(0, d(x) - m), \tag{4}$$

where $f_\psi(x)$ is the round-trip prediction on the re-encoded decode (rather than the pre-decode latent, which would let search exploit points corresponding to no molecule), $T(x)$ is maximum ECFP4 Tanimoto to the assay training set, and $d(x)$ a normalized latent distance to encoded anchors. All three penalties are hinges, which is the substantive choice: a molecule synthesizable enough, novel enough and in-distribution incurs no penalty, so within the feasible set ranking is pure predicted potency.

Enforcing synthesizability. The hinge only discourages infeasibility, and strong potency signal can outvote it. Our second remedy makes every scored molecule synthesizable by construction using PrexSyn (Luo & Coley, 2025a), a decoder-only transformer emitting the postfix notation of a synthetic pathway over purchasable building blocks and reaction templates (Appendix B). Because it emits a program over a fixed library rather than a structure, any output is assemblable by definition,

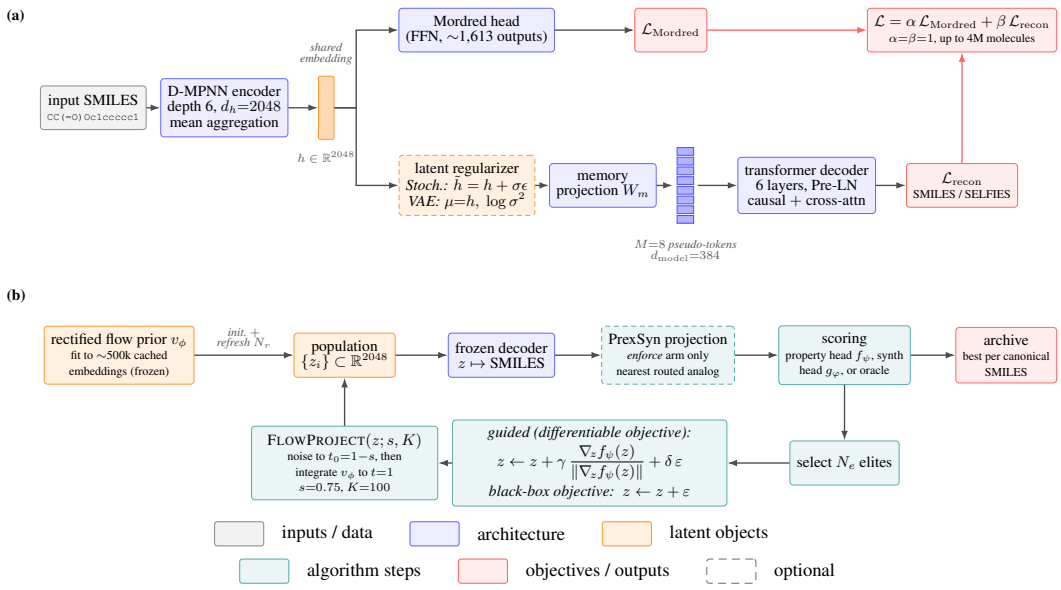

Figure 1: Generative CheMeleon. (a) Joint pretraining: one D-MPNN encoder produces h, read simultaneously by a Mordred head and by an autoregressive decoder reached through a memory projection into M=8 pseudo-tokens. The dashed block marks the two latent regularizers compared. (b) Generation and search with encoder, decoder, and prior frozen. The rectified flow both initializes the population and, as a partial noise-and-denoise projection on every offspring, keeps latents decodable. Offspring come from property-head gradients (§2.6) or Gaussian mutation (§2.5); the dashed PrexSyn block applies in the enforce arm only.

and its projection mode returns ranked synthesizable analogs with explicit routes. In the enforce arm this sits between decoding and scoring: each decode is replaced by its nearest routed analog and re-encoded, so every term is evaluated on the molecule that would actually be made and the archive contains only routed molecules, at the price of a small structural edit.

3 Experiments

3.1 Joint pretraining does not cost predictive performance

We pretrain on exactly the original corpus, 1M PubChem molecules (Kim et al., 2025; Burns et al., 2025), so any regression difference to the predictive CheMeleon is attributable to the reconstruction objective rather than to data. On 20k held-out molecules the Mordred head converges to the same validation loss as descriptor-only pretraining, 0.14 in both cases, while the decoder emits 98.2% valid strings and recovers 85.8% of the held-out molecules exactly (Table 3 in Appendix C); reconstruction is scored after RDKit canonicalisation and ignores stereochemistry, so exact match measures recovery of the molecular graph.

We then adapt to all 30 MoleculeACE assays (van Tilborg et al., 2022) against the published encoder-only baseline, using full fine-tuning (§2.4) because that is the regime in which the original numbers were obtained. Over three seeds the predictive cost is small: mean RMSE moves from 0.666 to 0.688, a 3.3% relative increase, and MAE from 0.497 to 0.513. The cliff/non-cliff decomposition shifts proportionally rather than concentrating in either subset, from 0.615 to 0.643 on non-cliff compounds against 0.762 to 0.772 on cliff compounds, so the decoder does not selectively damage hard cases (Table 4 in Appendix C). Reconstruction meanwhile improves during fine-tuning, from 85.8% to 93.7% exact match with validity rising from 98.2% to 98.9%, as expected when each assay occupies a narrower region than PubChem.

Table 1: GuacaMol distribution learning (10k samples).

Method	Valid ↑	Unique ↑	Novel ↑	KL ↑	FCD ↑
GLDM (Wang et al., 2024)	**1.000**	0.999	**0.997**	0.926	0.424
Ours (Stoch., SELFIES)	**1.000**	0.999	0.978	0.943	0.537
Ours (VAE, SELFIES)	**1.000**	0.999	0.983	0.941	0.524
Ours (Stoch., SMILES)	0.959	0.996	0.956	0.961	0.718
Ours (VAE, SMILES)	0.900	0.994	0.955	0.963	0.734
Ours (deterministic, SMILES)	0.779	0.999	0.982	**0.976**	**0.770**

3.2 GuacaMol: unconditional generation

GuacaMol (Brown et al., 2019) comprises a distribution-learning suite, asking whether samples are valid and statistically indistinguishable from a drug-like reference, and a goal-directed suite of 20 black-box objectives (§3.3), both on a curated ChEMBL split. Since published numbers are comparable only under an identical training set, we pretrain a separate model on that split ($\sim$1.4M molecules) here. We draw 10k latents per configuration and report validity, uniqueness, novelty, the Kullback-Leibler (KL) divergence over physicochemical descriptor distributions and the Fréchet ChemNet Distance (FCD) (Preuer et al., 2018). The FCD is the distance between distributions of molecules encoded by ChemNet, a neural network trained to predict biological activity. KL and FCD are exponentially transformed onto $[0, 1]$, so higher is better for all five. Table 1 exposes the design space. The vocabulary controls validity almost deterministically: SELFIES cannot express an invalid molecule (Krenn et al., 2020) and reaches 1.000 under both regularizers, while SMILES must learn well-formedness and does not fully achieve it. The regularizer controls proximity to the data distribution: within the SMILES family validity and fidelity trade off monotonically, deterministic decoding attaining the best KL/FCD (0.976/0.770) at the worst validity (0.779), with stochastic smoothing recovering 18 points of validity for 5 of FCD. Against GLDM (Wang et al., 2024), the latent graph diffusion model closest to our setting, every configuration here improves KL, the SMILES variants improve FCD by 0.10–0.35 absolute (24–82% relative), and our SELFIES configurations match GLDM's perfect validity and exceed its FCD. Figure 2 shows our runs collectively extend the empirical Pareto frontier into a previously unpopulated high-fidelity region; GLDM retains an edge on novelty, consistent with samples lying further from the reference. We adopt SMILES with stochastic smoothing throughout for its trade-off between validity and FCD. Figure 3 justifies the learned prior of §2.3: norm-matched Gaussian samples occupy a visibly disjoint region of the leading principal components, whereas rectified-flow samples cover the empirical distribution.

3.3 GuacaMol: black-box goal-directed search

The goal-directed suite spans rediscovery, similarity, isomer, median-molecule, MPO, and scaffold-hop families, detailed in Appendix B. All are oracle-only, so we apply the flow-projected GA of §2.5 on the SMILES-plus-Stoch. checkpoint with no property head. The dominant hyperparameter is projection strength s. Sweeping $s \in \{0, 0.25, 0.5, 0.75, 1.0\}$ at a fixed 500×20 budget (Figure 4a) gives a clear interior optimum at s=0.75, spanning 0.44–0.71. Both endpoints behave as the mechanism predicts: s=0 degenerates to a plain latent GA whose mutations leave the decodable region and waste oracle calls, while s=1 replaces every offspring with a prior sample and reduces search to flow sampling. That the optimum lies closer to the prior indicates that in 2048 dimensions re-anchoring matters more than preserving mutation detail. We fix s=0.75, K=100 throughout. Performance then scales monotonically without saturation in budget (Figure 4b), from 0.71 at 500×20 to 0.90 at 16000×640, so the ceiling is search effort rather than latent-space expressivity. That places Generative CheMeleon on the same score of Graph GA, which operates directly on graphs with hand-designed operators, and ahead of the SMILES LSTM (0.87), SMILES GA (0.72), the best molecule in the dataset (0.61) and Graph MCTS (0.45), as reported by Brown et al. (2019). We report this to show that the latent space is searchable with a plug-in objective, replaceable by an in-house model without retraining. The intended use, however, is a differentiable objective head on the jointly pretrained embedding.

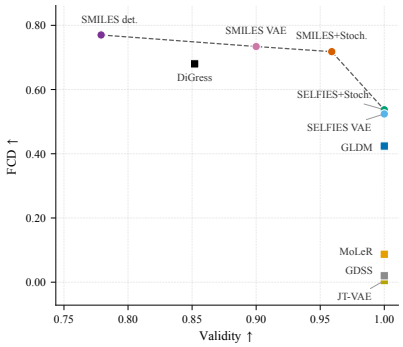

Figure 2: Validity–FCD frontier on GuacaMol distribution learning (10k samples, both axes higher better), against GLDM and its reported baselines.

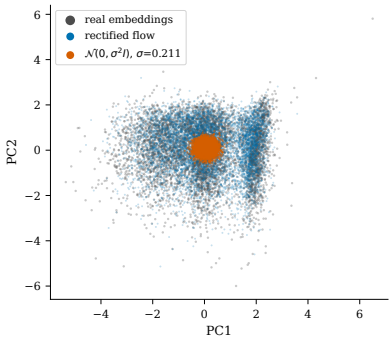

Figure 3: First two principal components of 20k embeddings (SMILES + Stoch. backbone): real embeddings (gray), rectified-flow samples (blue), and norm-matched Gaussian samples (orange).

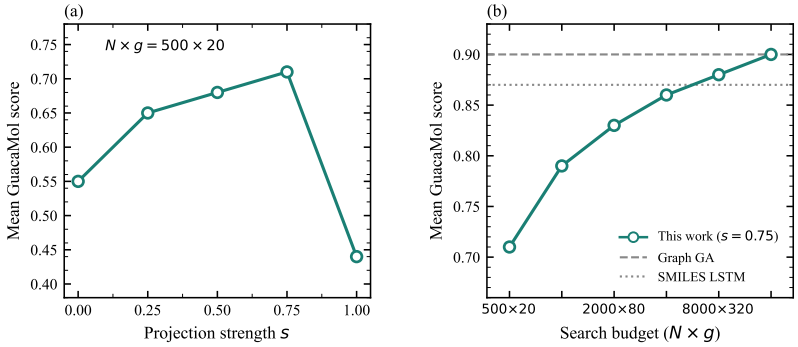

Figure 4: Dependence of goal-oriented GuacaMol score on (a) flow projection strength s at fixed population size $N = 500$ and number of generations $g = 20$; and on (b) search budget $N \times g$, thus increasing both population size and number of generations, at fixed s=0.75. Dashed and dotted lines mark Graph GA and SMILES LSTM as reported by Brown et al. (2019).

3.4 Property-driven molecular design on MoleculeACE

We evaluate on the 30 MoleculeACE bioactivity benchmarks (van Tilborg et al., 2022), the collection on which predictive CheMeleon was validated (Burns et al., 2025), which lets us hold the representation fixed while testing if it is suitable for the inverse problem, molecule-generation. Per assay we fine-tune a head, run the Langevin-guided search of §2.6 under both synthesizability regimes, and require a valid demonstration: one molecule simultaneously synthesizable, by the distilled criterion (optimize) or PrexSyn route feasibility (enforce); novel, nearest-neighbour ECFP4 Tanimoto below 0.6; and potent, predicted activity above the 90th training percentile. All three must hold for the same molecule, so a demonstration cannot be manufactured by trading axes.

Pretraining corpora. By design, our method tends to stay reasonably close to the chemical space defined by the pre-training set, to increase the likelihood that the potency predictions are accurate and that the model will generate reasonable molecules. These considerations suggest one should use a diverse pre-training set. However, different types of molecules are more synthesizable than others; for example Gao and Coley's synthesis planner (Gao & Coley, 2020) could generate syntheses for 90% of the molecules in the MOSES set (Polykovskiy et al., 2020), but only about 60% of the molecules in ZINC (Sterling & Irwin, 2015). If the pre-training set is too broad many of the

Table 2: Number of MoleculeACE Assays where our method, using different pre-training sets and methods for determining synthesizability, successfully generated a synthesizable potent novel molecule. 30 is a perfect score. *Optimize* uses the distilled synth-head hinge of Eq. equation 4; *enforce* uses PrexSyn projection. Fine-tuning was applied only to cases where the default MCP pre-training failed to generate a suitable molecule.

Synth. strategy	Pre-training corpus			
	ChEMBL	MOSES	MCP	
			as pre-trained	targeted fine-tuning
Optimize (synth head)	25	22	27	**29**
Enforce (PrexSyn)	26	21	28	**30**

computer-generated molecules may be difficult to synthesize (and so useless as drugs). Here we test the performance of our method using 3 pre-training sets: MOSES, ChEMBL (Gaulton et al., 2012), and MCP, which is a deduplicated union of MOSES, ChEMBL, and the 1 M molecules from PUBCHEM used to pre-train CheMeleon (Kim et al., 2025; Burns et al., 2025).

Valid demonstrations. Table 2 shows that for most assays our method generates at least one molecule that is simultaneously novel, synthesizable, and has high potency. The method's performance is best when the larger MCP pre-training set is employed and synthesizability is explicitly enforced using PrexSyn: in that case it generates satisfactory molecules for 28 of the 30 assays. The worst performance is achieved with the model pre-trained on MOSES, which is counterintuitive since it is by far the most synthesizable corpus (Gao & Coley, 2020). The explanation is which criterion binds: MOSES's synthesizability comes from aggressive filtering to small lead-like chemistry, narrowing its embedding distribution and limiting novel, high-potency analogs for targets with larger actives. This narrowness is measurable: MOSES embeddings have both lower mean norm and spread than ChEMBL and MCP (Appendix D). Once synthesizability is enforced during search, corpus-level synthesizability is no longer the bottleneck; coverage is, and MCP retains MOSES's chemistry while adding ChEMBL and PubChem breadth. The near-parity of the two regimes is itself informative: a distilled surrogate over frozen embeddings recovers most of the benefit of hard projection far more cheaply, while projection alone contributes the guarantee and an explicit route (Figure 5). Both MCP configurations fail on `CHEMBL2835` and `CHEMBL2971` by default. Applying targeted full fine-tuning (§2.4, with the prior refit) plus a doubled budget to those failures recovers them: enforcement reaches 30/30 and optimization 29/30. That hard cases yield to representation adaptation rather than to more search indicates their difficulty is coverage, not optimization, and confirms the two adaptation regimes are complementary: frozen backbones by default, full fine-tuning as escalation. Across corpora the auxiliary heads are essentially unaffected (Appendix D), and MOSES attains the highest exact reconstruction (99.36%) while yielding the fewest designs, so reconstruction fidelity is not the quantity to optimize.

4 Conclusion

A predictive molecular foundation model can be made generative without being diminished as a predictor. Attaching an autoregressive decoder to the CheMeleon D-MPNN through a small set of memory tokens, and optimizing descriptor regression and reconstruction jointly at pretraining scale, yields one 2048-dimensional embedding that is property representation, generative latent and search domain at once, while only slightly worsening the RMSE of the property predictions across 30 MoleculeACE assays. Three choices proved load-bearing. CheMeleon's width makes a fixed Gaussian prior untenable, so we learn one: a rectified flow that samples and projects arbitrary latents back onto the decodable manifold, making search in 2048 dimensions viable. Where the objective is differentiable we substitute normalized property gradients in a Langevin sampler; where it is not, the same machinery accepts any oracle. Synthesizability must act inside the loop, our two mechanisms proved nearly equivalent in yield. Counterintuitively, the most synthesizable public corpus (Gao & Coley, 2020), gives the best reconstruction but is the least successful at generating potent new molecules. Instead, the broadest pre-training corpus is the most successful as long as synthesizability

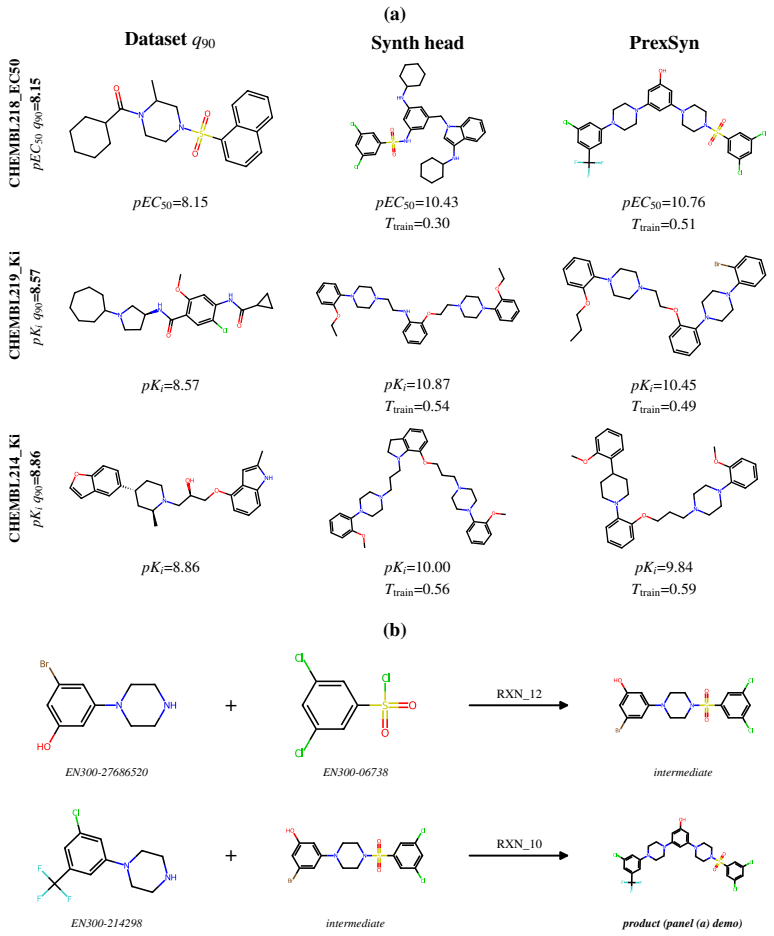

Figure 5: Panel (a) depicts one row per assay, one column per source of molecules. The three rows are three of the 30 MoleculeACE assays (CHEMBL218_EC50, CHEMBL219_Ki, CHEMBL214_Ki), and each row label reports the assay together with its q_{90}, the 90th percentile of the measured potencies in that assay. The first column shows the molecule that attains this q_{90} value: it is the reference a generated candidate must beat for the demonstration to count as valid. The second and third columns show, for the same assay, one molecule generated in this work using respectively the distilled synth-head hinge and the PrexSyn projection. Each generated molecule is annotated with its predicted potency (pEC_{50} or pK_i) and with T_{train}, the largest Tanimoto similarity between that molecule and any compound of the corresponding assay. (b) PrexSyn route for the CHEMBL218_EC50 demonstration; each precursor carries its Enamine catalogue ID.

is enforced, suggesting that coverage is more crucial than the aggregate quality of the pretraining data. Potency here is certified by the fine-tuned property head rather than by an assay, and that head is asked to extrapolate above the 90th percentile of its own labels, where its error is least constrained. The experiments establish that the generative half is not the weak link: reconstruction is near-exhaustive (§3.1), samples match the reference distribution (§3.2), and flow projection keeps search on the decodable manifold (§3.3). A decoded candidate is therefore a real, synthesizable molecule whose predicted activity is the remaining uncertainty. Confidence in a demonstration consequently reduces to confidence in the property predictor for that assay, and closing that gap calls for prospective synthesis, for which the enforce arm already supplies routes. Taken together, these results collapse a separation the field has largely taken for granted: one descriptor-supervised representation, trained once on cheap, noise-free labels, can score a molecule for potency, generate a new one, and vouch for its synthesis.

Reproducibility Statement

All components needed to reproduce our results are specified in the paper. The architecture is given in §2.1 and Figure 1; the joint objective in Eq. equation 1; the flow prior and projection operator in §2.3; and the search procedure in full as Algorithm 1. Appendix E tabulates every hyperparameter for pretraining, flow fitting, downstream adaptation, synthesizability distillation, and both search modes, together with the software versions and the hardware used. Appendix F documents corpus versions, preprocessing, exact split sizes, the deduplication and MoleculeACE leakage-exclusion procedure used to build the MCP corpus, and the evaluation protocol behind every reported metric; the held-out identifier lists are included in the supplementary material.

We build on publicly available components, each used with its released configuration: the CheMeleon encoder (Burns et al., 2025), Mordred descriptors (Moriwaki et al., 2018), AiZynthFinder with the public USPTO policies and ZINC stock (Genheden et al., 2020), PrexSyn (Luo & Coley, 2025a), and the GuacaMol (Brown et al., 2019) and MoleculeACE (van Tilborg et al., 2022) benchmarks. The supplementary material additionally contains the generated archives underlying Tables 1–2: all GuacaMol goal-directed candidates and the 30 MoleculeACE demonstrations with predicted activity, nearest-training-set Tanimoto, and full PrexSyn routes, so that every reported number can be recomputed from the released artifacts. Code will be released upon publication.

AI Use Statement

Large language model assistants were used in the preparation of this manuscript for writing support only: rephrasing and condensing author-written prose, improving clarity and grammar, and suggesting alternative formulations for titles, captions and section headings. No part of the research itself was AI-generated. The model architecture, training objectives, experimental design, evaluation protocol, and the analysis and interpretation of the results are the authors' own, and no sentence describing a result was written without an author having verified that result against the underlying experiments. Every AI-assisted passage was read, edited and approved by the authors, all citations were checked against the primary sources, and the authors take full responsibility for the entire contents of this submission.

References

Viraj Bagal, Rishal Aggarwal, P. K. Vinod, and U. Deva Priyakumar. MolGPT: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, 2022. doi: 10.1021/acs.jcim.1c00600.

Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. REINVENT 2.0: An AI tool for de novo drug design. *Journal of Chemical Information and Modeling*, 60(12):5918–5922, 2020. doi: 10.1021/acs.jcim.0c00915.

Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, Cambridge, 2020.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pp. 10–21, 2016.

Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019. doi: 10.1021/acs.jcim.8b00839.

Jackson W. Burns, Akshat Shirish Zalte, Charlles R. A. Abreu, Jochen Sieg, Christian Feldmann, Miriam Mathea, and William H. Green. Deep learning foundation models from classical molecular descriptors. *arXiv preprint arXiv:2506.15792*, 2025.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. SCScore: Synthetic complexity learned from a reaction corpus. *Journal of Chemical Information and Modeling*, 58(2): 252–261, 2018. doi: 10.1021/acs.jcim.7b00622.

Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations (ICLR)*, 2019.

Peter Eckmann, Kunyang Sun, Bo Zhao, Mudong Feng, Michael K. Gilson, and Rose Yu. LIMO: Latent inceptionism for targeted molecule generation. In *International Conference on Machine Learning (ICML)*, 2022.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, 2009. doi: 10.1186/1758-2946-1-8.

Wenhao Gao and Connor W. Coley. The synthesizability of molecules proposed by generative models. *Journal of Chemical Information and Modeling*, 60(12):5714–5723, 2020. doi: 10.1021/acs.jcim.0c00174.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2022a.

Wenhao Gao, Rocío Mercado, and Connor W. Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. In *International Conference on Learning Representations (ICLR)*, 2022b.

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist, and Esben Bjerrum. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of Cheminformatics*, 12(1):70, 2020. doi: 10.1186/s13321-020-00472-1.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations (ICLR)*, 2020.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

David E Graff, Nathan K Morgan, Jackson W Burns, Anna C Doner, Brian Li, Shih-Cheng Li, Joel Manu, Angiras Menon, Hao-Wei Pang, Haoyang Wu, et al. Chemprop v2: An efficient, modular machine learning package for chemical property prediction. *Journal of Chemical Information and Modeling*, 66(1):28–33, 2026.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020. doi: 10.1039/C9SC04026A.

Esther Heid, Kevin P. Greenman, Yunsie Chung, Shih-Cheng Li, David E. Graff, Florence H. Vermeire, Haoyang Wu, William H. Green, and Charles J. McGill. Chemprop: A machine learning package for chemical property prediction. *Journal of Chemical Information and Modeling*, 64(1): 9–17, 2024. doi: 10.1021/acs.jcim.3c01250.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3 (1):015022, 2022. doi: 10.1088/2632-2153/ac3ffb.

Andrew Jaegle, Felix Gimeno, Andy Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning (ICML)*, 2021.

Jan H. Jensen. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. doi: 10.1039/C8SC05372C.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning (ICML)*, 2018.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. PubChem 2025 update. *Nucleic Acids Research*, 53(D1):D1516–D1525, 2025. doi: 10.1093/nar/gkae1059.

Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Kerstin Klaser, Błażej Banaszewski, Samuel Maddrell-Mander, Callum McLean, Luis Müller, Ali Parviz, Shenyang Huang, and Andrew Fitzgibbon. MiniMol: A parameter-efficient foundation model for molecular learning. *arXiv preprint arXiv:2404.14986*, 2024.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Shitong Luo and Connor W. Coley. Efficient and programmable exploration of synthesizable chemical space. *arXiv preprint arXiv:2512.00384*, 2025a.

Shitong Luo and Connor W. Coley. Efficient and programmable exploration of synthesizable chemical space. *arXiv preprint arXiv:2512.00384*, 2025b.

Shitong Luo, Wenhao Gao, Zuofan Wu, Jian Peng, Connor W. Coley, and Jianzhu Ma. Projecting molecules into synthesizable chemical spaces. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33289–33304. PMLR, 2024.

Natalie Maus, Haydn T. Jones, Juston S. Moore, Matt J. Kusner, John Bradshaw, and Jacob R. Gardner. Local latent space Bayesian optimization over structured inputs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022.

Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*, 10(1):4, 2018. doi: 10.1186/s13321-018-0258-y.

AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alán Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. In *International Conference on Learning Representations (ICLR)*, 2020.

AkshatKumar Nigam, Robert Pollice, and Alán Aspuru-Guzik. Parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. *Digital Discovery*, 1(4):390–404, 2022. doi: 10.1039/D2DD00003B.

Emmanuel Noutahi, Cristian Gabellini, Michael Craig, Jonathan S. C. Lim, and Prudencio Tossou. Gotta be SAFE: a new framework for molecular design. *Digital Discovery*, 3(4):796–804, 2024. doi: 10.1039/D4DD00019F.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, 2017. doi: 10.1186/s13321-017-0235-x.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11:565644, 2020. doi: 10.3389/fphar.2020.565644.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018. doi: 10.1021/acs.jcim.8b00234.

Danny Reidenbach, Micha Livne, Rajesh K. Ilango, Michelle Gill, and Johnny Israeli. Improving small molecule generation using mutual information machine. *arXiv preprint arXiv:2208.09016*, 2022.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022. doi: 10.1038/s42256-022-00580-7.

Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018. doi: 10.1021/acscentsci.7b00512.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2022.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Amol Thakkar, Veronika Chadimová, Esben Jannik Bjerrum, Ola Engkvist, and Jean-Louis Reymond. Retrosynthetic accessibility score (RAscore) – rapid machine learned synthesizability classification from AI driven retrosynthetic planning. *Chemical Science*, 12(9):3339–3349, 2021. doi: 10.1039/D0SC05401A.

Jakub M. Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Derek van Tilborg, Alisa Alenicheva, and Francesca Grisoni. Exposing the limitations of molecular machine learning with activity cliffs. *Journal of Chemical Information and Modeling*, 62(23): 5938–5951, 2022. doi: 10.1021/acs.jcim.2c01073.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Number 47 in Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2018.

Conghao Wang, Hiok Hian Ong, Shunsuke Chiba, and Jagath C. Rajapakse. GLDM: hit molecule generation with constrained graph latent diffusion model. *Briefings in Bioinformatics*, 25(3): bbae142, 2024. doi: 10.1093/bib/bbae142.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022. doi: 10.1038/s42256-022-00447-x.

Guanghao Wei, Yining Huang, Chenru Duan, Yue Song, and Yuanqi Du. Navigating chemical space with latent flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237.

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

A FLOW-PROJECTED LATENT SEARCH

Algorithm 1 states the search loop of §2.5 and §2.6 in full. Black-box goal-directed runs use Gaussian mutation with the GuacaMol oracle as R; property-driven runs use the Langevin step of Eq. equation 3 with the composite score of Eq. equation 4.

B ADDITIONAL METHOD DETAILS

Why full fine-tuning also refits the prior (§2.4). Full fine-tuning carries two costs that frozen-backbone adaptation does not. First, the assay loss alone would leave the decoder unsupervised while the encoder drifts, which is why we retain the reconstruction term in $\mathcal{L}_{\text{ft}}$. Second, the pre-training prior no longer describes the adapted encoder's output: sampling from a stale prior places latents where the updated decoder has no obligation to invert, reintroducing the prior-hole problem the learned prior was introduced to remove. We therefore re-encode the cached corpus and retrain v_ϕ via Eq. equation 2 before generating. The refit is performed on the re-encoded corpus rather than on the assay set, since a few thousand labels cannot characterize a 2048-dimensional distribution, and a prior fit only to actives would collapse search onto known chemistry and defeat novelty.

GuacaMol goal-directed task families (§3.3). The suite of 20 objectives spans *rediscovery* and *similarity* tasks, which reward recovering or approaching a held-out reference ligand; *isomer* tasks, which fix a molecular formula; *median-molecule* tasks, which require simultaneous similarity to two dissimilar targets; *MPO* tasks, which combine similarity with logP, TPSA, and molecular-weight windows; and *hop* tasks, which demand pharmacophore retention under a scaffold change. Every objective is exposed only as a SMILES-to-scalar oracle, which is why the same flow-projected loop applies unchanged across all five families.

Algorithm 1 Flow-projected latent search (with optional property guidance)

Require: N, generations g, elites N_e, mutation scale σ_m, refresh N_r, strength s, steps K, score R (oracle or composite S, Eq. equation 4), optional head f_ψ, optional PrexSyn projection

$\mathcal{A} \leftarrow \emptyset$; $P \leftarrow \textsc{SamplePrior}(N)$ ▷ rectified flow prior, Eq. equation 2
for $t = 1$ **to** g **do**
  for each $z \in P$ **do**
    $x \leftarrow \textsc{Decode}(z)$
    if *enforcement* enabled **then**
      $x \leftarrow \textsc{PrexSynProject}(x)$; $z \leftarrow \textsc{Encode}(x)$ ▷ nearest routed analog
    end if
    $\mathcal{A} \leftarrow \textsc{UpdateArchive}(\mathcal{A}, x, R(x))$ ▷ best per canonical SMILES
  end for
  $E \leftarrow \textsc{TopElites}(P, N_e)$; $C \leftarrow \emptyset$
  for $i = 1$ **to** $N - N_e - N_r$ **do**
    $z \leftarrow$ uniform draw from E
    $z \leftarrow \textsc{LangevinStep}(z; f_\psi, \gamma, T_t)$ **if** guidance **else** $z + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, \sigma_m^2 I)$
    $C \leftarrow C \cup \{\textsc{FlowProject}(z; s, K)\}$
  end for
  $P \leftarrow E \cup C \cup \textsc{SamplePrior}(N_r)$
end for
return $\mathcal{A}$

PrexSyn projection (§2.6). PrexSyn (Luo & Coley, 2025a) is a decoder-only transformer trained on a billion-scale stream of synthetic pathways, conditioned on descriptors or fingerprints, that emits the postfix notation of a pathway: a stack-machine encoding over purchasable building blocks and reaction templates. Its projection mode returns ranked synthesizable analogs of a query together with explicit routes, with feasibility measured as ECFP4 Tanimoto between query and analog (1.0 when the query itself is exactly routed). In the enforce arm each decoded molecule is replaced by its nearest routed analog before scoring, so every archived candidate carries a route over a fixed building-block library. The two synthesizability regimes therefore certify against different libraries, AiZynthFinder's purchasable stock versus PrexSyn's building blocks.

C JOINT PRETRAINING DIAGNOSTICS

Tables 3 and 4 report in full the quantities quoted in §3.1: reconstruction and Mordred loss after joint pretraining and after downstream fine-tuning, and the MoleculeACE property-prediction comparison against the encoder-only CheMeleon baseline.

Table 3: Reconstruction and Mordred loss. Pretraining: 1M PubChem, 20k held out. Finetuning: 30-assay average.

Stage	Mordred ↓	Valid	Exact
Descriptor only	0.14	—	—
Joint pretrain	0.14	98.2%	85.8%
After finetune	—	98.9%	93.7%

Table 4: MoleculeACE property prediction (30-assay mean, three seeds, both fully fine-tuned).

Model	MAE	RMSE	Non-cl.	Cliff
CheMeleon	**0.497**	**0.666**	**0.615**	**0.762**
Ours (joint)	0.513	0.688	0.643	0.772

D PRETRAINING CORPUS COMPARISON

Table 5 reports validation metrics for every component of the stack across corpora. The property-head loss is essentially invariant, so neither the choice of corpus nor full fine-tuning compromises the predictive capability established in §3.1, and the same holds for the distilled synth head. Corpora differ in reconstruction: MOSES attains the highest exact match, exactly as expected for the narrowest corpus, and its slightly lower flow-matching loss reflects an easier embedding distribution. Neither quantity translates into downstream design success. Table 6 reports the embedding-

norm statistics referenced in §3.4, computed over the training set of each corpus's flow prior; the MOSES-pretrained encoder produces embeddings that are both smaller in norm and less dispersed, which is the geometric form its narrow coverage takes.

Table 5: Pretraining corpus comparison (validation metrics). "MCP ft." is the fully fine-tuned variant used for the targeted recoveries in Table 2.

Metric	ChEMBL	MOSES	MCP	CheMeleon	MCP ft.
Exact reconstruction ↑	96.84%	**99.36%**	90.63%	—	—
Mordred loss ↓	0.12	**0.08**	0.11	0.14	—
Flow-matching loss ↓	1.04	**0.91**	1.04	—	1.04
Synth-head loss ↓	0.10	**0.09**	0.10	—	0.11
Property-head loss ↓	0.69	0.73	0.69	**0.67**	**0.67**

Table 6: Mean and standard deviation of the embedding norm for the three pretrained models, computed over the flow-prior training set of each corpus.

	ChemBL	MOSES	MCP
Mean	9.5	7.4	9.2
Std. Dev.	1.8	0.8	2.1

E Implementation and hyperparameters

Software and hardware. All models are implemented in PyTorch 2.10 with Lightning 2.6.1 and Chemprop 2.2.2; chemistry uses RDKit 2026.03.2, Mordred descriptors come from `mordredcommunity` 2.0.7, SELFIES from `selfies` 2.2.0, and benchmarks from `guacamol` 0.5.5 with the PyTorch `fcd` package for Fréchet ChemNet Distance. Every run in this paper was executed on a single node of $8\times$ NVIDIA RTX 2080 Ti (11 GB each); pretraining used 1–4 of these GPUs per run, and all other stages a single GPU. The global seed is 42 everywhere unless a table reports a seed average.

Joint pretraining (§2.2). Table 7 lists the settings shared by every corpus. The encoder and Mordred head follow CheMeleon's published configuration (Burns et al., 2025) unchanged, so that any difference in §3.1 is attributable to the added reconstruction term. The per-step batch is B molecules per device; the effective batch is $B \times$ devices $\times$ accumulation, reported per run in Table 8. All runs reached the 100-epoch cap without triggering early stopping. Model selection is on validation reconstruction cross-entropy.

Latent regularizer configurations (§2.2). The four configurations compared in Table 1, plus the deterministic reference, differ only in the tokenization and in how the decoder input is regularized (Table 8). *Noise* adds $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ with σ=0.5 to h before decoding, together with a fixed-σ Gaussian penalty of weight 5.0 ramped linearly over epochs 5–15; with σ held constant this term reduces to an L^2 pull on $\|h\|$ and does not by itself yield a samplable prior, which is why §2.3 learns one. *VAE* predicts a per-dimension $\log \sigma^2$ over the same 2048 coordinates (zero-initialized linear head, bias -2.0 or -3.0) and uses the analytic diagonal-Gaussian KL, with free bits and word dropout as listed. *Deterministic* decodes h directly.

Rectified flow prior (§2.3). The velocity field v_ϕ is an MLP with sinusoidal time conditioning: a 128-dimensional time embedding passed through a two-layer SiLU MLP, an input projection to width 1024, 12 residual blocks (LayerNorm, two 1024×1024 linear layers, SiLU, additive per-block time projection), and a final LayerNorm plus projection back to 2048. The flow operates on per-dimension standardized embeddings, with the mean and standard deviation of the cached training set stored in the checkpoint and inverted at sampling time; no PCA and no optimal-transport coupling are used. Training: $100{,}000$ steps of AdamW at learning rate 2×10^{-4}, cosine-annealed to

Table 7: Joint pretraining hyperparameters, shared across all corpora.

Component	Setting	Value
Encoder	message passing	BondMessagePassing D-MPNN
	hidden size / depth	2048 / 6
	activation	LeakyReLU
	atom/bond featurizer	RIGR
	aggregation	mean over atoms
Mordred head	architecture	FFN, 1 hidden layer of width 2048
	outputs	1613 descriptors
	loss	RandomDropoutMSE, 30% columns masked/step
Decoder	memory tokens M	8
	d_{model} / heads	384 / 8
	layers / FFN width	6 (Pre-LN) / 1536
	dropout	0.1
	max sequence length	128 tokens
	label smoothing	0.1
	parameters	$\sim$20.6M
Optimization	loss weights α, β	1.0, 1.0
	LR schedule	$10^{-4} \to 10^{-3} \to 10^{-4}$, 5 warmup epochs
	weight decay	10^{-5}
	gradient clipping	1.0
	max epochs / patience	100 / 10
	initialization	random (no inherited weights)
	seed	42

Table 8: Per-run pretraining configurations. B is molecules per device per step; "eff." is the effective batch after accumulation and data parallelism. $|V|$ is the tokenizer vocabulary. All runs use the settings of Table 7 otherwise.

Run	Corpus	Tok.	Regularizer	B / eff.	$\lvert V\rvert$	Free bits	Word drop.
Table 3	PubChem-1M	SMILES	deterministic	324 / 324	334	—	—
Ours (Stoch., SMILES)	ChEMBL-GuacaMol	SMILES	σ=0.5, w=5.0	64 / 1024	106	—	—
Ours (Stoch., SELFIES)	ChEMBL-GuacaMol	SELFIES	σ=0.5, w=5.0	64 / 1024	109	—	—
Ours (VAE, SMILES)	ChEMBL-GuacaMol	SMILES	VAE, γ=10^{-3}	64 / 512	106	0.005	0.2
Ours (VAE, SELFIES)	ChEMBL-GuacaMol	SELFIES	VAE, γ=1.0	64 / 1024	109	0.2	0.1
Ours (determ., SMILES)	ChEMBL-GuacaMol	SMILES	none	64 / 1024	106	—	—
ChEMBL (§3.4)	ChEMBL-GuacaMol	SMILES	σ=0.5, w=5.0	64 / 1024	106	—	—
MOSES (§3.4)	MOSES	SMILES	σ=0.5, w=5.0	64 / 512	28	—	—
MCP (§3.4)	MCP	SMILES	σ=0.5, w=5.0	64 / 512	343	—	—

5% of that value, batch size 512, weight decay 0, gradient clipping 1.0, EMA decay 0.999, $\sigma_{\min}$=0, held-out set of 4096 embeddings, seed 42. The cache is 500,000 training embeddings encoded in batches of 256. Sampling and projection integrate the field with explicit Euler; K is stated per experiment below.

Downstream adaptation (§2.4). The frozen-backbone property head f_ψ is an FFN with 2 hidden layers of width 512 and dropout 0.1, trained on cached embeddings with AdamW at learning rate 3×10^{-4}, weight decay 10^{-5}, 200 warmup steps, batch size 64, up to 60 epochs with patience 15, and a 20% validation split of the assay training set. Targets are standardized to zero mean and unit variance using training statistics, and predictions are inverted before any reported metric. Full fine-tuning (Table 4, and the targeted recoveries in Table 2) updates encoder, decoder and head jointly with $\mathcal{L}_{\text{ft}} = \mathcal{L}_{\text{assay}} + \beta_{\text{ft}}\mathcal{L}_{\text{recon}}$, β_{ft}=1, a backbone learning rate of 10^{-5} against 3×10^{-4} for the new parameters, and refits v_ϕ on the re-encoded corpus with the recipe above before generating.

Synthesizability head (§2.6). Labels come from AiZynthFinder (Genheden et al., 2020) with the public USPTO expansion and filter policies and the ZINC purchasable stock, capped at 45 s and 100 MCTS iterations per molecule, run with 48 workers. The distillation set is 8000 molecules decoded from flow-prior samples plus 6000 molecules drawn from the pretraining corpus, so that the head sees both the generated and the reference distributions. g_φ is an FFN with 2 hidden layers of width 512 and dropout 0.1 on frozen embeddings, trained with soft BCE, AdamW at 3×10^{-4}, weight decay 10^{-5}, 200 warmup steps, batch size 128, up to 80 epochs with patience 15, and 10%/10% validation/test splits. One head is distilled per pretraining corpus.

PrexSyn projection (§2.6). We use PrexSyn (Luo & Coley, 2025a) with the `enamine2310_rxn115` chemical space (115 reaction templates over the Enamine 2023-10 building-block catalogue) and ECFP4 conditioning. Projection draws 16 pathway samples per query during search and 128 for the exported demonstrations, keeping the analog of highest ECFP4 Tanimoto to the query.

Search hyperparameters (§2.5, §2.6). Table 9 gives the settings for both search modes. Elites are the top 10% of the population in both. One detail of Eq. equation 4 is worth stating explicitly: in the optimize arm λ_s is not held fixed but adapted per generation by a Lagrangian controller that drives the fraction of the population clearing τ_s toward 0.5, initialized at λ_s=3.0 with step size 4.0 and capped at 20.0. This removes a per-assay weight to tune; λ_n and λ_{ood} are fixed. Runs stop early if the best composite score improves by less than 10^{-4} for 8 consecutive generations.

Table 9: Search hyperparameters. GuacaMol goal-directed uses Gaussian mutation with the oracle as R; MoleculeACE uses the Langevin step of Eq. equation 3 with the composite score of Eq. equation 4.

	Symbol	GuacaMol (§3.3)	MoleculeACE (§3.4)
Population	N	500–8000	500 (opt.) / 200 (enf.)
Generations	g	20–320	30 (opt.) / 20 (enf.)
Elite fraction	N_e/N	0.1	0.1
Prior refresh	N_r	50	50
Mutation scale	σ_m	0.3	— (Langevin)
Projection strength	s	0.75	0.75
Projection steps	K	100	20
Integrator	—	Euler	Euler
Decoding	—	greedy	greedy
Guidance scale	γ	—	0.5
Langevin noise	δ	—	$0.1 \to 0.0$ (annealed)
Synth. hinge weight	λ_s	—	adaptive, λ_0=3.0
Synth. threshold	τ_s	—	0.5 (opt.) / 0.8 (enf.)
Novelty weight	λ_n	—	3.0
Novelty threshold	τ_n	—	0.6
OOD weight	λ_{ood}	—	1.0
OOD margin	m	—	1.0

F Data, splits, and evaluation protocol

Corpora. Table 10 gives exact molecule counts for every pretraining corpus. PubChem-1M is the 1M-molecule subset used by CheMeleon (Burns et al., 2025; Kim et al., 2025). ChEMBL-GuacaMol is the curated ChEMBL split distributed with GuacaMol (Brown et al., 2019), used unmodified so that Table 1 is comparable to published numbers. MOSES is the released split (Polykovskiy et al., 2020). MCP is their deduplicated union, constructed as follows: concatenating the three sources gives 4,176,041 rows, of which 57,811 MOSES entries and 20,128 PubChem entries already appear in an earlier source; one SMILES fails RDKit parsing and a further 406 collide only after canonicalization, leaving 4,098,102 unique canonical SMILES. We then remove every molecule whose InChIKey appears anywhere in the union of the 30 MoleculeACE assays (35,633 InChIKeys,

matching 20,219 corpus molecules), so that the MCP-pretrained models used for §3.4 have not seen any assay molecule during pretraining. The final corpus is 4,077,476 molecules.

Table 10: Pretraining corpus sizes after preprocessing. Splits are $90/10$ train/validation with seed 42. Totals are below the raw source counts because a small number of molecules (between 52 and 141 per corpus) are dropped when Mordred descriptors cannot be computed.

Corpus	Train	Validation	Total
PubChem-1M	899,748	100,116	999,864
ChEMBL-GuacaMol	1,432,080	159,246	1,591,326
MOSES	1,425,924	158,598	1,584,522
MCP (union, deduplicated)	3,669,624	407,754	4,077,378

Preprocessing. SMILES are canonicalized with RDKit and deduplicated on the canonical string. For each molecule we compute the 1613 Mordred descriptors retained by CheMeleon (Moriwaki et al., 2018; Burns et al., 2025). Descriptor columns are winsorized at ± 6 standard deviations and then standardized to zero mean and unit variance using *training-split* statistics only; undefined and non-finite entries are masked out of the loss rather than imputed. The train/validation split is a uniform $90/10$ draw at seed 42. Tokenization is a character/bracket-level SMILES tokenizer with $\langle$pad$\rangle$, $\langle$bos$\rangle$, $\langle$eos$\rangle$ and $\langle$unk$\rangle$, vocabulary built on the training split (sizes in Table 8), sequences capped at 128 tokens. SELFIES runs use `selfies` 2.2.0 over the same molecules and are converted back to SMILES before any RDKit evaluation.

Reconstruction metrics. Reconstruction is evaluated by greedy autoregressive decoding of held-out embeddings, on a fixed cap of 20,000 validation molecules. *Validity* is the fraction of decoded strings RDKit can parse. *Exact match* compares canonical SMILES. We report two variants: *strict*, which compares full canonical SMILES, and *stereo-stripped*, which applies `RemoveStereochemistry` to both gold and prediction before comparison and therefore measures recovery of the molecular graph. Table 3 reports stereo-stripped exact match, as stated in §3.1; Table 5 reports strict exact match. Unparseable predictions never count as matches under either variant.

GuacaMol. Both suites use benchmark version `v2` of `guacamol` 0.5.5 with its bundled ChEMBL reference set. Distribution learning draws 10,000 latents from the flow prior and decodes greedily (no temperature, top-k or top-p sampling). KL and FCD are the package's own exponentially transformed scores. Goal-directed runs expose each objective only as a SMILES-to-scalar callable; the search budget is $N \times g$, and the archive keeps the best score per canonical SMILES.

MoleculeACE. We use the 30 assays and the published train/test splits of van Tilborg et al. (2022), with the cliff/non-cliff annotations as distributed. Table 4 averages over three seeds. A *valid demonstration* (Table 2) requires a single molecule that is simultaneously synthesizable, by $g_\varphi \geq \tau_s$ in the optimize arm or by returning a PrexSyn route in the enforce arm; novel, with maximum ECFP4 Tanimoto to the assay training set below 0.6 (radius 2, 2048 bits); and potent, with predicted activity above the 90th percentile of the assay training labels. Predicted activity is the round-trip prediction f_ψ on the re-encoded decoded molecule, not on the pre-decode latent.